Property of
Flood Control District of MC Library
Please Return to
2801 W. Durango
Phoenix, AZ 85009

DRAFT

OPTIMAL CONTROL OF HYDROSYSTEMS

by

Larry W. Mays, Ph.D., P.E., P.H.

LIBRARY

Department of Civil Engineering
Arizona State University
Tempe, Arizona 85287

PREFACE

Optimal Control of Hydrosystems is a book that addresses the mathematical modeling of the optimal operation of hydrosystems. Optimization problems are characterized or mathematically formulated to include an objective function that is optimized (maximized or minimized) subject to a set of constraints which are algebraic equations and/or inequalities. **Optimal control** problems are optimization problems in which part or all of the constraints are differential equations. In particular these systems are modeled in the framework of a certain type of optimization problem referred to as **discrete-time optimal control problems**. These types of optimization problems are unique in that the physics (laws of motion) of the problem are described through differential equations that simulate the physical behavior of the problem. The main theoretical approaches to solve optimal control problem are calculus of variations, the maximum/or minimum principle, which may be regarded as a special application of calculus of variation, dynamic programming approaches and mathematical programming (nonlinear programming). The methods to solve these types of problems include the combined use of (a) the hydraulic simulation of the physical process and (b) operation research techniques such as nonlinear programming and differential dynamic programming. The methods to solve these types of hydrosystems problems have never been presented in one book, but instead have only been presented in various locations in the literature.

The term **hydrosystems** was originally coined by V. T. Chow to collectively describe the technical areas of hydrology, hydraulics, and water resources. Hydrosystem has also been a term used for reference to types of water projects including groundwater systems, surface water storage systems, water distribution systems, flood control systems, drainage systems, etc. Hydrosystems as used in this book, actually applies to both definitions. Specifically the types of hydrosystems in

LIBRARY

this book include river-reservoir systems, groundwater systems, bay and estuary systems, and water distribution systems. Operation for reservoir systems include both the long-term operation for water supply, sediment control, and freshwater inflow to bays and estuaries and short term operation for flood control and sediment control.

The book is divided into three major parts: Principles and Methodologies for Optimal Control Using Mathematical Programming Approach, Mathematical Programming Application, and Differential Dynamic Programming Application. The first part on Principles and Methodologies for Optimal Control has chapters that introduce hydrosystems control problems as discrete-time optimal control problem, introduce system and optimal control concepts, and nonlinear programming concepts. The second part of the book has four chapters that apply the optimal control concepts using mathematical programming to develop models and solution algorithms for groundwater systems operation, real-time operation of river-reservoir system for flood control, water distribution systems operation, and reservoir operation for optimizing freshwater inflows to bays and estuaries. The third part of the book has chapters on optimal control using differential dynamic programming, reservoir operation for water supply, groundwater systems operation, and reservoir operation for sediment control in rivers and reservoirs.

This book is written at an advanced level for those with some background in operations research, hydraulics, and water resources engineering. Both graduate students and practicing engineers will find this book to be a valuable reference book and text. The book can be used in graduate level water resources engineering courses. Intentionally this book is not a review of the literature but instead is an introduction to the concepts of optimal control theory and its applications to various types of hydrosystems using mathematical programming techniques and differential dynamic programming. A major focus is to illustrate how hydraulic

simulators can be interfaced with optimizers in an optimal control framework to solve realistic, large-scale hydrosystems operation problems that are optimal control problems.

Much of the work presented in this book is based upon the research work of my former Ph.D. students. In particular Chapter 4 is based upon the research efforts of Dr. Nisai Wanahule; Chapter 5 is based upon the research efforts of Dr. Olcay Unver; Chapter 6 is based upon the research efforts of Dr. Lehar Brion; Chapter 7 is based upon the research efforts of Dr. Yixing Bao; and Chapter 11 is based upon the present research efforts of Mr. Carlos Carriaga at Arizona State University. Dr. Leon Lasdon, a friend and former colleague at the University of Texas at Austin, over the years, has been rather influential in teaching me and my former graduate students at the University of Texas many of the concepts presented in this book, particularly those related to the optimal control concepts using mathematical programming. I also need to thank Dr. Lasdon for his willingness to provide us versions of his GRG2 code and his friendly advice on its use over the years. Other former graduate students of mine at Texas also have helped in the development of many of our concepts in solving the optimal control problems for hydrosystems including Dr. M. John Cullinane, Dr. Ning Duan, Dr. Joong-Hoon Kim, Dr. Kevin Lansey, Dr. Jungi Matsumato, Dr. Chang-Kang Taur and Dr. Yeou-Koung Tung.

This book is intended to be a contribution toward the eventual goal of better engineering and management practice in the hydrosystems field.

Larry W. Mays

○ = Reading assigned 10/13/92

OPTIMAL CONTROL FOR HYDROSYSTEMS
by Dr. Larry W. Mays

Chapter 1 Introduction to Hydrosystems Problems

1.1 Optimization of Hydrosystems
1.2 Groundwater Management Systems
1.3 Real-Time Operation of River-Reservoir Systems for Flood Control
1.4 Reservoir System Operation for Water Supply
1.5 Water Distribution System Operation
1.6 Freshwater Inflows to Bays and Estuaries
1.7 Sediment Control in Rivers and Reservoirs
1.8 General Problem Formulation

Chapter 2 Systems Theory and Optimal Control

2.1 Introduction
2.2 Concepts in Systems
2.3 Definition of Optimal Control Problems
2.4 Maximum Principle
2.5 Continuous Dynamic Systems
2.6 Optimal Control Problem for Multi-Reservoir Operation

Chapter 3 Optimal Control Using Nonlinear Optimization Methods

3.1 Matrix Algebra for Nonlinear Programming
3.2 Unconstrained Nonlinear Optimization
3.3 Constrained Optimization: Optimality Conditions
3.4 Generalized Reduced Gradient(GRG) Method
3.5 Constrained Nonlinear Optimization: Penalty Function Methods
3.6 Augmented Lagrangian Method
3.7 Optimal Control via Unconstrained Optimization
3.8 Solution of Discrete-Time Optimal Control Problems

Chapter 4 Groundwater Management Systems

4.1 Hydraulic Management
4.2 Problem Formulation
4.3 Problem Solution
4.4 Example Application
4.5 Groundwater Remediation Models

Chapter 5 Real-Time Operation of River-Reservoir Systems for Flood Control

5.1 Problem Indentification
5.2 Problem Formulation
5.3 Problem Solution
5.4 Example Application

Chapter 6 Water Distribution System Operation

6.1 Problem Identification
6.2 Problem Formulation
6.3 Problem Solution
6.4 Example Application

Chapter 7 Freshwater Inflows to Bays and Estuaries

7.1 Problem Identification
7.2 Problem Formulation
7.3 Problem Solution
7.4 Example Application

Chapter 8 Optimal Control by Feedback Control Methods

8.1 Dynamic Programming
8.2 Feedback Control for Linear Systems
8.3 Groundwater Management
8.4 Feedback Control for Nonlinear Systems
8.5 LQP

Chapter 9 Differential Dynamic Programming (DDP)

9.1 Differential Dynamic Programming
9.2 Differential Dynamic Programming with Constraints
9.3 Multi-Reservoir Operation
9.4 Non LQP Groundwater Management
9.5 Groundwater Remediation

Chapter 10 Linear Quadratic Gaussian Control

10.1 LQG Method
10.2 Real-Time Operation of Reservoir Systems
10.3 Operation of Multi-Reservoir Systems During Floods
10.4 Real-Time Control of Large Hydropower Resevoir Systems

Chapter 11 Sediment Control in Rivers and Reservoirs

11.1 Problem Identification
11.2 Problem Formulation
11.3 Problem Solution
11.4 Example Application

Chapter 12 Future

CHAPTER 1 - INTRODUCTION

1.1 Optimization of Hydrosystems

Many problems for the operation of hydrosystems can be formulated in a general optimization framework in terms of **state** (or dependent) variables (**x**) and **control** (or independent) variables (**u**)

$$\text{Minimize } f(\mathbf{x}, \mathbf{u}) \tag{1.1.1}$$

subject to process simulation equations

$$\mathbf{G}(\mathbf{x}, \mathbf{u}) = \mathbf{0} \tag{1.1.2}$$

and additional constraints for operation on the dependent (**u**) and independent (**x**) variables

$$\underline{\mathbf{w}} \leq \mathbf{w}(\mathbf{x}, \mathbf{u}) \leq \overline{\mathbf{w}} \tag{1.1.3}$$

The **process simulation equations** for hydrosystems applications basically consist of the governing physical equations (1.1.2) that simulate a physical process such as conservation of mass, energy and momentum. These equations are typically large in number, sparse and nonlinear in terms of the state and control variables. In most hydrosystem applications, these governing equations are ordinary or partial differential equations. Conceptually, the simplest approach is to have the optimizer directly solve the above optimization problem by embedding finite differences or finite element equations for the governing process equations. Unfortunately, many of the real-world problems cannot be solved in this manner as a result of

their size and nonlinearity. The existing nonlinear programming (NLP) codes cannot solve such large, sparse problems.

An alternative approach is to use the appropriate process simulator to solve the constraints process simulation equations (1.1.2) each time the constraints need to be evaluated for the optimizer. The major advantage of such an approach is the reduced size of problem seen by the nonlinear optimizer so that only a small subset of the complete set of constraint equations is evaluated by the optimizer. The basic idea is that the optimizer only sees the following **reduced problem** :

$$\text{Minimize } F(\mathbf{u}) = f(\mathbf{x}(\mathbf{u}), \mathbf{u}) \tag{1.1.4}$$

$$\text{subject to } \underline{\mathbf{w}} \leq \mathbf{w}(\mathbf{x}(\mathbf{u})) \leq \overline{\mathbf{w}} \tag{1.1.5}$$

as opposed to the much larger problem defined by equations (1.1.1) - (1.1.3).

The class of problems that are being considered in this book essentially have differential equations as part of the constraint set (process simulation equation) making them more complex than the standard type of optimization problem. These optimization problems are referred to herein as **optimal control problems**. Examples of hydrosystem optimal control problems are presented in sections 1.2 - 1.7. Each of these are nonlinear programming problems that can be solved by interfacing the appropriate simulator (simulation model) with the optimizer to solve a reduced nonlinear programming problem. Application to systems such as groundwater systems (1.1.1), river-reservoir systems (1.1.2) for flood control, reservoir systems (1.1.3) for water supply, water distribution systems (1.1.4) operation, and estuary systems (1.1.5) for salinity control.

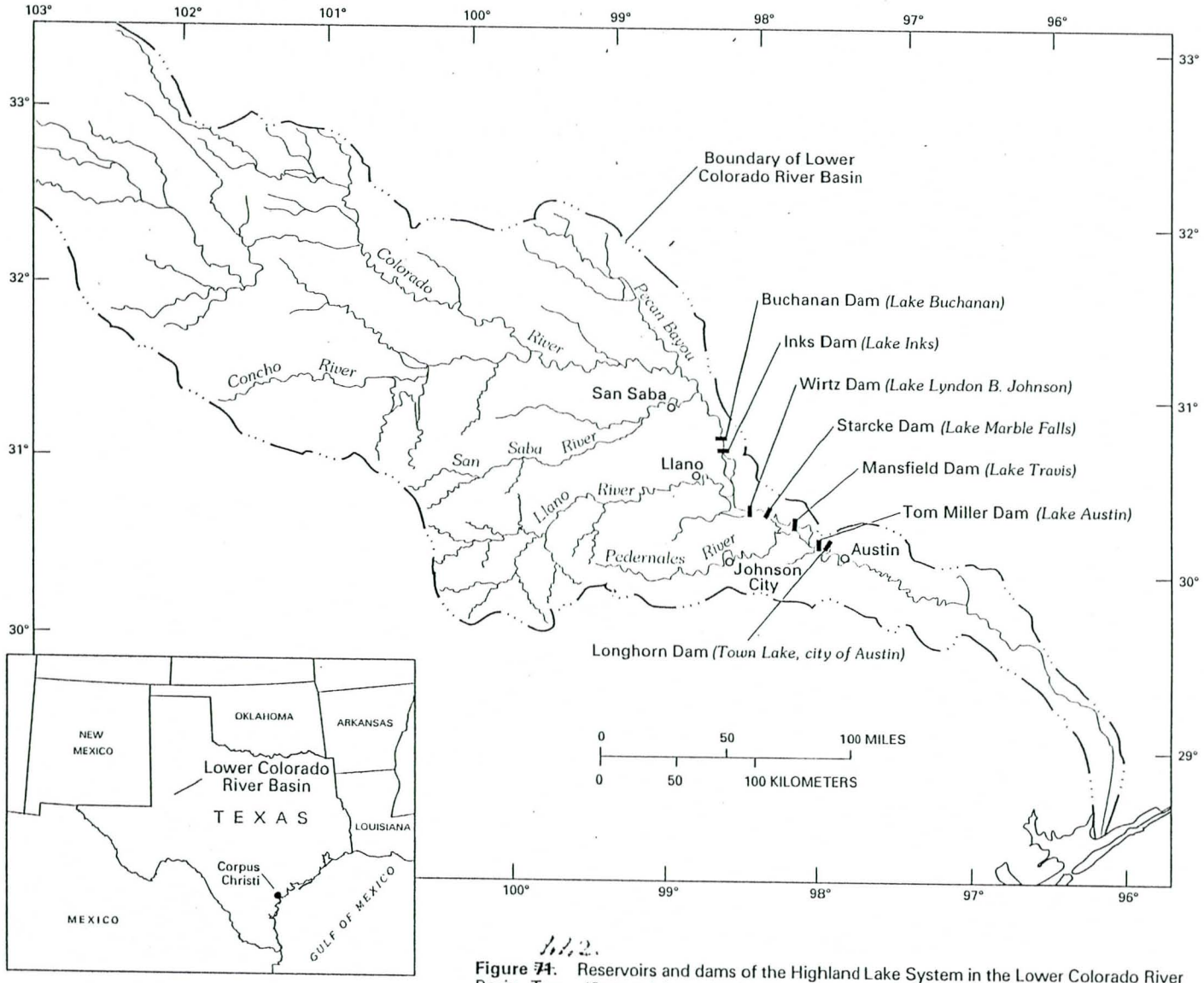


Figure 74. Reservoirs and dams of the Highland Lake System in the Lower Colorado River Basin, Tex. (Source: Information from Lower Colorado River Authority.)

SRPSIM
Schematic Diagram

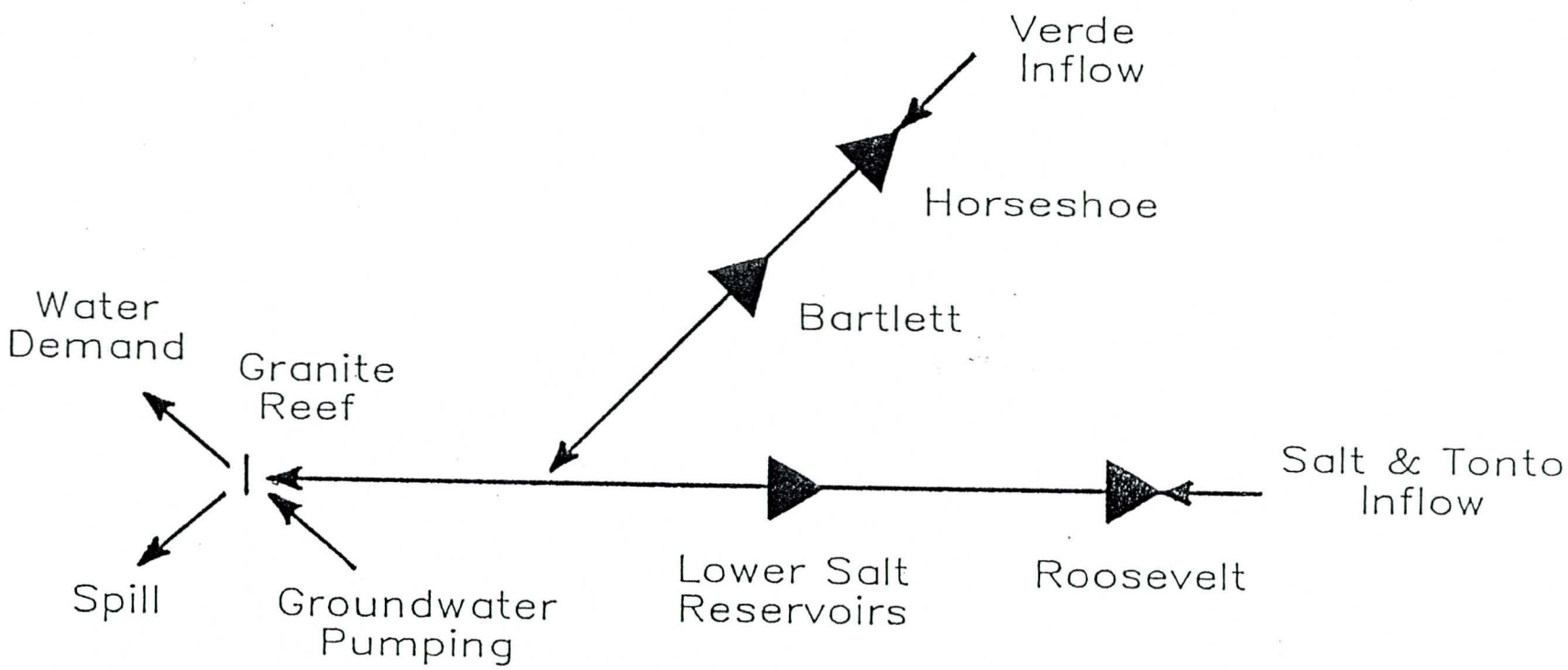


Figure 1.1.3 Salt River Project Reservoir System

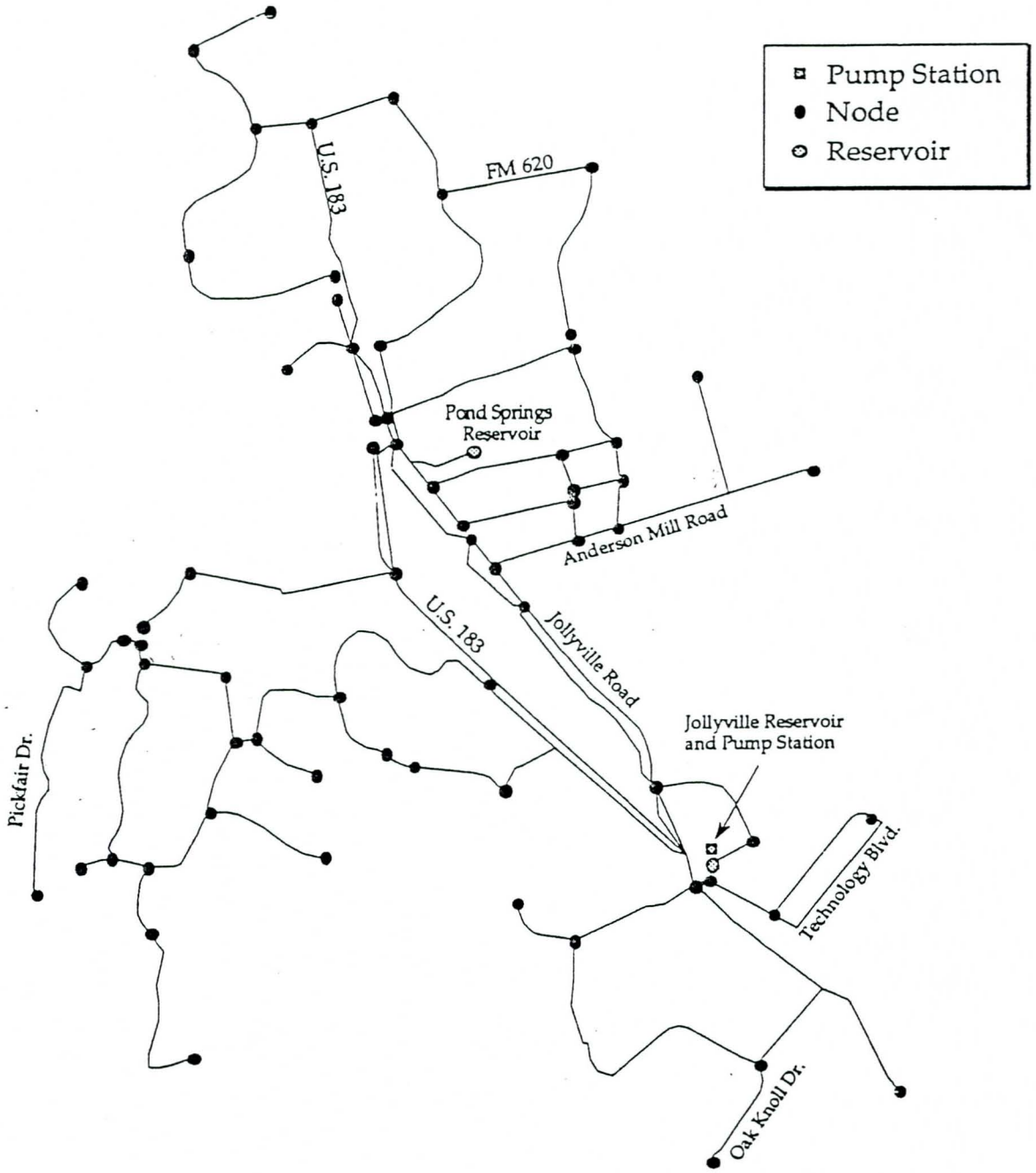


FIG. ~~A.~~ 1.1.4 Water Distribution System for City of Austin, Northwest B Pressure Zone.

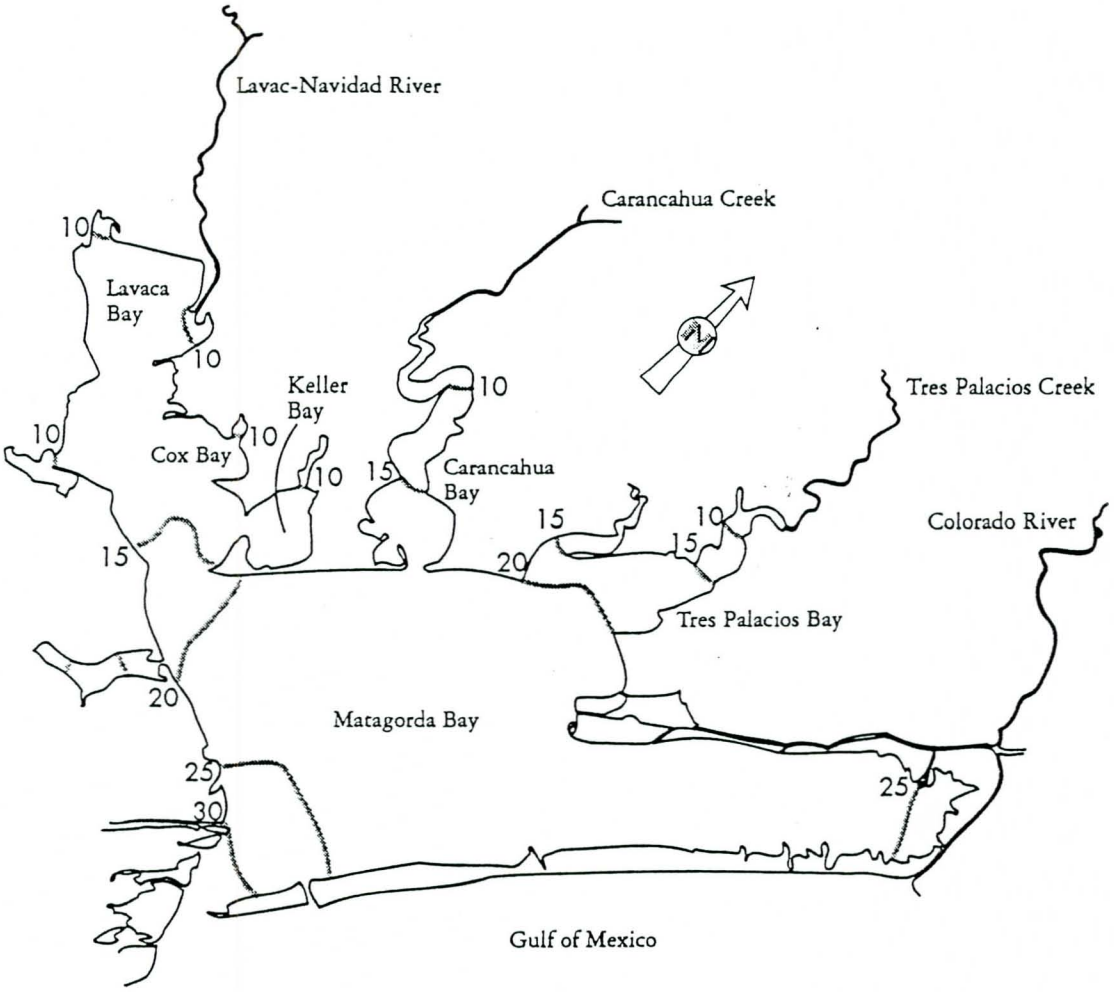


Figure ~~34~~ 1.1.5 Bay System Map and Example of Simulated Salinity Contour (ppt) in Lavaca-Tres Palacio Estuary

1.2 Groundwater Management Systems

The general groundwater management problem (GGMP) can be expressed mathematically as follows :

Objective

$$\text{Optimize } Z = f(\mathbf{h}, \mathbf{q}) \tag{1.2.1}$$

where **h** and **q** are vectors of heads and pumpages (or recharge), respectively. The objective function may be either maximization (e.g., sum of heads) or minimization (e.g., minimize pumpage), and can be a linear or nonlinear function. Also, it may be nonseparable or contain only terms of pumpages or heads.

Constraints

a. The general groundwater flow constraints represent a system of equations governing ground-water flow which are finite difference or simulator equations when **q** is unknown.

$$\mathbf{G}(\mathbf{h}, \mathbf{q}) = \mathbf{0} \tag{1.2.2}$$

b. The upper ($\overline{\mathbf{q}}$) and lower ($\underline{\mathbf{q}}$) bounds on pumpages physically may or may not exist. Unlike pumpage, the lower bound on heads ($\underline{\mathbf{h}}$) can be viewed as the bottom elevation of the aquifer while the upper head bound ($\overline{\mathbf{h}}$) can be viewed as ground surface elevations for the unconfined cells.

$$\underline{\mathbf{q}} \leq \mathbf{q} \leq \overline{\mathbf{q}} \tag{1.2.3}$$

$$\underline{\mathbf{h}} \leq \mathbf{h} \leq \overline{\mathbf{h}} \tag{1.2.4}$$

c. In addition to constraint Eqs. (1.2.2) - (1.2.4), other constraints may be included to impose restrictions such as water demands, operating rules, budgetary limitations, etc.

$$\mathbf{w}(\mathbf{h}, \mathbf{u}) \leq 0 \tag{1.2.5}$$

Dimension = (# Nodes) • (# Time Steps)

Both head, **h**, and pumpage (or recharge) **q** are vectors of decision variables which have maximum dimensions equal to the product of the number of active nodes within the aquifer boundary and time steps. Fixed pumpages or recharges are considered to be constants. By convention, available pumpages have a positive value and the elements of **q** have a negative value where there is available recharge. Usually the number of variable pumpages and/or recharges (thereafter the terms pumpages that refer to **q** will imply both pumpages and/or recharges) is small and results in a much smaller dimension of **q** ~~and~~ than **h**.

1.3 Real-Time Operation of River-Reservoir Systems for Flood Control

The optimization problem for the real time operation of multireservoir systems under flooding conditions can be stated as follows:

Objective

$$\text{Minimize } Z = f(\mathbf{h}, \mathbf{Q}) \tag{1.3.1}$$

where **h** and **Q** are the vectors of water surface elevations and discharges, respectively. The objective is defined by minimizing: (a) the total flood damages; (b) deviations from target levels; (c) water surface elevations in the flood areas; or (d) spills from reservoirs or maximizing storage in reservoirs.

Constraints

a. Hydraulic constraints are defined by the Saint-Venant equations for one-dimensional gradually varied unsteady flow and other relationships such as upstream, downstream, and internal boundary conditions and initial conditions that describe the flow in the different components of a river-reservoir system,

$$\mathbf{G}(\mathbf{h}, \mathbf{Q}, \mathbf{r}) = 0 \tag{1.3.2}$$

where **h** is the matrix of water surface elevations; **Q** is the matrix of discharges; and **r** is the matrix of gate settings for spillway structures, all given in matrix form to consider the time and space dimensions of the problem.

b. Bounds on discharges defined by minimum and maximum allowable reservoir releases and flow rates at specified locations,

$$\underline{\mathbf{Q}} \leq \mathbf{Q} \leq \overline{\mathbf{Q}} \tag{1.3.3}$$

Bars above and below a variable denote the upper and lower bounds, respectively, for that variable.

c. Bounds on elevations defined by minimum and maximum allowable water surface elevations at specified locations (including reservoir levels).

$$\underline{\mathbf{h}} \leq \mathbf{h} \leq \overline{\mathbf{h}} \tag{1.3.4}$$

d. Physical and operational bounds on spillway gate operations.

$$\mathbf{0 \leq \underline{r} \leq r \leq \bar{r} \leq 1} \tag{1.3.5}$$

e. Other constraints such as operating rules, target storages, storage capacities, etc.

$$\mathbf{W(r) \leq 0} \tag{1.3.6}$$

The constraints of the model can be divided into two groups: the hydraulic constraints (Eqs. 1.3.2) and the operational constraints (Eqs. 1.3.3. - 1.3.6). The hydraulic constraints are equality constraints consisting of the equations that describe the flow in the system. These are: (a) the Saint-Venant equations for all computational reaches (except internal boundary reaches); (b) relationships to describe the upstream and downstream conditions for the extremities; and (c) internal boundary conditions which describe the flow that cannot be described by the Saint-Venant equations such as critical flow resulting from flow over a spillway or waterfall.

The operational constraints are basically greater-than or less-than type constraints that define the variable bounds, operational targets, structural limitations, and capacities. Options for an operator to set or limit the limits of certain variables are also classified under this category. Bound constraints are used to impose operational or optimization-related requirements. Nonnegativity constraints on discharges are not used because discharges are allowed to take negative values in order to be able to realistically represent the reverse flow phenomena (backwater effects) due to a rising lake or large tributaries into a lake or tidal condition. Nonnegativity of water surface elevations are ~~is~~ always satisfied since the system hydraulics are solved

implicitly by the simulation model, DWOPER. The lower limits on elevations and discharges can be used to indirectly impose water quality considerations, minimum required reservoir releases, and other policy requirements. The upper bound on bound on elevations and discharges can be used to set the maximum allowable levels (values beyond are either catastrophic or physically impossible) such as the overtopping elevations for major structures, spillway capacities, etc. Damaging elevations and/or discharges must be given to the model through the constraints, as the objective functions do not have any terms to control them.

The third model variable, gate opening, can be allowed to vary between zero and one, which corresponds to zero and one hundred percent opening of the available total spillway gate areas, respectively. The bounds on gate settings are intended primarily to reflect the limitations on gate operations as well as to enable the operator to prescribe any portion(s) of the operation for any reservoir(s). Operational constraints other than bounds can be imposed for various purposes. The maximum allowable rates of change of gate openings, for instance, for a given reservoir, can be specified through this formulation, as a time-dependent constraint. This particular formulation may be very useful, especially for cases where sharp changes in gate operations, that is, sudden openings and closures, are not desirable or physically impossible. It is handled by setting an upper bound to the change in the percentage of gate opening from one time step to the next. This constraint can also be used to model another aspect of gate operations for very short time intervals, that is, the gradual settings that have to be followed when opening or closing a gate. For this case, the gate cannot be opened (or closed) by more than a certain percentage during a given time interval.

1.4 Reservoir System Operation for Water Supply

Reservoir system operation is for the purposes: to meet water supply demand; recreation demands; maintain minimum flow levels for navigation and environmental concerns; provide flood protection, power production, and flood control. The mathematical formulation of the reservoir system operation problem can be stated in general form as follows :

Objective

S = storage
U = releases
I = inflows
L = losses

$$\text{Maximize Benefits} = \text{Max} \sum_{0}^{T} f\,(S_t, U_t, t) \qquad (1.4.1)$$

Constraints

a) The system equations which are the conservation of mass equations for the reservoirs and river reaches are

$$G\,(S_{t+1}, S_t, U_t, I_t, L_t) = 0 \qquad t = 0, \ldots, T-1 \qquad (1.4.2)$$

where S_{t+1}, S_t are the vectors of reservoir storages at the beginning of time period t+1 and t respectively; U_t is the vector of the reservoir releases for M reservoirs during time t; I_t is the vector of hydrologic inputs (such as inflow to reservoirs); and L_t is the vector of reservoir losses.

b) The bound constraints on reservoir releases, U_t, are

$$\underline{U}_t \le U_t \le \overline{U}_t \qquad t = 1, \ldots, T \qquad (1.4.3)$$

where $\underline{U}_t$ and $\overline{U}_t$ are lower and upper bounds on the reservoir releases.

c) The bound constraints on reservoir storages are deterministically defined as

$$\underline{S}_t \le S_t \le \overline{S}_t \qquad t = 1, \ldots, T \qquad (1.4.4)$$

where $\underline{S}_t$ and $\overline{S}_t$ are the lower and upper bounds on storage or

d) The bound constraints on reservoir storage could be defined in probabilistic form as storage reliability constraints as

$$P[S_t \ge \underline{S}_t] \le \alpha_t^{min} \qquad t = 1, \ldots, T \qquad (1.4.5)$$

and

$$P[S_t \le \overline{S}_t] \le \alpha_t^{max} \qquad t = 1, \ldots, T \qquad (1.4.6)$$

where P [ ] denotes the probability and α_t^{min} and α_t^{max} represent the minimum and maximum reliability or tolerance levels on storage, respectively.

e) Other reservoir operational constraints are expressed as

$$\mathbf{w}(\mathbf{S}_t, \mathbf{U}_t) = \mathbf{0} \qquad (1.4.7)$$

1.5 Water Distribution System Operation

The optimization problem for water distribution system operation can be stated in terms of the nodal pressure heads, **H**, pipe flows, **Q** , tank water surface elevations, **E**, and pump operating times, **D**. The objective is to minimize energy costs

H - heads
Q - flows
D - time of pumping
E = tank water surface elevation

Objective

Minimize energy costs = $f(\mathbf{H}, \mathbf{Q}, \mathbf{D})$ (1.5.1)

Constraints

a. Conservation of flow and energy constraints

$$\mathbf{G(H, Q, D, E) = 0} \tag{1.5.2}$$

b. Pump operation constraints

$$\mathbf{w(E) = 0} \tag{1.5.3}$$

c. Nodal pressure head bounds

$$\underline{\mathbf{H}} \leq \mathbf{H} \leq \overline{\mathbf{H}} \tag{1.5.4}$$

d. Bounds on pump operating times

$$\underline{\mathbf{D}} \leq \mathbf{D} \leq \overline{\mathbf{D}} \tag{1.5.5}$$

e. Bounds on tank water surface elevation

$$\underline{\mathbf{E}} \leq \mathbf{E} \leq \overline{\mathbf{E}} \tag{1.5.6}$$

1.6 Freshwater Inflows to Bays and Estuaries

The overall optimization model can be stated in the following general nonlinear programming format using an objective to minimize freshwater inflows or to maximize harvest.

Objective

$$\text{Optimize } z = f(\mathbf{Q}, \mathbf{s}, \mathbf{H}) \tag{1.6.1}$$

The general mathematical model can consider the following objective functions:

(a) minimize the sum of freshwater inflows into the bay and estuary over an operational time frame, such as a year;

(b) maximize the harvest over an operational time frame, such as a year; and,

(c) multiobjective to minimize freshwater inflows and maximize the harvest over an operational time frame, such as a year.

Constraints

a. Hydrodynamic transport equations that relate salinity, **s**, at a given point in an estuary to inflow, **Q**,

$$\mathbf{G}(\mathbf{Q}, \mathbf{s}) = \mathbf{0} \tag{1.6.2}$$

where **Q** is the independent variable (control variable) as a function of time and **s** is the dependent variable (state variable) as a function of time and location;

b. Regression equations that relate inflow to fish harvest

$$\mathbf{h}(\mathbf{Q}, \mathbf{H}) = \mathbf{0} \tag{1.6.3}$$

c. Constraints that define limitations on freshwater inflows due to upstream demands and water uses, and historical ranges

$$\underline{Q} \leq Q \leq \overline{Q} \tag{1.6.4}$$

d. constraints that define limitations on salinity.

$$\underline{s} \leq s \leq \overline{s} \tag{1.6.5}$$

1.8 General Problem Formulation

Each of the above optimization problems in Sections 1.2 - 1.7 can be written in the following general form:

Objective

$$\text{Optimize } z = f(\mathbf{x}, \mathbf{u}) \tag{1.8.1}$$

Constraints

$$\mathbf{x}_{t+1} = \mathbf{g}(\mathbf{x}_t, \mathbf{u}_t, t) \qquad t = 0, \ldots, T-1 \tag{1.8.2}$$

$$\underline{\mathbf{x}}_t \leq \mathbf{x}_t \leq \overline{\mathbf{x}}_t \qquad t = 0, \ldots, T \tag{1.8.3}$$

$$\underline{\mathbf{u}}_t \leq \mathbf{u}_t \leq \overline{\mathbf{u}}_t \qquad t = 0, \ldots, T-1 \tag{1.8.4}$$

where $\mathbf{x}_t$ is a column vector of dependent (state) variables at time t; $\mathbf{u}_t$ is the column vector of independent (control) variables at time t; $\underline{\mathbf{x}}_t$ and $\underline{\mathbf{u}}_t$ are column vectors of lower bounds; and $\overline{\mathbf{x}}_t$ and $\overline{\mathbf{u}}_t$ are column vectors of upper bounds. The objective function is assumed to be continuously differentiable in $(\mathbf{x}_t, \mathbf{u}_t)$. Time t can take only a finite number of discrete values.

The above optimization problem defined by equations (1.8.1) - (1.8.4) is a **discrete-time optimal control problem**. Note that in each of the different hydro- systems problems in sections 1.2 - 1.7 there is a set of hydraulic process (simulation) equations **G ( ) = 0**. These process simulation equations define the physics of the problem, that is, the governing physical equations that simulate the physical process. These are the conservation of mass, conservation of energy, and/or conservation of momentum.

Groundwater Management

In the case of the groundwater management model, equation 1.2.2, **G(h,q) = 0** is the set of general groundwater flow equations. For nonsteady state groundwater flow the governing physical equations for two-dimensional flow are

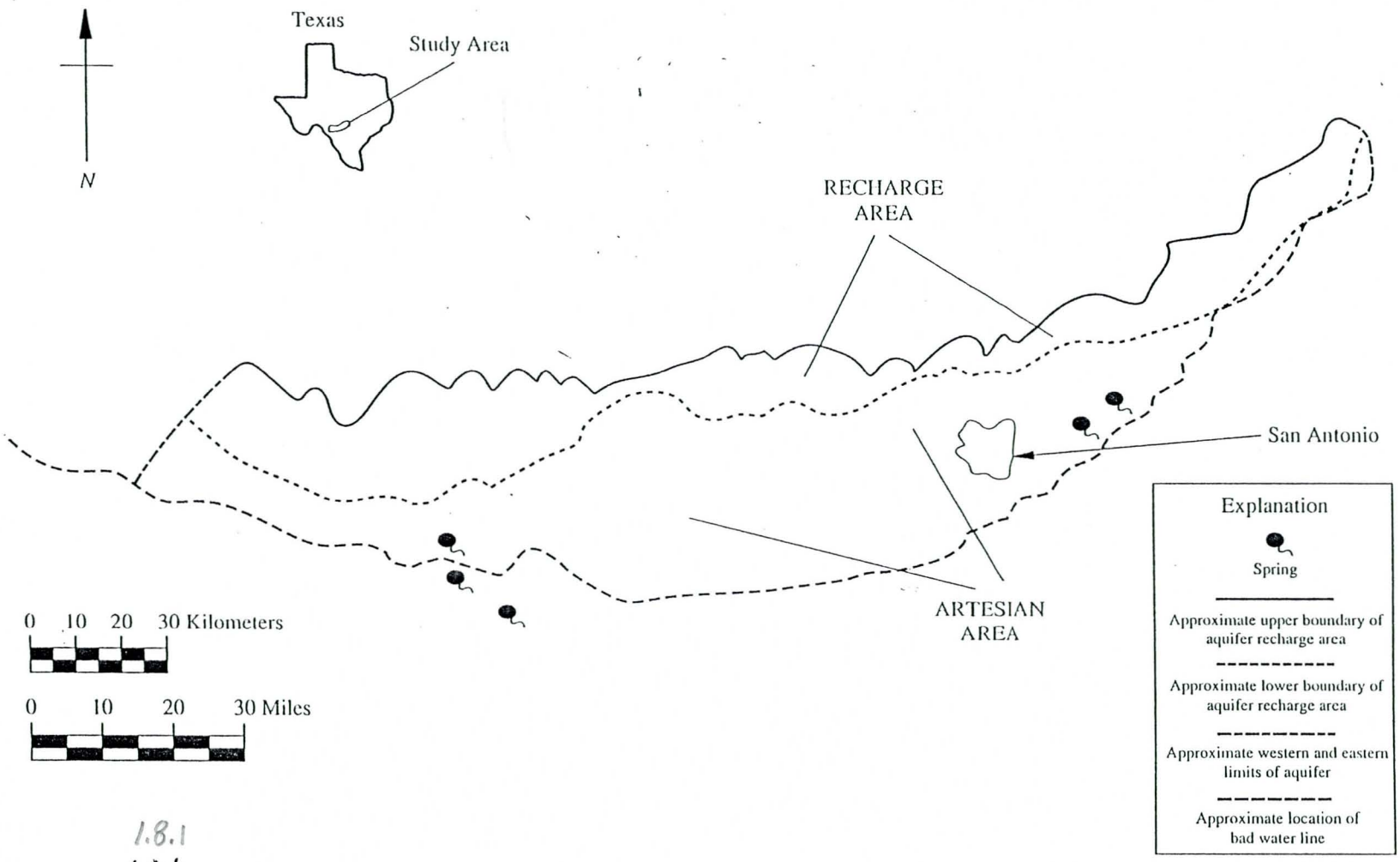
$$\frac{\partial}{\partial x_i}\left(T_{ij}\frac{\partial h}{\partial x_i}\right) = S\frac{\partial h}{\partial t} + W \qquad i, j = 1,2 \qquad (1.8.5)$$

where T_{ij} is the transmissivity vector; h is the hydraulic head; W is the volume flux per unit area; S is the storage coefficient; x_i, x_j are Cartesian coordinates; and t is time. The above partial differential equations can be written in a finite difference form

$$\mathbf{G}\,(\mathbf{h}_{t+1}, \mathbf{h}_t, \mathbf{q}_{t+1}) = \mathbf{0} \qquad t=0, \ldots, T-1 \qquad (1.8.6)$$

letting the volume flux to be replaced by the pumpage or recharge q. Alternatively equation (1.8.6) can be written as

$$\mathbf{h}_{t+1} = \mathbf{g}\,(\mathbf{h}_t, \mathbf{q}_{t+1}, t) \qquad t=0, \ldots, T-1 \qquad (1.8.7)$$



1.8.1

FIGURE ~~1.2.5~~

Edwards (Balcones fault zone) aquifer, San Antonio region. (After Guyton and Associates, 1979)

$x_{t+1} = g(x_t, u_t, t)$

which is in the form of equation (1.8.2). In this case the state variable is the hydraulic head, $\mathbf{h_t}$, and the control variable is the pumpage or recharge $\mathbf{q_t}$. The finite difference cell map for an aquifer is shown in Figure 1.8.1.

Real-Time Operation of River-Reservoir System for Flood Control

In the case for the real-time operation of river-reservoir systems for flood control, the set of governing physical equations, $\mathbf{G(h,Q,r) = 0}$ are the Saint-Venant equations for one-dimensional unsteady flow,

Continuity:

$$\frac{\partial Q}{\partial t} + \frac{\partial (A + A_0)}{\partial t} - q = 0 \tag{1.8.7}$$

Momentum:

not defined

$$\frac{\partial Q}{\partial t} + \frac{\partial(\beta Q^2 / A)}{\partial x} + gA\left(\frac{\partial h}{\partial x} + S_f + S_e\right) - \beta w v_x + W_f B = 0 \tag{1.8.8}$$

where

x = longitudinal distance along the channel or river

t = time

A = cross-sectional area of flow

A_O = cross-sectional area of off-channel dead storage (contributes to continuity, but not momentum)

q = lateral inflow per unit length along the channel

h = water surface elevation

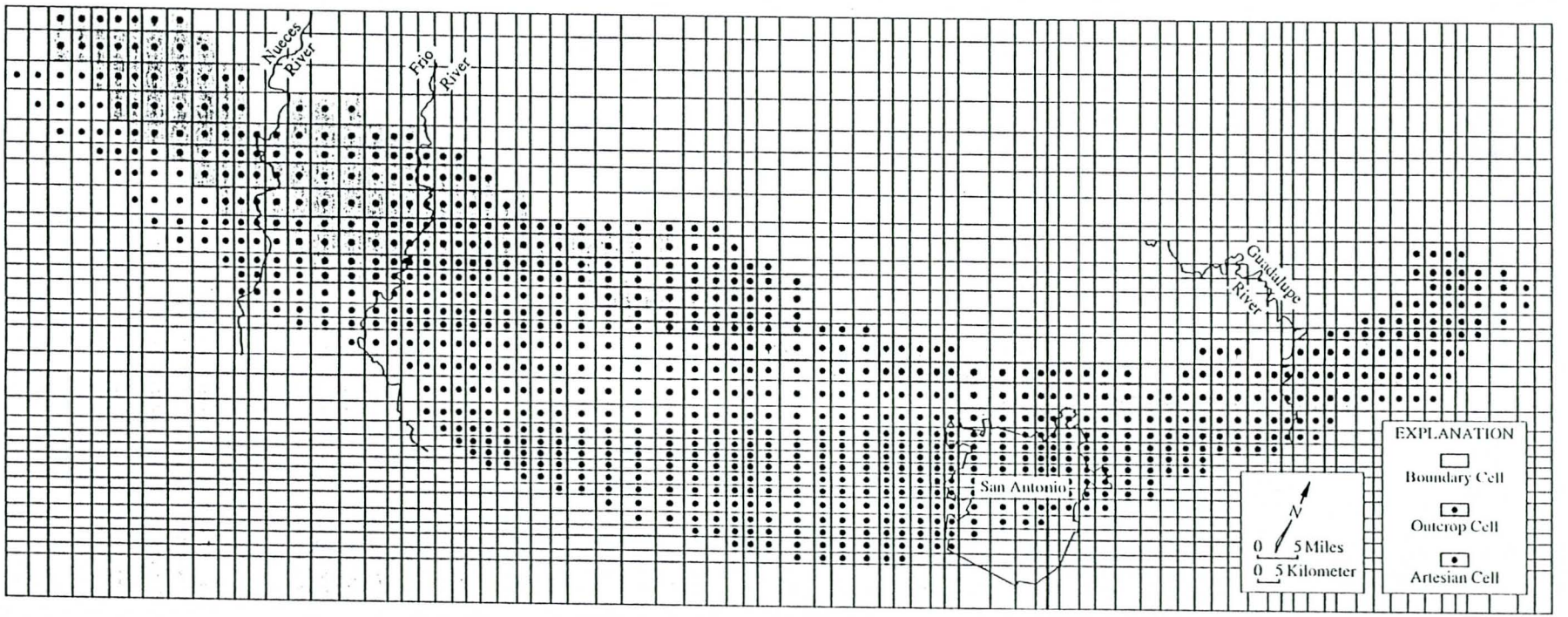


FIGURE ~~8.2.2~~ 1.8.1
Cell map used for the digital computer model of the Edwards (Balcones fault zone) aquifer (after Klemt et al., 1979).

v_x = velocity of lateral flow in the direction of channel flow

S_f = friction slope

S_e = eddy loss slope

B = width of the channel at the water surface

W_f = wind shear force

β = momentum correction factor

g = acceleration due to gravity.

The above set of partial differential equations can be expressed respectively in general form with the continuity and momentum, respectively as

$$G_C\,(\mathbf{h}_t, \mathbf{h}_{t+1}, \mathbf{Q}_t, \mathbf{Q}_{t+1}, \mathbf{r}_{t+1}) = 0 \tag{1.8.9}$$

$$G_M\,(\mathbf{h}_t, \mathbf{h}_{t+1}, \mathbf{Q}_t, \mathbf{Q}_{t+1}, \mathbf{r}_{t+1}) = 0 \tag{1.8.10}$$

or respectively as

$$\mathbf{h}_{t+1} = \mathbf{g}_C\,(\mathbf{h}_t, \mathbf{Q}_t, \mathbf{Q}_{t+1}, \mathbf{r}_{t+1}) \tag{1.8.11}$$

$$\mathbf{Q}_{t+1} = \mathbf{g}_M\,(\mathbf{h}_t, \mathbf{Q}_t, \mathbf{Q}_{t+1}, \mathbf{r}_{t+1}) \tag{1.8.12}$$

The state variables in this problem are the water surface elevations, **h**, and the discharge, **Q**. The control variable is the gate setting (spillway operation), **r**.

Reservoir System Operation for Water Supply

The system equations basically describe the dynamics of a reservoir system which is a configuration of reservoir whose coordinated operation is a function of hydrologic condition and/or institutional requirements. The dynamics of a particular reservoir j is represented by the conservation of mass

$$\frac{dS_{jt}}{dt} = I_{j,t} - U_{j,t} - L_{j,t} \tag{1.8.13}$$

The dynamics of the reservoir system can be described in a vector differential equation form as :

$$\dot{S}_t = F(S_t, t) + BU_t + CI_t \tag{1.8.14}$$

where S_t is an n_s - dimensional state vector including all reservoir storage variables; U_t is the n_u - dimensional vector of controllable releases; I_t is the n_I - dimensional vector of hydrologic inputs; $F(S_t, t)$ is an n_s - dimensional time varying nonlinear function with storage dependant term as shown in the above dynamic equation for a single reservoir; B and C are $n_s * n_u$ and $n_s * n_u$ - dimensional permutation matrices associating and each control and input vector element with the pertinent differential equation.

The state vector describes the storage in the various reservoirs and other system elements such as river reaches throughout the system as a function of time. At a particular time t_k when the state vector is known and for a known or specified set of inputs I_t and release U_t over the time interval $t \leq [t_k, T]$, then the state trajectory $\{S_t, t \leq [t, T]\}$ can be computed by integrating the above equation. A state vector summarizes the knowledge or information from the system history prior to time t_k. This information is necessary to

compute (predict) the reservoir system's future resource to input and output sequences. The purpose of reservoir operation control model is to identify control scheduler (reservoir releases) which generate optimum (desirable) state trajectories (storage).

Freshwater Inflows to Bays and Estuaries

In the case of the optimization of freshwater inflow to estuaries the governing physical equations (1.6.2), **G(Q, s)** = 0 for a two-dimensional (plan) formulation (see Figure 1.8.2) are the vertical-mean equations of momentum, continuity, and salinity mass budget given respectively as

$$\frac{du}{dt} = -g\nabla h + \frac{g}{D}\int_{z_b}^{D}\frac{1}{\rho}\int_{z}^{D}\nabla\rho dz\, dz + \frac{\tau_s - \tau_b}{\rho D} \tag{1.8.13}$$

$$\frac{\partial h}{\partial t} + D\nabla \bullet u = 0 \tag{1.8.14}$$

$$\frac{ds}{dt} = \frac{\partial}{\partial x}E_x\frac{\partial s}{\partial x} + \frac{\partial}{\partial y}E_y\frac{\partial s}{\partial y} \tag{1.8.15}$$

where $d/dt \equiv \partial/\partial t + \mathbf{u} \bullet \nabla$ and all vectors are referred to the plane coordinates (j, j, 0). ijk

?

Here

$\mathbf{u} = u\mathbf{i} + v\mathbf{j}$	vertical-mean current
$D = h(t)$	total depth, a function of position (x,y)
τ_s, τ_b	horizontal stress at surface and bottom
s	vertical-mean salinity
E_x, E_y	horizontal "dispersion" coefficients

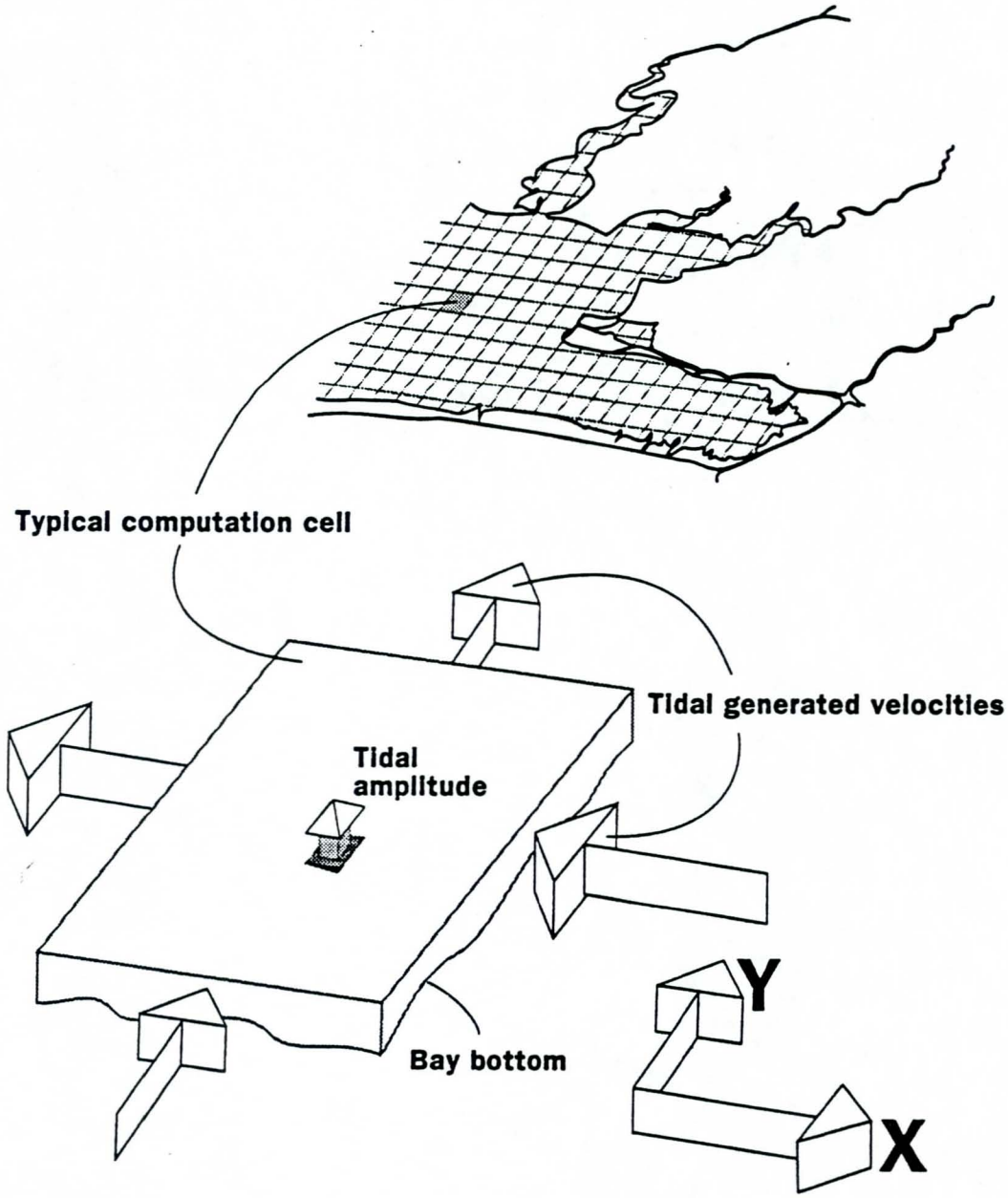


Figure ~~2.2~~ 1.8.2 Conceptual Illustration of Discretization of a Bay for Depth-Averaged Two-Dimensional Flow (Modified from: TDWR, 1980)

and density ρ and salinity s are related by an equation of state $\rho = \rho_0 + Ls$. The tidal-mean equations are exactly the same, except that the dependent variables h, **u**, s are now tidal-mean quantities (as well as vertical means) and the dispersion terms absorb the effective flux due to time correlation in s and **u**. The finite difference grid for the Lavaca-Tres Palacio Esturary along the Gulf of Mexico in Texas is shown in Figure 1.8.3.

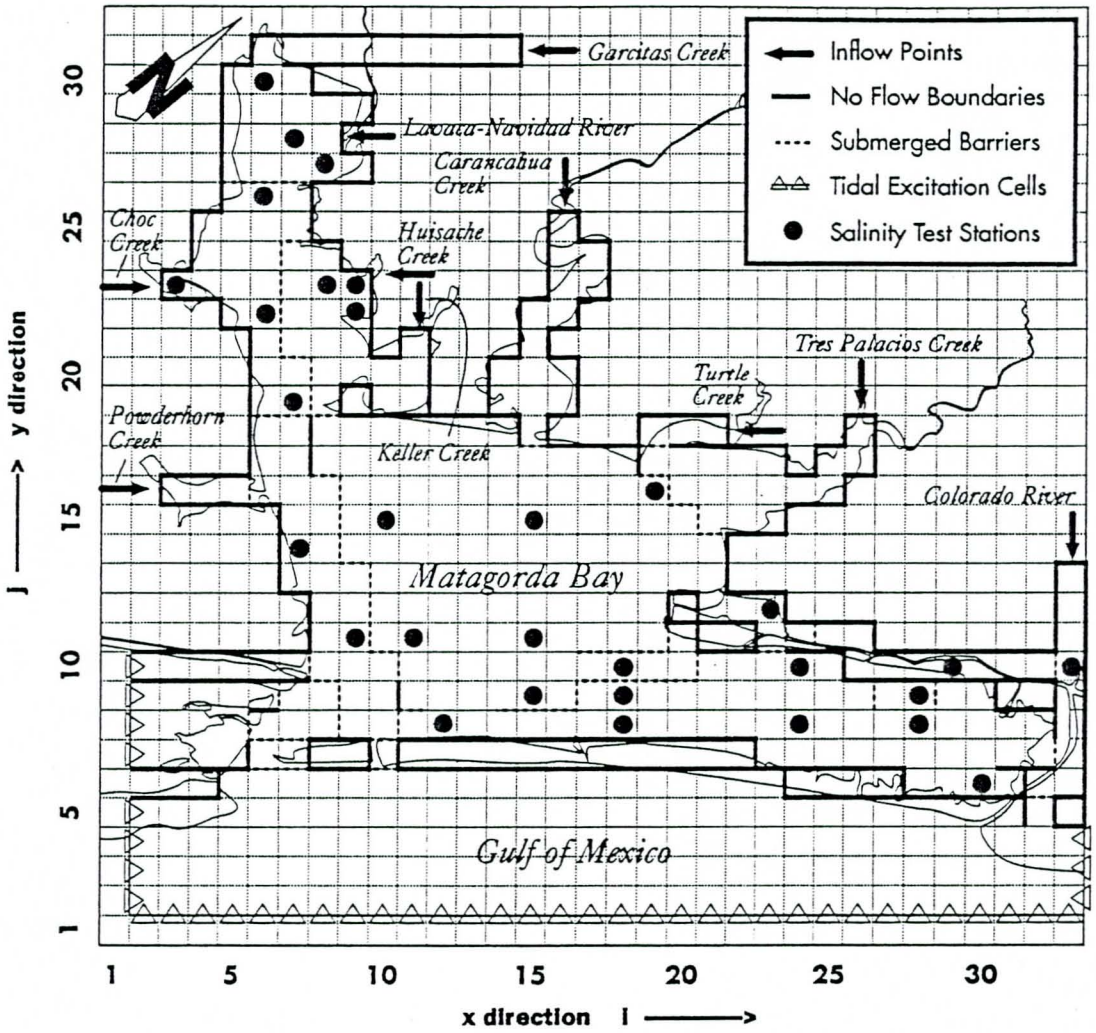


Figure ~~6.8~~ 1.8.3 Salinity Test Station Sites in Lavaca-Tres Palacio Estuary for OPTFOW

CHAPTER 2

SYSTEMS THEORY AND OPTIMAL CONTROL

2.2 Concepts of Systems

2.2.1 Concept of the State

In the "classical" system theory, the output is directly related to the input through a transfer function, ϕ.

y = output
u = input

$$\mathbf{y} = \phi \mathbf{u} \tag{2.2.1}$$

Differential equations are often used for the purpose of describing the transfer function. For a system to be amenable to state variable modeling analysis it must be lumped. This means that it must evolve in only one dimension such as time or space and be describable by ordinary differential or difference equations. Water resource systems are usually distributed and properly described by partial differential equations with respect to time and space. Since the nature of hydrosystems are inherently distributed, typically they are divided into several subsystems such that each individual subsystem is treated as a lumped system. It is possible, in many cases, to obtain a good approximation to distributed system behavior by using linked lumped systems.

In the so called "modern" system theory, the system structure is given explicit representation as a vector **x**, where $\mathbf{x} = (x_1, \ldots, x_n)$ and the state variables $x_1 \ldots, x_n$, are a function of time. The state of the system at any given time t_1, is given by the value of state variables $x_1(t_1), x_2(t_1), \ldots, x_n(t_1)$ which constitutes the state vector, $x(t_1)$. This is the fundamental concept of state variable modeling. In hydrosystems, the state variables are usually expressed in volumetric or mass units and can represent, for example, the volume of

x_1, x_2 } state variables

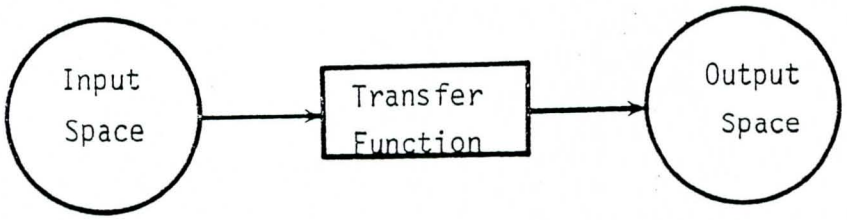
water or the amount of prescribed pollutants contained in various parts of the system. The input and output variables commonly correspond to volume or mass flow rates, which may be expressed as rainfall intensity or the rate of discharge of pollutant. The state of the system is a measure of the level of activity in each of its components and can be thought of as the interface between the past and the future of the system's time history.

Formally, the state vector may be defined as the minimum number of variables needed so that if the state at time t_1, $\mathbf{x}(t_1)$ is known and the input from time t1 to some later time t_2, $\mathbf{u}(t)$, $t_1 < t < t_2$ is also known, then the state, $\mathbf{x}(t_2)$, is completely determined from this information. Sometimes the state variable methodology is also called "state space" analysis.

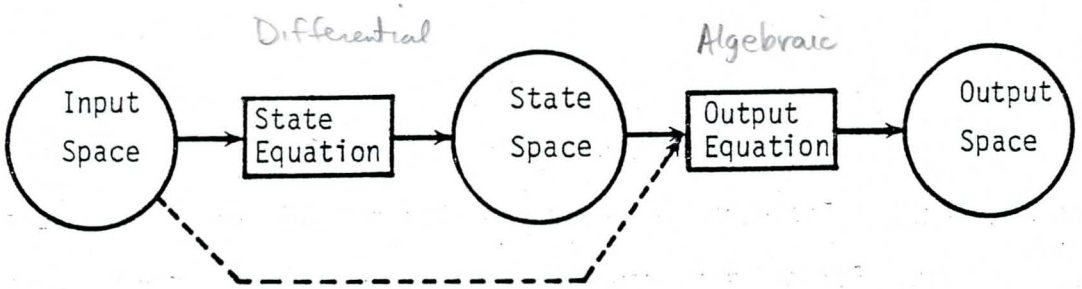
Comparison between the "classical" and "modern" approaches to dynamic systems modeling, shown in Figure 2.2.1, may be visualized in vector space mappings. If an "input space" for the input vector, **u**, and an "output space" for the output vector, **y**, are defined in the same way as the state space has been defined, the transfer function is seen as a mapping from the input space directly to the output space, Figure (2.2.1a). In the "modern" approach, Figure (2.2.1b), the input space is first related to the state space through the so-called "state equation" which is a differential or difference equation. Then the state space, and in some cases the input space, are related to the output space through the so-called "output-equation" which is algebraic.

Descriptions of continuous-time and discrete-time deterministic state variable models are given in the following two sections. Where vector-matrix operations are used, the notation employed is that a lower case letter, a, is a scalar; an underlined lower case letter, **a**, is a vector; and an upper case letter, A, is a matrix.

Figures on next 4 pages need to be done but only after complete text is done



(a) "Classical" Approach



(b) "Modern" Approach

Figure ~~2.1~~ 2.2.1 Comparison Between "Classical" and "Modern" Approachs to Dynamic System Modeling

2.2.2 Continuous-Time Deterministic State Variable Model

A deterministic model is one in which a given input always produces the same output. The continuous-time deterministic state variable model is mathematically formulated by means of two equations: the state equation and the output equation. The state equation is a set of ordinary, first-order differential equations, one for each state variable, which is written in vector-matrix form to simplify the notation. The state equation describes the change in the state of the system over time in response to the inputs. The output equation is a set of algebraic equations, one for each output variable relating the output to the state of the system and in some cases to the inputs. The output equation is also commonly written in vector-matrix form. For the most general case, the state and output equations may be expressed as in equations (2.2.2a) and (2.2.2b), respectively:

State Eqn

$$\dot{\mathbf{x}}(t) = \mathbf{g}\left[\mathbf{x}(t), \mathbf{u}(t), t\right] \tag{2.2.2a}$$

Output Eqn

$$\mathbf{y}(t) = \mathbf{h}\left[\mathbf{x}(t), \mathbf{u}(t), t\right] \tag{2.2.2b}$$

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \cdot \\ \cdot \\ \cdot \\ x_n(t) \end{bmatrix} ; \quad \dot{\mathbf{x}}(t) = \begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_n(t) \end{bmatrix} ; \quad \mathbf{y}(t) = \begin{bmatrix} y_1(t) \\ y_2(t) \\ \cdot \\ \cdot \\ \cdot \\ y_r(t) \end{bmatrix} ; \quad \mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \cdot \\ \cdot \\ \cdot \\ u_p(t) \end{bmatrix}$$

and

$$\dot{x}_1(t) = \frac{d\left[x_1(t)\right]}{dt}, \quad \dot{x}_2(t) = \frac{d\left[x_2(t)\right]}{dt}, \quad \dot{x}_n(t) = \frac{d\left[x_n(t)\right]}{dt}$$

The functions, g [ ], in equation (2.2.2a), and h [ ] in equation (2.2.2b) are nonlinear and time-variant. They are nonlinear because products or powers

of the variables may occur, and time-variant because the time, t, is included as an explicit variable.

For practical purposes, this model is usually simplified to the form shown in the following equations, which is the basic continuous-time, deterministic, state variable model

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.2.3}$$

$$y(t) = Cx(t) + Du(t) \tag{2.2.4}$$

where A, B, C, and D are the matrices:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \quad ; \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1p} \\ b_{21} & b_{22} & \cdots & b_{2p} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ b_{n1} & b_{n2} & \cdots & b_{np} \end{bmatrix} ;$$

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ c_{r1} & c_{r2} & \cdots & c_{m} \end{bmatrix} \quad ; \quad D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1p} \\ d_{21} & d_{22} & \cdots & d_{2p} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ d_{r1} & d_{r2} & \cdots & d_{rp} \end{bmatrix}$$

The system representation given the the above equations is shown schematically in Figure (2.2.2). From Figure (2.2.2), it may be seen that the time rate of change of the state of the system, $\dot{x}(t)$, is formed as the sum of the modified inputs, Bu(t), and the modified current state, Ax(t). The matrix A is the most important of the four system matrices because it represents the proportion of the current system state, x(t), which contributes to changing that state. This state feedback has a major role in determining the future behavior of the system. The elements of matrix B are scalars and represent the

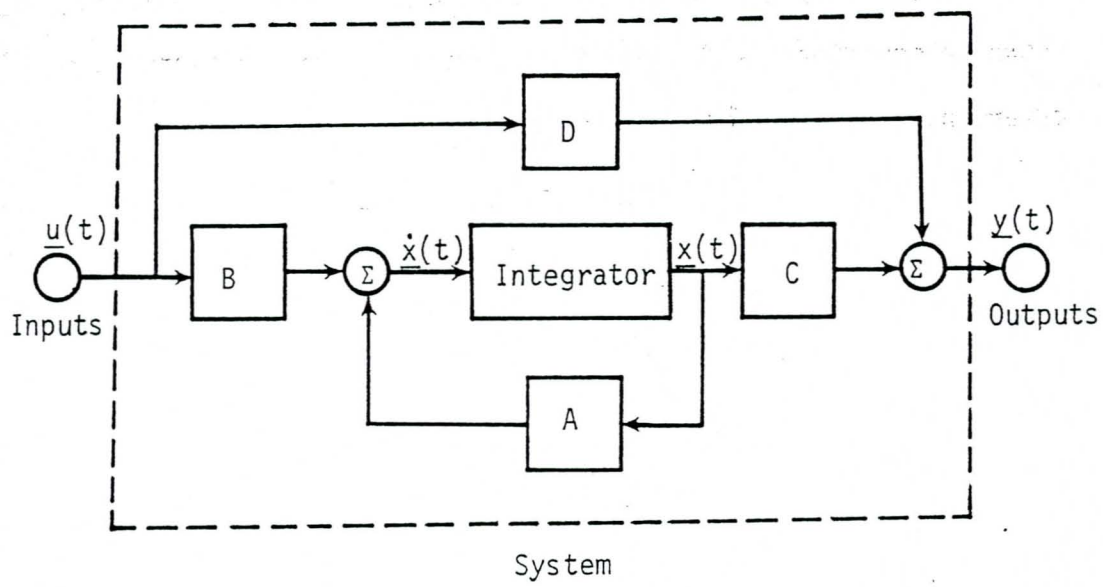


Figure ~~2.2~~ 2.2.2 Schematic Diagram of State Variable Model

proportion of the value of each of the input variables that affects each of the state variables.

The rate of change of the state, $\dot{\mathbf{x}}(t)$, is continuously integrated with the current state to produce the new state. The outputs, $\mathbf{y}(t)$, are formed by summing the new state which has been scaled by matrix C with a direct contribution from the modified input, $\mathbf{Du}(t)$. The elements of C and D are scalars which represent the proportions of each of the state and the input variables which produce the outputs, respectively.

The behavior of water resources systems often changes with time. For example, as the urbanization proceeds the proportion of the urbanized watershed area which is impervious increases causing the relationship between storm rainfall and to runoff changes. This time-variant behavior can be incorporated into state variable models by making some of the elements in the matrices A, B, C, and D functions of time. Non-linear response occurs when changes in the system's inputs do not produce linearly proportional changes in the system's outputs. These effects may be accounted for in state variable models, by formulating some of the elements in the four system matrices A, B, C, and D as functions of the current system state.

2.2.3 Discrete-Time Deterministic State Variable Model

Although the nature of water resources systems operate continuously in time, the data are often collected and analyzed using discrete-time intervals, especially when a digital computer is involved in the data storage and analysis. For this situation, it is advantageous to formulate a discrete-time version of the deterministic state variable model. To do this, the time horizon is divided into K intervals or stages, $k = 1, 2, ..., K$; of length Δt. Time

intervals, Δt, are not necessarily equal. The state, $\mathbf{x}(t + \Delta t)$ may be related to the state, $\mathbf{x}(t)$, at time, t by using Taylor's expansion:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + (\Delta t)\,\dot{\mathbf{x}}(t) + \frac{\Delta t^2}{2}\,\ddot{\mathbf{x}}(t) + \ldots \tag{2.2.5}$$

where $\ddot{\mathbf{x}}(t) = \dfrac{d^2[\mathbf{x}(t)]}{dt^2}$. If the terms of order of $(\Delta t)^2$ and higher are neglected, equation (2.2.5) may be written as:

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \dot{\mathbf{x}}(t) \bullet \Delta t \tag{2.2.6}$$

and the output equation is

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \tag{2.2.7}$$

Basic Discrete-Time Deterministic State Variable Model

Equations (2.2.6) and (2.2.7) form the basic discrete-time, deterministic, state variable model. In the situation when the time intervals are equal and set to one unit of time, then $t = k\Delta t$ where k is the state index. The state equation (2.2.6) and output equation (2.2.7) can then be expressed as equation (2.2.8) and (2.2.9), respectively, by substituting $\mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$ in equation (2.2.3) for $\dot{\mathbf{x}}(t)$.

$$\mathbf{x}(k+1) = (\mathbf{A}+\mathbf{I})\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \tag{2.2.8}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) \tag{2.2.9}$$

where I is an identity matrix of rank n.

In the discrete time model, the input and output variables corresponding to the volume or mass of flow across the system boundaries in the unit time interval instead of being volume or mass flow rate as they are in the continous-time model.

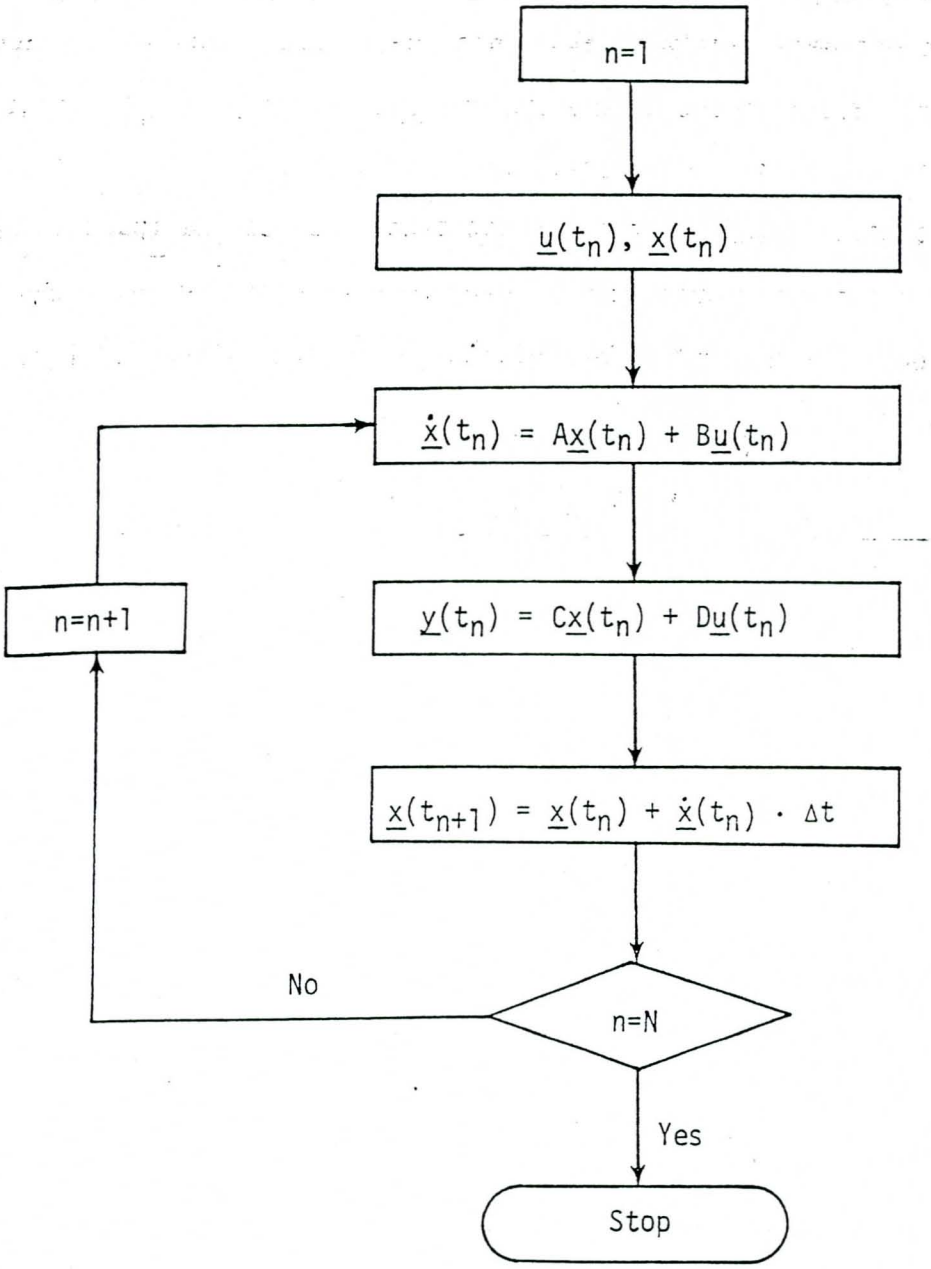
At first the input instruction to the system is divided into several stages with total number N. The time interval between two adjacent stages is not necessarily equal. Once the initial state of the system, $\mathbf{x}(t_0)$, and initial

input to the system, $\mathbf{u}(t_0)$, are given equations (2.2.6) and (2.2.7) may be solved to obtain the state, $\mathbf{x}(t_0 + \Delta t)$, at the next stage and the output, $\mathbf{y}(t_0)$, at the current stage. Since the input at each stage is known, the process can be performed recursively until the last stage is reached. The algorithm for solving the discrete-time model is illustrated as a flow diagram shown in Figure 2.2.3.

← 9:40 PM
10/14/92

2.2.4 Applications of State Variable Model in Water Resources

State variable modeling has been applied to only a few water resource systems. Fan et al. (1973) developed a model to find control strategies for biological waste treatment using a state variable model of a contineously stirred tank reactor. Young and Beck (1974) formulated a state variable model for dissolved oxygen and biochemical oxygen demand in a river. This model was used to determine control schemes for sewage effluent discharges to rivers. Erscheler et al. (1974) developed a control strategy for the operation of the penstrock inlet gates in a hydroelectric power station based on a state variable model of the system. State variable approaches have also been used to model the storm rainfall and runoff processes. Muzik (1974) used a state variable approach to model overland flow. Duong et al. (1975) applied stochastic estimation theory to fit the parameters of a state variable rainfall-runoff model. Maidment (1976) developed a stochastic state variable dynamic programming model for reservoir operation. Tung and Mays (1978) developed a kinematic wave model for sewer network flow routing based upon the state variable approach. Tung and Mays (1981) developed a rainfall-runoff model using the concepts of state variables modeling as described below.



2.2.3

Figure 2.3- Flow Chart for Discrete-Time State Variable Model

Because of the characteristics of a watershed, the system is inherently non-linear. The linear reservoir storage-discharge relation,

$$S = kQ \tag{2.2.10}$$

was modified by Prasad (1967) having the form

$$S = K_1 Q^N + K_2 \frac{dQ}{dt} \tag{2.2.11}$$

where k_1, k_2, and n are assumed to be constants. Combining equation (2.2.10) with equation (2.2.11) for the conservation of mass, the following differential equation is obtained:

$$K_2 \frac{d^2Q}{dt^2} + NK_1 Q^{N-1} \frac{dQ}{dt} + Q = I \tag{2.2.12}$$

Equation (2.2.11) is rearranged to

$$\frac{d^2Q}{dt^2} = -\frac{K_1}{K_2} NQ^{N-1} \frac{dQ}{dt} - \frac{1}{K_2} Q + \frac{1}{K_2} I \tag{2.2.13}$$

The simulation diagram of equation (2.2.13) is illustrated in Figure 2.2.4. The state variable formulation of equation the state equation in matrix form is written as follows:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -e_0 & -e_1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ h \end{bmatrix} I(t) \tag{2.2.14}$$

where $e_0 = \frac{1}{K_2}$, $e_1 = \frac{K_1}{K_2} NQ^{N-1}$, and $h = \frac{1}{K_2}$ and the output equation in matrix form is

$$Q(t) = [1 \quad 0] \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \tag{2.2.15}$$

To next page.

2.3 Definition of Optimal Control Problems

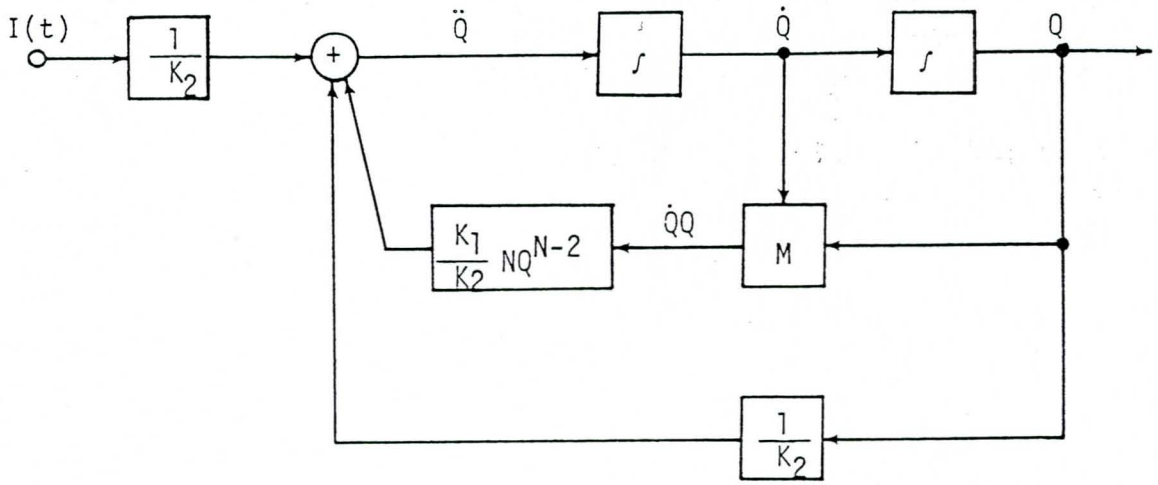


Figure ~~4.4~~ 2.2.4 Simulation Diagram of Nonlinear System Model

Optimization problems are typically thought of as problems with constraints that are algebraic equations and/or inequalities. Another important class of optimization problems are optimal control problems which have constraints that are differential equations, referred to as the state equations.

Optimal control problems can be stated as follows:

Given:

(1) state equations;

(2) a set of boundary conditions on the state variables at the initial time and the terminal time; and

(3) a set of constraints on the state variables and control variables;

Determine the **admissible control** (values of the control variable) so that an objective function (performance index) is optimized (minimized or maximized).

Mathematically the optimal control model in continuous form is to optimize the objective function

$$\text{Optimize } F(\mathbf{u}) = \int_0^T f\big(\mathbf{x}(t), \mathbf{u}(t), t\big)\, dt \tag{2.3.1}$$

subject to the state equations,

$$\dot{\mathbf{x}} = \mathbf{g}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \tag{2.3.2}$$

the set of boundary conditions

$$\underline{\mathbf{x}} \le \mathbf{x}(t) \le \overline{\mathbf{x}} \tag{2.3.3}$$

$$\underline{\mathbf{u}} \le \mathbf{u}(t) \le \overline{\mathbf{u}} \tag{2.3.4}$$

and the set of constraints on $\mathbf{x}(t)$ and $\mathbf{u}(t)$

$$\mathbf{w}\,(\mathbf{x}(t), \mathbf{u}(t)) = 0 \tag{2.3.5}$$

bold

where $\mathbf{u}(t)$ is the control variable $\mathbf{u} = [u_1, \ldots, u_m]^T$ with lower and upper bounds of $\underline{\mathbf{u}}$ and $\overline{\mathbf{u}}$, respectively; and $\mathbf{x}(t)$ is the state variable $x = (x_1, \ldots, x_n)$ with lower and upper bounds of $\mathbf{x}$ of $\overline{\mathbf{x}}$, respectively.

The objective function f is a given continuous real-valued function and the integral in equation (2.3.1) is interpreted as taking a control u(t) such that $\underline{\mathbf{u}} \leq \mathbf{u}(t) \leq \overline{\mathbf{u}}$; solving the state equation to obtain the corresponding x(t); then calculating fo as a function f time. The control variables and state variables are related through the state equation which is expressed as a differential equation in (2.3.2)

$$\frac{d\mathbf{x}}{dt} = \mathbf{g}\big(\mathbf{x}(t), \mathbf{u}(t)\big) \tag{2.3.6}$$

for a continuous system and as

$$\mathbf{x}(t+1) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) \tag{2.3.7}$$

for a discrete system.

In many applications of control theory the objective function (or performance index) has the form

$$F(\mathbf{u}) = \phi\left(\mathbf{x}(T)\right) + \int_0^T f\left(\mathbf{x}(t), \mathbf{u}(t)\right) dt \tag{2.3.8}$$

where T > 0 is fixed and $\phi\,(\mathbf{x}(\cancel{t}\,T))$ is a given continuously differentiable function that represents the terminal objective value at the final time T.

The optimal control problem in discrete form is to

Optimize $$F(\mathbf{u}) = \sum_{t=1}^{T} f\left(\mathbf{x}(t), \mathbf{u}(t)\right)$$

subject to the state equation in discrete form

$$\mathbf{x}(t+1) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t))$$

2.5 Continuous Systems

2.5.1 No Terminal Constraints, Fixed Terminal Time

Consider the optimal control problem to minimize (or maximize) the scalar performance index of the form of equation (2.3.8)

$$F(u) = \phi\left[\mathbf{x}(T), T\right] + \int_0^T f\left[\mathbf{x}(t), \mathbf{u}(t), t\right]dt \tag{2.5.1}$$

subject to the state equation (nonlinear differential equations)

$$\dot{\mathbf{x}} = \mathbf{g}\left[\mathbf{x}(t), \mathbf{u}(t), t\right] \qquad 0 \leq t \leq T \tag{2.5.2}$$

with $\mathbf{x}(0)$. $\mathbf{x}(t)$ is an n-vector function determined by $\mathbf{u}(t)$ an m-vector function.

The system differential equation (2.5.2) can be adjoined to the $F(\mathbf{u})$ using the multiplier function $\lambda(t)$

$$\overline{F}(\mathbf{u}) = \phi\left[\mathbf{x}(T), T\right] + \int_0^T \left\{ f\left[\mathbf{x}(t), \mathbf{u}(t), t\right] + \lambda^T(t)\left\{ \mathbf{g}\left[\mathbf{x}(t), \mathbf{u}(t), t\right] - \dot{\mathbf{x}} \right\} \right\} dt \tag{2.5.3}$$

A scalar function H referred to as the **Hamiltonian** is

$$H\left[\mathbf{x}(t), \mathbf{u}(t), \lambda(t), t\right] = f\left[\mathbf{x}(t), \mathbf{u}(t), t\right] + \lambda^T(t)\mathbf{g}\left[\mathbf{x}(t), \mathbf{u}(t), t\right] \tag{2.5.4}$$

Integrating the last term on the right side of (2.5.3) by parts yields

$$\overline{F} = \phi\left[\mathbf{x}(T), T\right] - \lambda^T(T)\,\mathbf{x}(T) + \lambda^T(0)\,\mathbf{x}(0) +$$

$$\int_0^T \left\{ H\left[\mathbf{x}(t), \mathbf{u}(t), t\right] + \lambda^T(t)\,\mathbf{x}(T) \right\} dt \tag{2.5.5}$$

The variation in F (δF) due to variations in the control vector $\mathbf{u}(t)$ for fixed times $t = 0$ and $t = T$ is

$$\delta\overline{F} = \left[\left(\frac{\partial\phi}{\partial x} - \lambda^T\right)\delta x\right]_{t=T} + \left[\lambda^T\delta x\right]_{t=0} +$$

$$\int_0^T\left[\left(\frac{\partial H}{\partial x} + \dot{\lambda}^T\right)\delta x + \frac{\partial H}{\partial \mathbf{u}}\delta \mathbf{u}\right]dt \tag{2.5.6}$$

To determine the variation $\delta x(t)$ produced by a given $\delta \mathbf{u}(t)$ would be difficult, so the multiplier function $\lambda(t)$ can be selected to cause the coefficients of δx in (2.5.6) to vanish

$$\dot{\lambda}^T = -\frac{\partial H}{\partial x} = -\frac{\partial f}{\partial x} - \lambda^T\frac{\partial g}{\partial x} \tag{2.5.7}$$

with the boundary conditions

$$\lambda^T(T) = \frac{\partial\phi}{\partial x(T)} \tag{2.5.8}$$

Equation (2.5.6) then becomes

$$d\overline{F} = \lambda^T(t=0)\,\delta x\,(t=0) + \int_0^T \frac{\partial H}{\partial \mathbf{u}}\delta \mathbf{u}\,dt \tag{2.5.9}$$

Thus $\lambda^T(t=0)$ is the gradient of F with respect to variations in the initial condition while holding $\mathbf{u}(t)$ constant and satisfying (2.5.1). For an extremum, δF must be zero for an arbitrary $d\mathbf{u}(t)$ which occurs when

$$\frac{\partial H}{\partial \mathbf{u}} = 0 \qquad 0 \le t \le T \tag{2.5.10}$$

Equations (2.5.7), (2.5.8), and (2.5.10) are known as the **Euler-Lagrange equations** in the calculus of variations.

The above functions $\lambda(t)$ are called **influence functions** on F of variations in $\mathbf{x}(t)$ since t is arbitrary. The function $\frac{\partial H}{\partial \mathbf{u}}$ are called **impulse response functions**. Since each component of $\frac{\partial H}{\partial \mathbf{u}}$ represents the variation in F due to a unit impulse (Dirac function) in the corresponding component of $d\mathbf{u}$ at time t, while holding $\mathbf{x}(t=0)$ constant and satisfying (2.5.1).

Summarizing, to determine the control vector function $\mathbf{u}(t)$, solve the following differential equations

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \tag{2.5.11}$$

$$\dot{\lambda} = -\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^T \lambda - \left(\frac{\partial f}{\partial \mathbf{x}}\right)^T \tag{2.5.12}$$

where $\mathbf{u}(t)$ is determined by

$$\frac{\partial H}{\partial \mathbf{u}} = \left(\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right)^T \lambda + \left(\frac{\partial f}{\partial \mathbf{u}}\right)^T = 0 \tag{2.5.13}$$

Boundary conditions for (2.5.11) and (2.5.12) are

$$\mathbf{x}(0) \text{ given} \tag{2.5.14}$$

and

$$\lambda(T) = \left(\frac{\partial \phi}{\partial \mathbf{x}}\right)^T \tag{2.5.15}$$

so that multistage system optimal control problems are two-point boundary value problems.

The first integral of the boundary-value problem exists if f and **g** are not explicit functions of t since

$$\dot{H} = H_t + H_x\dot{x} + H_u\dot{u} + \dot{\lambda}^T g$$

$$= H_t + H_u\dot{u} + \left(H_x + \dot{\lambda}^T\right)g \tag{2.5.16}$$

where H_t represents the partial derivative of H with respect to t holding **x** and **u** constant. According to equation (2.5.7), $\dot{\lambda}^T = -H_x$ so that $\left(H_x + \dot{\lambda}^T\right) = 0$, then

$$\dot{H} = H_t + H_u\dot{u} \tag{2.5.17}$$

If f and g (hence H) are not explicit functions of t and **u**(t) then $\dot{H} = 0$ or H is a constant on the optimal trajectory. For F to be a local minimum then $\frac{\partial H}{\partial u} = 0$ and the second-order expression for δF, holding that $\dot{x} - g = 0$, must be non-negative for all values (infinitesimal) of δu so that

$$\delta F = \frac{1}{2}\left[\delta x^T \frac{\partial^2 \phi}{\partial x^2}\delta x\right]_{t=T}$$

$$+ \frac{1}{2}\int_0^T \left[\delta x^T, \delta u^T\right]\begin{bmatrix} \frac{\partial^2 H}{\partial x^2} & \frac{\partial^2 H}{\partial x \partial u} \\ \frac{\partial^2 H}{\partial u \partial x} & \frac{\partial^2 H}{\partial u^2} \end{bmatrix}\begin{bmatrix} \delta x \\ \delta u \end{bmatrix} dt \geq 0 \tag{2.5.18}$$

where $\delta(\dot{x} - g) = 0$ or

$$\frac{d}{dt}(\delta x) = \frac{\partial g}{\partial x}\delta x + \frac{\partial g}{\partial u}\delta u \qquad \delta x(t=0) = 0 \tag{2.5.19}$$

The above equation (2.5.19) expresses $\delta x(t)$ in terms of $\delta \mathbf{u}(t)$ in a some what complicated fashion.

Consider the motion of a conservative system from time t = 0 to t = T such that

$$F = \int_0^T f(\mathbf{x}, \mathbf{u})\, dt \tag{2.5.20}$$

has a stationary value where $f = E_k(\mathbf{x}, \mathbf{u}) - E_p(\mathbf{x})$ such that E_k is the kinetic energy of the system and E_p is the potential energy of the system; $\mathbf{x}$ is the generalized coordinate vector defining the state of the system, and $\mathbf{u} = \dot{\mathbf{x}}$ is the generalized velocity vector. The Hamiltonian is then

$$H = f + \lambda^T \mathbf{u} \tag{2.5.21}$$

and the Euler-Lagrange equations are

$$\dot{\lambda}^T = -\frac{\partial H}{\partial \mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} \tag{2.5.22}$$

$$0 = -\frac{\partial H}{\partial \mathbf{u}} = \frac{\partial f}{\partial \mathbf{u}} + \lambda^T \tag{2.5.23}$$

Combining the above two vector equations

$$\frac{d}{dt}\left(\frac{\partial f}{\partial \dot{\mathbf{x}}}\right) - \frac{\partial f}{\partial \mathbf{x}} = 0 \tag{2.5.24}$$

which happens to be Lagrange's equation of motion for a conservative system. If f is not an explicit function of time, the first integral of the motion is H = constant:

$$H = f - \frac{\partial f}{\partial \mathbf{u}}\mathbf{u} = E_k - E_p - \frac{\partial E_k}{\partial \mathbf{u}}\mathbf{u} = \text{constant} \tag{2.5.25}$$

where $f = E_k - E_p$. E_k is a homgeneous quadratic form in $\mathbf{u}$, which is the velocity vector so that

$$\frac{\partial E_k}{\partial \mathbf{u}} \mathbf{u} = 2E_k \tag{2.5.26}$$

From equation (?)

$$H = E_k - E_p - 2E_k = \text{constant}$$

so that

$$-H = E_k + E_p = \text{constant} \tag{2.5.27}$$

which says that the kinetic energy plus the potential energy is constant during motion. The above argument for the motion of conservative system is referred to as Hamilton's principle in mechanics.

2.5.2 Function of State Variable Prescribed at a Fixed Terminal Time

Some hydrosystems problems may have constraining functions of the terminal state,

$$\psi\left[\mathbf{x}(T), T\right] = 0 \tag{2.5.28}$$

where ψ is a j vector ($j \leq n - 1$ if $f = 0$, $j \leq n$ if $f \neq 0$). Equation (2.5.28) is also adjoined to the performance index (2.5.3) by a multiplier vector $\mathbf{v}$ (a j vector).

$$F(\mathbf{u}) = \phi\left[\mathbf{x}(T)\right] + \mathbf{v}^T\psi\left[\mathbf{x}(T), T\right] + \int_0^T \left\{ f\left[\mathbf{x}(t), \mathbf{u}(t), t\right] \right.$$

$$\left. + \lambda^T(t)\left\{ \mathbf{g}\left[\mathbf{x}(t), \mathbf{u}(t), t\right] - \dot{\mathbf{x}} \right\} \right\} dt \tag{2.5.29}$$

The development presented in section 2.5.1 applies here. Bryson and Ho (1975) present the following necessary condition for F to have a stationary value

$$\dot{\mathbf{x}} = \mathbf{g}(\mathbf{x}, \mathbf{u}, t) \qquad \text{(n differential equations)} \quad (2.5.30)$$

$$\dot{\lambda} = -\left(\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right)^T \lambda - \left(\frac{\partial f}{\partial \mathbf{x}}\right)^T \qquad \text{(n differential equations)} \quad (2.5.31)$$

$$\left(\frac{\partial H}{\partial \mathbf{u}}\right)^T = \left(\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right)^T \lambda + \left(\frac{\partial f}{\partial \mathbf{u}}\right)^T = 0 \qquad \text{(m algebraic equations)} \quad (2.5.32)$$

$$x_k\,(t=0) \text{ given as } \lambda_k\,(t=0) = 0,\ k = 1, \ldots, n \qquad \text{(n boundary conditions)} \quad (2.5.33)$$

$$\lambda^T(T) = \left(\frac{\partial \phi}{\partial \mathbf{x}} + \mathbf{v}^T \frac{\partial \psi}{\partial \mathbf{x}}\right)_{t=T} \qquad \text{(n boundary condition)} \quad (2.5.34)$$

$$\psi\left[\mathbf{x}(T), T\right] = 0 \qquad \text{(j side condition)} \quad (2.5.35)$$

The stationarity condition (2.5.32) determines the m-vector $\mathbf{u}(t)$. The 2n differential equations (2.5.29) and (2.5.30) with the 2n boundary conditions (2.5.33) and (2.5.34), from a two-point boundary value problem with j parameters $\mathbf{v}$ to be found in (2.5.34) so that the j side condition (2.5.35) are satisfied.

2.6 Optimal Control Problem for Multi-reservoir Operation

The optimal control problem for multi-reservoir operation can be based upon a deterministic optimal control algorithm to find an optimal release policy for the future time periods (typically months) of operation using current storages and forecasted or historical inflows and demands. Only decision (releases) for the next time period are implemented, and the entire procedure is repeated in each subsequent time period.

Mathematically the objective function for the multi-reservoir operation problem for N reservoirs can be stated as follows:

cost function

$$\text{Minimize } Z = \phi[\mathbf{S}\,(T+1)] + \sum_{t=1}^{T} G_t[\mathbf{S}(t+1), \mathbf{R}(t)] \tag{2.6.1}$$

Releases

where $\mathbf{S}(t)$ is the state variable sequence (reservoir storages) of N-vectors (N reservoirs); $\mathbf{R}(t)$ is the decision variable sequences of M-vectors (M releases); ϕ [ ] is the cost function of the terminal condition; G_t [ ] is the cost function for each stage t (time period) under consideration.

The basic constraints of the multi-reservoir problem consist of mass balance constraints, minimum and maximum storage constraints, and minimum and maximum release constraints. Mass balance is expressed as

Loss

$$\mathbf{S}(t+1) = \mathbf{S}(t) + \mathbf{A}\cdot\mathbf{R}(t) + \mathbf{Q}(t) - \mathbf{L}(t) \tag{2.6.2}$$

Inflows

in which **A** is the routing matrix with N-rows and M-columns where a member n,m is 1 if link m delivers water to node n or -1 if link m takes (releases) water from node n or 0 if link m is not connected to node n; **Q**(t) is the inflow vector to the reservoir; and **L**(t) is the reservoir loss vector.

Constraints on minimum and maximum reservoir storages (states) are expressed as

$$\mathbf{S}_{min} \le \mathbf{S}(t) \le \mathbf{S}_{max} \tag{2.6.3}$$

where $\mathbf{S}_{min}$ is the minimum storage and $\mathbf{S}_{max}$ is the maximum storage, respectively. Constraints on minimum and maximum releases are expressed in a similar manner as

$$\mathbf{R}_{min} \leq \mathbf{R}(t) \leq \mathbf{R}_{max} \tag{2.6.4}$$

where $\mathbf{R}_{min}$ and $\mathbf{R}_{max}$ are the minimum and maximum releases, respectively. Initial conditions of the reservoir system define the storage as

$$\mathbf{S}(1) = \mathbf{S}_o \tag{2.6.5}$$

The discrete maximum principle (Section 2.5) can be applied to the above multi-reservoir operation problem (Mizyed, et al. 1992). To derive the Hamiltonian function, H, the mass balance constraint (2.6.2) is combined with the objective (2.6.1) using the Lagrange multiplier vector, $\lambda(t)$,

$$H = \phi\,[\mathbf{S}(T+1)] + \sum_{t=1}^{T} G[\mathbf{S}(t+1), \mathbf{R}(t)] + \sum_{t=1}^{T} \lambda^T(t)\,[\mathbf{S}(t) + \mathbf{A}\cdot\mathbf{R}(t)$$

$$+ \mathbf{Q}(t) - \mathbf{L}(t) - \mathbf{S}(t+1)] \tag{2.6.6}$$

To include the state-variables constraints (2.6.3) a penalty function is added to the objective function in order to minimize violations in these constraints. Adding the penalty function, the objective function becomes

to solve for opt. solns, you must solve. 2.6.2 2.6.3 2.6.5 2.6.10, 11, 13

$$H = \phi\,[\mathbf{S}(T+1)] + \sum_{t=1}^{T} G_t[\mathbf{S}(t+1), \mathbf{u}(t)] + \eta(t)^T \mathbf{P}\eta(t)\}$$

$$+ \sum_{t=1}^{T} \lambda^T(t)[\mathbf{S}(t) + \mathbf{A}\cdot\mathbf{R}(t) + \mathbf{Q}(t) - \mathbf{L}(t) - \mathbf{S}(t+1)] \tag{2.6.7}$$

Euler Lagrange eqs.

in which $\mathbf{P}$ is a diagonal weighting (penalty) matrix; and $\eta(t)$ is the violation in state variables (N-vectors), or

$$\eta(t) = \min[\mathbf{O}, \mathbf{S}(t+1)\mathrm{L} - \mathbf{S}_{min}] + \max[\mathrm{O}, \mathbf{S}(t+1) - \mathbf{S}_{max}] \tag{2.6.8}$$

The differential changes in H(dH) due to differential changes in $\mathbf{R}(t)$ as follows:

$$dH = \sum_{t=1}^{T}\left[\frac{\partial G_t}{\partial \mathbf{R}(t)} + \lambda^T(t)\mathbf{A}\right]d\mathbf{R}(t)\left[\frac{\partial \phi}{\partial \mathbf{S}(T+1)} - \lambda^T(T) + 2\eta^T(T)\mathbf{P} + \frac{\partial G_T}{\partial \mathbf{S}(T+1)}\right]$$

$$d\mathbf{S}(T+1) + \sum_{t=1}^{T-1}\left[\frac{\partial G_t}{\partial \mathbf{S}(t+1)} - \lambda^T(t) + \lambda^T(t+1) + 2\eta^T(t)\mathbf{P}\right]d\mathbf{S}(t+1) \tag{2.6.9}$$

To determine the differential change $d\mathbf{S}(t)$ produced by a given $d\mathbf{R}(t)$ sequence so that dH is determined in terms of $\mathbf{R}(t)$. The following multiplier sequence $\lambda(t)$, (Bryson and Ho 1975)

$$\lambda^T(t) = \frac{\partial G_t}{\partial \mathbf{S}(t+1)} + \lambda^T(t+1) + 2\eta^T(t)\mathbf{P} \tag{2.6.10}$$

for $t = 1,2,3,\ldots T-1$, and

$$\lambda^T(T) = \frac{\partial \phi[\mathbf{S}(T+1)]}{\partial \mathbf{S}(T+1)} + 2\eta^T(T)\mathbf{P} + \frac{\partial G_T}{\partial \mathbf{S}(T+1)} \tag{2.6.11}$$

is introduced.

Equation (2.6.9) becomes

$$dH = \sum_{t=1}^{T}\left[\frac{\partial G_t}{\partial \mathbf{R}(t)} + \lambda^T(t)\mathbf{A}\right]d\mathbf{R}(t) \tag{2.6.12}$$

by substitution

For an extremum, dH must be zero for any arbitrary $d\mathbf{R}(t)$; so that (Bryson and Ho 1975)

$$\frac{\partial H}{\partial \mathbf{R}(t)} = \frac{\partial G_t}{\partial \mathbf{R}(t)} + \lambda^T(t)\mathbf{A} = 0 \qquad (2.6.13)$$

An optimal solution for (2.6.7) should satisfy (2.6.2), (2.6.3), (2.6.5), (2.6.10), (2.6.11), and (2.6.13).

Conjugate Gradient Method used to Solve this (i.e. mathematical programming)

Mizyed, et al. (1992) implemented the following conjugate gradient method to solve the reservoir operation problem.

1. Assume a set of penalty values for the matrix **P**. Starting with small values.
2. Guess an initial decision vector $\mathbf{R}(t)^{(o)}$.
3. Determine $\mathbf{S}(t)^{(o)}$ from (2.6.3) and (2.6.2)
4. Determine λt from and (2.6.10) and (2.6.11). Solve (2.6.11) for $\lambda(T)$, then solve (2.6.10) backwards from t = T - 1 to t = 1.
5. Determine the gradient of H from (2.6.13), or

$$\frac{\partial H}{\partial \mathbf{R}(t)} = \frac{\partial G_t}{\partial \mathbf{R}(t)} + \lambda^T(t)\mathbf{A} = \mathbf{g}^i \qquad (2.6.14)$$

At optimum, $\mathbf{g}^i = 0$ where i is the iteration number

6. Select a search direction $\mathbf{D}^i$ to be

$$\mathbf{D}^i = \mathbf{g}^i + \frac{||\mathbf{g}^i||}{||\mathbf{g}^{i-1}||^2}\mathbf{D}^{i-1} \qquad \text{for } i \geq 2 \qquad (2.6.15a)$$

$$\mathbf{D}^i = \mathbf{g}^i \qquad \text{for } i = 1 \qquad (2.6.15b)$$

in which $||\mathbf{g}||^2 = \mathbf{g}^T \cdot \mathbf{g}$.

7. Determine the step size δ by one-dimensional search to minimize $H(\mathbf{u}')$, where $\mathbf{u}'$ is the revised decision vector, $\mathbf{R} - \delta\mathbf{D}$. Details of the one-dimensional search may be found in Hiew (1987).
8. The new estimate of **u** or $\mathbf{u}^{(i+1)}$ is given by

$$\mathbf{R}' = \mathbf{R} - \delta\mathbf{D}^i \qquad (2.6.16)$$

$$\mathbf{R}^{(i+1)} = \mathbf{R}_{max} \quad \text{if } \mathbf{R}' \geq \mathbf{R}_{max} \qquad (2.6.17)$$

$$\mathbf{R}^{(i+1)} = \mathbf{R}_{min} \quad \text{if } \mathbf{R} < \mathbf{R}_{min} \qquad (2.6.18)$$

$$\mathbf{R}^{(i+1)} = \mathbf{R}' \quad \text{otherwise} \qquad (2.6.19)$$

9. Go to step 3 using this new estimate for **R**(t).
10. Continues until the gradient or δ vanishes, set $\mathbf{g}^i = 0$ if **R** is equal to $\mathbf{R}_{max}$ or $\mathbf{R}_{min}$.
11. Check the values of n , or the violations in the state variables. If the violations are not permissible, then increase the penalty values (**P**) are increased and the procedure is repeated. Continue increasing the values of **P** until the procedure converges to a feasible solution. The convergence to a feasible optimal solution is discussed in a later section.

Constraint (2.6.5), bounds on decision variables, is invoked at every iteration of the conjugate gradient procedure. Bounds on states are handled through penalty terms, (2.6.7). Pagurek and Woodside (1968) presented a slight refinement of the above procedure for handling bounded control variables which ensures that the method converges to the true optimum. Mizyed, Loffis, and Fontane (1992) applied the above algorithm to the Mahaweli reservoir system in Sir Lanka which includes 19 reservoirs and 35 release links. This application was to minimize hydroelectric energy shortage with prespecified irrigation demand constraints.

REFERENCES

Bryson, A. E., Jr., and Y. C. Ho, Applied Optimal Control: Optimization, Estimation, and Control, Hemisphere Publishing Corp. New York, N.Y. 1975.

Duong, N., Wynn, C. B., and Johnson, G. R., Modern Control Concepts in Hydrology, Trans. on Systems, Man and Cybernetics, Inst. El. El Eng., Vol. SMC-5, No. 1, 1975.

Erschler, J., Roubellat, F., and Vernhes, J. P., Automation of a Hydroelectric Power Station Using Variable Structure Control Systems, Automatica, Vol. 10, No, 1, 1974.

Fan, L. T., Shah, P. S., Periera, N. C., and Erickson, L. E., Dynamic Analysis and Optimal Feedback Control Synthesis Applied to Biological Waste Treatment, Water Research, Vol. 7, No. 11, 1973.

Hiew, K.-L., Optimization Algorithms for Large-Scale Multi-reservoir Hydropower Systems., Ph.D. Thesis, Colorado State University, Fort Collins, Colorado, 1987.

Lapidus, L. and R. Luus, Optimal Control of Engineering Processes, Blaisdell Publishing Co., 1967,

Maidment, D. R., Stochastic State Variable Dynamic Programming for Water Resources Systems Analysis, Ph.D. thesis presented to the University of Illinois at Urbana-Champaign, Ill., 1976.

Maidment, D. R. and Chow, V. T., A New Approach to Urban Water Resources Systems Optimization. Proceedings of the World Environment and Resources Council (WERC) Conference on the Environment of Human Settlements, Brussels, Belgium, March 1976.

Mizyed, N. R., J. C. Loftis, and D. G. Fontane, Operation of Large Multi-reservoir Systems Using Optimal-Control Theory, Journal of Water Resources Planning and Management, ASCE, Vol. 118, No. 4., July/Aug. 1992.

Muzik, I., State Variable Model of Overland Flow, Journal of Hydrology, Vol. 22, No. 3/4, 1974.

Newhauser, G., Introduction to Dynamic Programming, John Wiley and Sons, Inc., N. Y. 1966.

Pagurek, B., and C. M. Woodside, The Conjugate gradient method for optimal control problems with bounded control variables. Automatica, Vol. 4, No. 5, pp. 337-349, 1968.

Prasad, R., A Non-linear Hydrologic Systems Response Model, Journal of Hydraulic Engineering Division, ASCE, Vol. 93, No. HY4, pp. 201-221, 1967.

Pontryagin, L. S., V. G. Boltyanskii, R. V. Gamkrelidzo, and E. F. Mishchenko, The Mathematical Theory of Optimal Processes, (translated by K. N. Trirogoff), Wiley Interscience, New York, 1962.

Tung, Y. K. and L. W. Mays, "State Variable Model for Sewer Network Flow Routing," Journal of the Environmental Engineering Division, ASCE, Vol. 104, No. 1, pp. 15-30, Feb. 1978.

Tung, Y. K. and L. W. Mays, "State Variable Model for Rainfall-Runoff Process, Water Resources Bulletin, AWRA, Vol. 17, No. 2, pp. 181-189, April 1981.

Young, P., and Beck, B., The Modeling and Control of Water Quality in a River System, Automatica, Vol. 10, No. 5, 1974.

Chapter 3
NONLINEAR OPTIMIZATION METHODS

3.1 Matrix Algebra for Nonlinear Programming . . . . . .

3.2 Unconstrained Nonlinear Optimization . . . . . .

3.2.1 Basic Concepts . . . . . .
3.2.2 One-Dimensional Search . . . . . .
3.2.3 Multi-variable Methods . . . . . .

3.3 Constrained Optimization: Optimality Conditions . . . . . .

3.3.1 Lagrange Multiplier . . . . . .
3.3.2 Kuhn-Tucker Conditions . . . . . .

3.4 Constrained Optimization: Generalized Reduced Gradient (GRG) Method . . . . . .

3.4.1 Basic Concepts. . . . . .
3.4.2 General Algorithms and Basis Changes . . . . . .
3.4.3 The Reduced Gradient . . . . . .
3.4.4 Optimality Conditions for GRG Method . . . . . .

3.5 Constrained Nonlinear Optimization: Penalty Function Methods

3.6 Augmented Lagrangian Method . . . . . .

3.7 Optimal Control via Unconstrained Optimization . . . . . .

3.8 Solution of Discrete Optimal Control Problem . . . . . .

CHAPTER 3
NONLINEAR OPTIMIZATION METHODS

Earlier hydrosystems applications of operations research techniques relied mainly upon the use of linear and dynamic programming techniques. The use of these techniques applied to solving hydrosystem problems has been rather widespread in the literature. Linear programming codes are widely available whereas dynamic programming requires a specific code for each application. The use of nonlinear programming in solving hydrosystems problems has not been as widespread even though most of the problems requiring solutions are nonlinear problems. The recent development of new nonlinear programming techniques and the availability of nonlinear programming codes have attracted new applications of nonlinear programming in hydrosystems. Unconstrained nonlinear optimization procedures are described followed by descriptions of constrained nonlinear optimization procedures.

3.1 Matrix Algebra for Nonlinear Programming

To explain the concepts of nonlinear programming, various techniques of matrix algebra and numerical linear algebra are used. A brief introduction to some of the concepts is provided in this secion.

A **function** of many variables f (x) at point x is also an important concept. For a function that is continuous and continuously differentiable, there is a vector of first partial derivatives called the **gradient** or **gradient vector**

$$\nabla f(x) = \left[\frac{\partial f}{\partial x}\right] = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right)^T \tag{3.1.1}$$

where ∇ is the vector of **gradient operator** $(\partial/\partial x_1,\ldots,\partial/\partial x_n)^T$. Geometrically, the gradient vector at a given point represents the direction along which the maximum rate of increase in function value would occur. For f(x) twice continuously differentiable there exists a matrix of second partial derivatives called the **Hessian matrix** or **Hessian**

$$H(x) = \nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{3.1.2}$$

The Hessian is a square and symmetric matrix.

The concepts of **convexity** and **concavity** are used to establish whether a **local optimum, local minimum** or **local maximum,** is also the **global optimum,** which is the best among all solutions. In the univariate case, a function f(x) is said to be **convex** over a region if for every x_a and x_b, $x_a \neq x_b$, the following holds

$$f\left[\theta x_a + (1-\theta)x_b\right] \leq \theta f(x_a) + (1-\theta) f(x_b), \; 0 \leq \theta \leq 1 \tag{3.1.3}$$

The function is **strictly convex** when the above relation holds with a less than (<) sign.

Conversely, a function is **concave** over a region if for every x_a and x_b, $x_a \neq x_b$, the following holds

$$f\left[\theta x_a + (1-\theta)\, x_b\right] \geq \theta f(x_a) + (1-\theta)\, f(x_b),\ 0 \leq \theta \leq 1 \tag{3.1.4}$$

The function is **strictly concave** when the above relation holds with a greater than (>) sign.

Equations (3.1.4) and (3.1.5) are not convenient to use in testing for convexity or concavity of a univariate function. Instead, it is easier to examine the sign of its second derivative, $d^2 f(x)/dx^2$. From fundamental calculus, if $\frac{d^2 f}{dx^2} < 0$ then the function is concave and if $\frac{d^2 f}{dx^2} > 0$ then the function is convex.

The convexity and concavity of multivariable functions f (x) can also be determined using the Hessian matrix. First, the definitions of **positive definite, negative definite** and **indefinite** are used to identify the type of Hessian, i.e.

<u>Positive definite</u> **H**:	$\mathbf{x}^T \mathbf{H}\mathbf{x}$	> 0	for all $\mathbf{x} \neq \mathbf{0}$
<u>Negative definite</u> **H**:	$\mathbf{x}^T \mathbf{H}\mathbf{x}$	$< \mathbf{0}$	for all $\mathbf{x} \neq \mathbf{0}$
<u>Indefinite</u> **H**:	$\mathbf{x}^T \mathbf{H}\mathbf{x}$	< 0	for some **x**;
		> 0	for other **x**
<u>Positive semidefinite</u> **H**:	$\mathbf{x}^T \mathbf{H}\mathbf{x}$	≥ 0	for all **x**
<u>Negative semidefinite</u> **H**:	$\mathbf{x}^T \mathbf{H}\mathbf{x}$	≤ 0	for all **x**

The basic rules for convexity and concavity of a multivariate function f(x) with continuous second partial derivatives are:

(1) f(x) is concave, **H**(x) is negative semidefinite;

(2) f(x) is strictly concave, **H**(x) is negative definite;

(3) f(x) is convex, **H**(x) is positive semidefinite;

(4) f(x) is strictly convex, **H**(x) is positive definite.

To test the status of **H**(x) for strict convexity, two tests are available (Edgar and Himmelblau, 1988). The first is that all diagonal elements of **H**(x) must be positive and the determinants of all leading principal minors, det $\{M_i(\mathbf{H})\}$, and also of **H**(x), det (**H**) are positive (> 0) . Another test is that all eigenvalues of **H**(x) are positive (> 0). For strict concavity all diagonal elements must be negative and det (**H**) and det $\{M_i(\mathbf{H})\} > 0$ if i is even (i = 2, 4, 6, ...); det (**H**) and det $\{M_i(\mathbf{H})\} < 0$ if i is odd (i = 1, 3, 5, ...). The strict inequalities > or < in these tests are replaced by ≥ or ≤, respectively, to test for convexity and concavity.

Convex regions or sets are used to classify constraints. A convex region exists if for any two points in the region, $\mathbf{x}_a \neq \mathbf{x}_b$, all points $\mathbf{x} = \theta\mathbf{x}_a + (1-\theta)\,\mathbf{x}_b$, where $0 \leq \theta \leq 1$, are on the line connecting $\mathbf{x}_a$ and $\mathbf{x}_b$ are in the set. Figure 3.1.1 illustrates convex and nonconvex regions.

The convexity of a feasible region and the objective function in nonlinear optimization has an extremely important implication with regard to the type of optimal solution to be obtained. For linear programming problems, the objective function and feasible region both are convex therefore

the optimal solution is a global. On the other hand, the convexity of both the objective function and feasible region in a nonlinear programming problem cannot be ensured, the optimal solution achieved, therefore, cannot be guaranteed to be global.

3.2 Unconstrained Nonlinear Programming

This section describes the basic concepts of unconstrained nonlinear optimization including the necessary and sufficient conditions of a local optimum. Further, unconstrained optimization techniques for univariate and multivariate problems are described. Understanding unconstrained optimization procedures is important because these techniques are the fundamental building blocks in many of the constrained nonlinear optimization algorithms.

3.2.1 Basic Concepts

The problem of unconstrained minimization can be stated as

$$\underset{x \in E^n}{\text{Minimize}} \quad f(x) \tag{3.2.1}$$

in which x is a vector of n decision variables $x=(x_1, x_2, ..., x_n)^T$ defined over the entire Euclidean space E^n. Since the feasible region is infinitely extended without bound, the optimization problem does not contain any constraints.

Assume that f(x) is a nonlinear function and twice differentiable; it could be convex, concave, or a mixture of the two over E^n. In the one-dimensional case, the objective function f(x) could behave as Figure 3.2.1(a) consisting of peaks, valleys, and inflection points. The **necessary conditions**

for a solution to equation (3.2.1) at $\mathbf{x}^*$ are (1) $\nabla f(\mathbf{x}^*)=\mathbf{0}$ and (2) $\nabla^2 f(\mathbf{x}^*) = \mathbf{H}(\mathbf{x}^*)$ is semi-positive definite. The **sufficient conditions** for an unconstrained minimum are (1) $\nabla f(\mathbf{x}^*)=\mathbf{0}$ and (2) $\nabla^2 f(\mathbf{x}^*)=\mathbf{H}(\mathbf{x}^*)$ is strictly positive definite.

In theory, the solution to equation (4.4.1) can be obtained by solving the following system of n nonlinear equations with n unknowns,

$$\nabla f(\mathbf{x}^*)=\mathbf{0} \tag{3.2.2}$$

The approach has been viewed as indirect in the sense that it backs away from the original problem of minimizing $f(\mathbf{x})$. Furthermore, an iterative numerical procedure is required to solve the system of nonlinear equations which tends to be computationally inefficient.

By contrast, the preference is given to those solution procedures which directly attack the problem of minimizing $f(\mathbf{x})$. Direct solution methods, during the course of iteration, generate a sequence of solution points in E^n that terminate or converge to a solution to equation (3.2.1). Such methods can be characterized as search procedures.

In general, all search algorithms for unconstrained minimization consist of two basic steps. The first step is to determine the **search direction** along which the objective function value decreases. The second step is called a **line search** (or **one dimensional search**) to obtained the optimum solution point along the search direction determined by the first step. Mathematically, minimization for the line search can be stated as

$$\min_{\beta} f(\mathbf{x}^0 + \beta \mathbf{d}) \tag{3.2.3}$$

in which $\mathbf{x}^0$ is the current solution point, **d** is the vector indicating the search direction, and β is a scalar, $-\infty < \beta < \infty$, representing the step size whose optimal value is to be determined. There are many search algorithms whose differences primarily lie in the way the search direction **d** is determined.

Due to the very nature of search algorithms, it is likely that different starting solutions might converge to different local minima. Hence, there is no guarantee of finding the global minimum by any search technique applied to solve equation (3.2.1) unless the objective function is a convex function over E^n.

In implementing search techniques, specification of convergence criteria or stopping rules is an important element that affects the performance of the algorithm and the accuracy of the solution. Several commonly used stopping rules in an optimum seeking algorithm are

(a) $$\| x^k - x^{k+1} \| < \varepsilon_1; \tag{3.2.4a}$$

(b) $$\frac{\| x^k - x^{k+1} \|}{\| x^k \|} < \varepsilon_2; \tag{3.2.4b}$$

(c) $$| f(x^k) - f(x^{k+1}) | < \varepsilon_3; \tag{3.2.4c}$$

(d) $$\left| \frac{f(x^k) - f(x^{k+1})}{f(x^k)} \right| < \varepsilon_4; \tag{3.2.4d}$$

in which superscript 'k' is the index for iteration, ε represents the tolerance or accuracy requirement, $\| x \|$ is the length of the vector **x**, and $| x |$ is the absolute value. The specification of the tolerance depends on the nature of the problem and on the accuracy requirement. Too small a value of ε (corresponding to high accuracy requirement) could result in excessive

iterations, wasting computer time. On the other hand, too large a value of ε could make the algorithm terminate prematurely at a non-optimal solution.

3.2.2 Unconstrained Optimization: One-Dimensional Search

The line search techniques for solving one-dimensional optimization problems form the backbone of nonlinear programming algorithms. Multi-dimensional problems are ultimately solved by executing a sequence of successive line searches. One-dimensional search techniques can be classified as curve fitting (approximation) techniques or as interval elimination techniques. **Interval elimination** techniques for a one-dimensional search essentially eliminate or delete a calculated portion of the range of the variable from consideration in each successive iteration of the search for the optimum of f (x). After a number of iterations when the remaining interval is sufficiently small the search procedure terminates. These methods determine the minimum value of a function over a closed interval [a,b] assuming that a function is **unimodal**, i.e., it has only one minimum value in the interval (Figure 3.2.1). Two interval elimination techniques commonly used are the golden section method and the Fibonacci search method (Mays and Tung, 1992).

3.2.3 Unconstrained Optimization: Multivariable Methods

Unconstrained optimization problems can be stated in a general form as

$$\text{Minimize } z = f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n) \tag{3.2.5}$$

For maximization, the problem is to minimize -f (x). The solution of these

types of problems can be stated in an algorithm involving the following basic steps or phases:

Step (0) Select an initial starting point $\mathbf{x}^{k=0} = \left(x_1^0, x_2^0, \ldots, x_n^0\right)$.

Step (1) Determine a search direction, $\mathbf{d}^k$.

Step (2) Find a new point $\mathbf{x}^{k+1} = \mathbf{x}^k + \beta^k \mathbf{d}^k$ where β^k is the step size, a scalar, which minimizes $f(\mathbf{x}^k + \beta^k \mathbf{d}^k)$.

Step (3) Check the convergence criteria such as equations (3.2.4a-e) for termination, if not satisfied set $k = k+1$ and return to step (1).

The various unconstrained multivariate methods differ in the way the search directions are determined. The recursive line search for an unconstrained minimization problem is expressed in Step (2) above as

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \beta^k \mathbf{d}^k \tag{3.2.6}$$

Table 3.2.1 lists the equations for determining the search direction for four basic groups of methods: descent methods, conjugate direction methods, quasi-Newton methods and Newton's method. The simplest are the steepest descent methods while the Newton methods are the most computationally intensive.

In the **steepest descent method** the search direction is $-\nabla f(\mathbf{x})$. $\nabla f(\mathbf{x})$ points in the direction of the maximum rate of increase in objective function value, therefore, a negative sign is associated with the gradient vector in

equation (4.4.6) because the problem is a minimization type. The recursive line search equation for the steepest descent method is, then, reduced to

$$x^{k+1} = x^k - \beta^k \nabla f(x^k) \tag{3.2.7}$$

Using **Newton's method,** the recursive equation for line search is

$$x^{k+1} = x^k - H^{-1}(x^k) \nabla f(x^k) \tag{3.2.8}$$

Although Newton's method converges faster than most other algorithms, the major disadvantage is that it requires inverting the Hessian matrix in each iteration which is a computationally cumbersome task.

The **conjugate direction methods** and **quasi-Newton's methods** are intermediate between the steepest descent and Newton's method. The conjugate direction methods are motivated by the need to accelerate the typically slow convergence of the steepest descent methods. Conjugate direction methods, as can be seen in Table 3.2.1, define the search direction by utilizing the gradient vector of the objective funciton of the current iteration and the information on the gradient and search direction of the previous iteration. The motivation of quasi-Newton methods is to avoid inverting the Hessian matrix as required by Newton's method. These methods use approximations to the inverse Hessian with a different form of approximation for the different quasi-Newton methods. Detailed descriptions and theoretical development can be found in textbooks such as Luenberger (1984), Fletcher (1980), Dennis and Schnable (1983), and Gill, Murray and Wright (1981) and Edgar and Himmelblau (1988).

3.3 Constrained Optimization: Optimality Conditions

3.3.1 Lagrange Multiplier

Consider the general nonlinear programming problem with the nonlinear objective:

$$\text{Minimize } f(\mathbf{x}) \tag{3.3.1a}$$

subject to

$$g_i(\mathbf{x}) = 0 \qquad i = 1,\ldots, m \tag{3.3.1b}$$

and

$$\underline{x}_j \le x_j \le \bar{x}_j \qquad j = 1, 2,\ldots, n \tag{3.3.1c}$$

in which equation (3.3.1c) is a bound constraint for the j-th decision variable x_j with $\underline{x}_j$ and $\bar{x}_j$ being the lower and upper bounds, respectively.

In a constrained optimization problem, the feasible space is not infinitely extended, unlike an unconstrained problem. As a result, the solution that satisfies the optimality condition of the unconstrained optimization problem does not guarantee to be feasible in constrained problems. In other words, a local optimum for a constrained problem might be located on the boundary or a corner of the feasible space at which thegradient vector is not equal to zero. Therefore, modifications to the optimality conditions for unconstrained problems must be made.

The most important theoretical results for nonlinear constrained optimization are the **Kuhn-Tucker conditions**. These conditions must be satisfied at any constrained optimum, local or global, of any linear and nonlinear programming problems. They form the basis for the development of many computational algorithms.

Without losing generality, consider a nonlinear constrained problem stated by equation (3.3.1) with no bounding constraints. Note that constraint equations (3.3.1b) are all equality constraints. Under this condition, the **Lagrange multiplier method** converts a constrained nonlinear programming problem into an unconstrained one by developing an augmented objective function, called the **Lagrangian**. For a minimization, the **Lagrangian function** $L(x, \lambda)$ is defined as

$$L(x, \lambda) = f(x) + \lambda^T g(x) \tag{3.3.2}$$

in which λ is the vector of **Lagrange multipliers** and $g(x)$ is a vector of constraint equations. Algebraically, equation (3.3.2) can be written

$$L(x_1, \ldots, x_n, \lambda_1, \ldots, \lambda_m) = f(x_1, \ldots, x_n) + \sum_{i=1}^{m} \lambda_i g_i(x_1, \ldots, x_n) \tag{3.3.3}$$

$L(x, \lambda)$ is the objective function, with m+n variables, that is to be minimized. The necessary and sufficient conditions for x^* to be the solution for minimization are:

(1) $f(x^*)$ is convex and $g(x^*)$ is convex in the vicinity of x^*

(2) $$\frac{\partial L(x^*)}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^{m} \lambda_i \frac{\partial g_i}{\partial x_j} = 0 \qquad j = 1, \ldots, n \tag{3.3.4a}$$

(3) $$\frac{\partial L}{\partial \lambda_i} = g_i(x) = 0 \qquad i = 1, \ldots, m \tag{3.3.4b}$$

(4) λ_i is unrestricted-in-sign $\qquad i = 1, \ldots, m$ (3.3.4c)

Solving equations (3.3.4a) and (3.3.4b) simultaneously provides the optimal solution.

Lagrange multipliers have an important interpretation in optimization. For a given constraint, these multipliers indicate how much the optimal objective function value will change for a differential change in the right-hand side of the constraint. That is,

$$\left.\frac{\partial f}{\partial b_i}\right|_{x=x^*} = \lambda$$

illustrating that the Lagrange multiplier λ_i is the rate of change of the optimal value of the original objective function with respect to a change in the value of the right-hand side of the i-th constraint. The λ_i's are called **dual variables** or **shadow prices**.

3.3.2 Kuhn-Tucker Conditions

Equations (3.3.4a)-(3.3.4c) form the optimality conditions for an optimization problem involving only equality constraints. The Lagrange multipliers associated with the equality constraints are unrestricted-in-sign. Using the Lagrange multiplier method, the optimality conditions for the following generalized nonlinear programming problem can be derived.

Minimize f(x)

subject to

$$g_i(x) = 0 \qquad i = 1,\ldots, m$$

and

$$\underline{x}_j \le x_j \le \bar{x}_j \qquad j = 1,\ldots, n$$

In terms of the Lagrangian method, the above nonlinear minimization problem can be written as

$$\text{Min } L = f(x) + \lambda^T g(x) + \underline{\lambda}^T (\underline{x} - x) + \bar{\lambda}^T (x - \bar{x}) \tag{3.3.5}$$

in which λ, $\underline{\lambda}$, and $\bar{\lambda}$ are vectors of Lagrange multipliers corresponding to constraints $g(x) = 0$, $\underline{x} - x \leq 0$, and $x - \bar{x} \leq 0$, respectively. The Kuhn-Tucker conditions for the optimality of the above problem are

$$\nabla_x L = \nabla_x f + \lambda^T \nabla_x g - \underline{\lambda} + \bar{\lambda} = 0 \tag{3.3.6a}$$

$$g_i(x) = 0 \qquad i = 1, 2, \ldots, m \tag{3.3.6b}$$

$$\underline{\lambda}_j (\underline{x}_j - x_j) = \bar{\lambda}_j (x_j - \bar{x}_j) = 0 \qquad j = 1, 2, \ldots, n \tag{3.3.6c}$$

$$\lambda \text{ unrestricted-in-sign}, \ \underline{\lambda} \geq 0, \bar{\lambda} \geq 0 \tag{3.3.6d}$$

3.4 Constrained Nonlinear Optimization: Generalized Reduced Gradient (CRG) Method

3.4.1 Basic Concepts

Similar to the linear programming simplex method, the fundamental idea of the generalized reduced gradient method is to express m (number of constraint equations) of the variables, called **basic variables**, in terms of the remaining n-m variables, called **nonbasic variables**. The decision variables can then be partitioned into the basic variables, x_B, and the nonbasic variables, x_N,

$$x = (x_B, x_N)^T \tag{3.4.1}$$

Nonbasic variables not at their bounds are called **superbasic variables,** Murtaugh and Saunders (1978).

The optimization problem can now be restated in terms of the basic and nonbasic variables

$$\text{Minimize } f(x_B, x_N) \tag{3.4.2a}$$

subject to

$$g(x_B, x_N) = 0 \tag{3.4.2b}$$

and

$$\underline{x}_B \le x_B \le \bar{x}_B \tag{3.4.2c}$$

$$\underline{x}_N \le x_N \le \bar{x}_N \tag{3.4.2d}$$

The m basic variables in theory can be expressed in terms of the n-m nonbasic variables as $x_B(x_N)$. Assume that constraints $g(x) = 0$ is differentiable and the m by m **basis matrix B** can be obtained as

$$B = \left[\frac{\partial g(x)}{\partial x_B}\right]$$

which is nonsingular such that there exists a unique solution of $x_B(x_N)$. **Nonsingular** means that the determinant of $B \neq 0$.

The objective called a **reduced objective** can be expressed in terms of the nonbasic variables as

$$F(x_N) = f(x_B(x_N), x_N) \tag{3.4.3}$$

The original nonlinear programming problem is transformed into the following **reduced problem**

$$\text{Minimize } F(x_N) \tag{3.4.4a}$$

subject to

$$\underline{x}_N \le x_N \le \bar{x}_N \tag{3.4.4b}$$

which can be solved by an unconstrained minimization technique with slight modification to account for the bounds on nonbasic variables. Generalized reduced gradient algorithms, therefore, solve the original problem (3.3.1) by solving a sequence of reduced problems (3.4.4), using unconstrained minimization algorithms.

3.4.2 General Algorithm and Basis Changes

Consider solving the reduced problem (3.4.4) starting from an initial feasible point x^0. To evaluate $F(x_N)$ by equation (3.4.3), the values of the basic variables x_B must be known. Except for a very few cases, $x_B(x_N)$ cannot be determined in closed form; however, it can be computed for any x_N by an iterative method which solves a system of m nonlinear equations with the same number of unknowns as equations. A procedure for solving the reduced problem starting from the initial feasible solution $x^{k=0}$ is

Step (0) Start with initial feasible solution $x^{k=0}$ and set $x_N^k = x^{k=0}$

Step (1) Substitute x_N^k into equation (3.4.2b) and determine the corresponding values of x_B by an iterative method for solving m nonlinear equations $g(x_B(x_N^k), x_N^k) = 0$.

Step (2) Determine the search direction d^k for the nonbasic variables by a line search scheme.

Step (3) Choose a step size for the line search scheme, β^k such

that

$$x_N^{k+1} = x_N^k + \beta^k d^k \tag{3.4.5}$$

This is done by solving the one-dimensional search problem

$$\text{Minimize } F\left(x_N^k + \beta d^k\right)$$

with x restricted so that $x_N^k + \beta d^k$ satisfies the bounds

on xN. This one-dimensional search requires repeated applications of Step (1) to evaluate F for the different

β values.

Step (4) Test the current point $x^k = (x_B^k, x_N^k)$ for optimality, if not optimal, set k = k + 1 and return to Step (1).

Refer to Figure 3.4.1 the optimization problem can be stated as

Minimize $f\left(x_1, x_2\right)$

subject to

$g_1(x_1, x_2) \geq 0$

$g_2(x_1, x_2) \geq 0$

$x_1, x_2 \geq 0$

The two inequality constraints can be converted to equality constraints using slack variables x_3 and x_4

$$g_1(x_1, x_2) - x_3 = 0$$

$$g_2(x_1, x_2) - x_4 = 0$$

$$x_i \geq 0 \qquad i = 1, \ldots, 4$$

The initial point A is on the curve g_2 $(x_1, x_2, x_3) = 0$ where the only variable that cannot be basic is x_4 which is at its lower bound of zero. The reduced objective is F (x_2, x_4) which is the objective function f(x) evaluated on g_2 $(x_1, x_2, x_3) = 0$.

For purposes of illustrating the basis changes, assume that the algorithm moves along the curve g_2 $(x_1, x_2, x_3) = 0$ as indicated by the arrow in Figure 3.4.1 until the curve g_1 $(x_1, x_2, x_4) = 0$ is reached. It should be kept in mind that an algorithm could move interior from the initial point A, releasing x_4 from its lower bound of zero, but for the sake of illustration of basis changes, the procedure here will stay on the curves. At the point B where constraints g_1 and g_2 intersect, the slack variable x_3 goes to zero. Because x_3 is originally basic it must leave the basis and be replaced by one of the nonbasics, x_2 or x_4. Because x_4 is zero, x_2 becomes basic and the new reduced objective is F_2 (x_3, x_4) with x_3 and x_4 at their lower bounds of zero. Once again, assuming the algorithm moves along the curve g_1 $(x_1, x_2, x_4) = 0$ towards the x_2 axis, F_2 is minimized at point C where x_1 becomes zero (nonbasic) and x_4 becomes basic. The procedure would then move along the x_2 axis to point D which is obviously the minimum.

3.4.2 The Reduced Gradient

Computation of the reduced gradient is required in the generalized reduced gradient method in order to define the search direction. Consider the simple problem

$$\text{Minimize } f\left(x_1, x_2\right)$$

subject to

$$g\left(x_1, x_2\right) = 0$$

The total derivative of the objective function is

$$df(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial x_1} dx_1 + \frac{\partial f(\mathbf{x})}{\partial x_2} dx_2 \tag{3.4.6}$$

and the total derivative of the constraint function is

$$dg(\mathbf{x}) = \frac{\partial g(\mathbf{x})}{\partial x_1} dx_1 + \frac{\partial g(\mathbf{x})}{\partial x_2} dx_2 = 0 \tag{3.4.7}$$

The **reduced gradients** are $\nabla f(\mathbf{x})$ and $\nabla g(\mathbf{x})$ defined by the coefficients in the total derivatives,

$$\nabla f(\mathbf{x}) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right]^T \tag{3.4.8}$$

$$\nabla g(\mathbf{x}) = \left[\frac{\partial g}{\partial x_1}, \frac{\partial g}{\partial x_2}\right]^T \tag{3.4.9}$$

Consider the basic (dependent) variable to be x_1 and the nonbasic (independent) variable to be x_2. Equation (3.4.7) can be used to solve for dx_1

$$dx_1 = \frac{\partial g(x)/\partial x_2}{\partial g(x)/\partial x_1} dx_2 \tag{3.4.10}$$

which is then substituted into equation (4.6.6) in order to eliminate dx_1. The resulting total derivative of the objective function f(x) can be expressed as

$$df(x) = \left\{ -\left(\frac{\partial f(x)}{\partial x_1}\right)\left(\frac{\partial g(x)}{\partial x_1}\right)^{-1}\left(\frac{\partial g(x)}{\partial x_2}\right) + \left(\frac{\partial f(x)}{\partial x_2}\right)\right\} dx_2 \tag{3.4.11}$$

The reduced gradient is the expression in brackets { } and can be reduced to

$$\frac{df(x)}{dx_2} = \frac{\partial f(x)}{\partial x_2} - \left(\frac{\partial f(x)}{\partial x_1}\right)\left(\frac{\partial x_1}{\partial x_2}\right) \tag{3.4.12}$$

which is scalar because there is only one nonbasic variable x_2.

The reduced gradient can be written in vector form for the multiple variable case as

$$\nabla_N F = \left[\frac{\partial F}{\partial x_N}\right] = \left[\frac{\partial f(x)}{\partial x_N}\right] - \left[\frac{\partial f(x)}{\partial x_B}\right]^T \left[\frac{\partial g(x)}{\partial x_B}\right]^{-1} \left[\frac{\partial g(x)}{\partial x_N}\right] \tag{3.4.13}$$

in which

$$\left[\frac{\partial x_B}{\partial x_N}\right] = \left[\frac{\partial g(x)}{\partial x_B}\right]^{-1} \left[\frac{\partial g(x)}{\partial x_N}\right] \quad = \mathbf{B}^{-1}\left[\frac{\partial g(x)}{\partial x_N}\right] \tag{3.4.14}$$

The **Kuhn-Tucker multiplier** vector π is defined by

$$\left[\frac{\partial f(x)}{\partial x_B}\right]^T \left[\frac{\partial g(x)}{\partial x_B}\right]^{-1} = \left[\frac{\partial f(x)}{\partial x_B}\right]^T \mathbf{B}^{-1} = \pi^T \tag{3.4.15}$$

Using these definitions the reduced gradient in equation (4.6.13) can be expressed as

$$\nabla_N F = \left[\frac{dF}{dx_N}\right] = \left[\frac{\partial f(x)}{\partial x_N}\right] - \pi^T \left[\frac{\partial g(x)}{\partial x_N}\right] \tag{3.4.16}$$

3.4.4 Optimality Conditions for GRG Method

Consider the nonlinear programming problem

Minimize f (x)

subject to

$$g_i(x) = 0 \qquad i = 1,\ldots, m$$

$$\underline{x}_j \le x_j \le \bar{x}_j \qquad j = 1,\ldots, n$$

In terms of basic and nonbasic variables, the Lagrangian function for the problem can be stated as

$$L = f(x) + \lambda^T g(x) + \underline{\lambda}^T (\underline{x} - x) + \bar{\lambda}^T (x - \bar{x})$$

$$= f(x_B, x_N) + \lambda^T g(x_B, x_N) + \underline{\lambda}_B^T (\underline{x}_B - x_B) + \underline{\lambda}_N^T (\underline{x}_N - x_N)$$

$$+ \bar{\lambda}_B^T (x_B - \bar{x}_B) + \bar{\lambda}_N^T (x_N - \bar{x}_N) \tag{3.4.17}$$

in which λ_N and λ_B are vectors of Lagrange multipliers for nonbasic and basic variables, respectively.

Based on equation (4.5.6), the Kuhn-Tucker conditions for optimality in terms of the basic and nonbasic variables are

$$\nabla_B L = \nabla_B f + \lambda^T \nabla_B g - \underline{\lambda}_B + \bar{\lambda}_B = 0 \tag{3.4.18a}$$

$$\nabla_N L = \nabla_N f + \lambda^T \nabla_N g - \underline{\lambda}_N + \bar{\lambda}_N = 0 \tag{3.4.18b}$$

$$\underline{\lambda}_B \geq 0 \qquad \underline{\lambda}_N \geq 0 \tag{3.4.18c}$$

$$\bar{\lambda}_B \geq 0 \qquad \bar{\lambda}_N \geq 0 \tag{3.4.18d}$$

$$\underline{\lambda}_B^T (\underline{x}_B - x_B) = \bar{\lambda}_B^T (x_B - \bar{x}_B) = 0 \tag{3.4.18e}$$

$$\underline{\lambda}_N^T (\underline{x}_N - x_N) = \bar{\lambda}_N^T (x_N - \bar{x}_N) = 0 \tag{3.4.18f}$$

If x_B is strictly between its bounds then $\underline{\lambda}_B = \bar{\lambda}_B = 0$ by equation (4.6.18e) so that from equation (4.6.18a),

$$\lambda^T = \left[-\frac{\partial f}{\partial x_B}\right]^T \left[\frac{\partial g}{\partial x_B}\right]^{-1} = \left[-\frac{\partial f}{\partial x_B}\right]^B B^{-1} = -\pi^T \tag{3.4.19}$$

In other words, when $\underline{x}_B < x_B < \bar{x}_B$, the Kuhn-Tucker multiplier vector π is the Lagrange multiplier vector for the equality constraints $g(x)=0$. Then from equations (3.4.16) and (3.4.18b)

If x_N is strictly between its bounds, i.e., $\underline{x}_N < x_N < \bar{x}_N$, then $\underline{\lambda}_N = \bar{\lambda}_N = 0$ by equation (3.4.18f) so that

$$\left[\frac{\partial F}{\partial x_N}\right] = 0 \tag{3.4.20}$$

If x_N is at its lower bound, $x_N = \underline{x}_N$, then $\bar{\lambda}_N = 0$ so

$$\left[\frac{\partial F}{\partial x_N}\right] = \underline{\lambda}_N \geq 0 \tag{3.4.21}$$

If x_N is at its upper bound, $x_N = \bar{x}_N$, then $\underline{\lambda}_N = \mathbf{0}$ so that

$$\left[\frac{\partial F}{\partial x_N}\right] = \bar{\lambda}_N \leq 0 \tag{3.4.22}$$

The above three equations, equations (4.6.20) - (4.6.22), define the optimality conditions for the reduced problem (4.6.4). The Kuhn-Tucker conditions for the original problem may be viewed as the optimality conditions for the reduced problem.

3.5 Constrained Nonlinear Optimization: Penalty Function Methods

The essential idea of penalty function methods is to transform constrained nonlinear programming problems into a sequence of unconstrained optimization problems. The basic idea of these methods is to add one or more functions of the constraints to the objective function and to delete the constraints. Basic reasoning for such approaches is that the unconstrained problems are much easier to solve. Using a penalty function a constrained nonlinear programming problem is transformed to an unconstrained problem.

$$\left.\begin{array}{l}\text{Minimize } f(\mathbf{x}) \\ \text{subject to } \mathbf{g}(\mathbf{x})\end{array}\right\} \Rightarrow \text{ Minimize } L\,[f(\mathbf{x}), \mathbf{g}(\mathbf{x})$$

where L [f(x), g(x)] is a **penalty function**. Various forms of penalty functions have been proposed which can be found elsewhere (McCormick, 1983; Gill, Murray and Wright, 1981) The penalty function is minimized by stages for a series of values of parameters associated with the penalty. In fact, the Lagrangian function (described in Section 3.3.) is one form of penalty function. For many of the penalty functions, the Hessian of the penalty

function becomes increasingly **ill-conditioned** (i.e. the function value is extremely sensitive to a small change in the parameter value) as the solution approaches the optimum. This section briefly describes a penalty function method called the **augmented Lagrangian method**.

The augmented Lagrangian method adds a quadratic penalty function loss term to the Lagrangian function (equation 3.3.2), to obtain

$$L_A(x, \lambda, \psi) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \frac{\psi}{2} \sum_{i=1}^{m} g_i^2(x)$$

$$= f(x) + \lambda^T g(x) + \frac{\psi}{2} g(x)^T g(x) \qquad (3.5.1)$$

where ψ is a positive penalty parameter. Some desirable properties of equation (3.5.1) are discussed by (Gill, Murray and Wright, 1981).

λ^* ?

For ideal circumstances, x^* can be computed by a single unconstrained minimization of the differentiable function (equation 3.5.1). However, in general, λ^* is not available until the solution has been determined. An augmented Lagrangian method, therefore, must include a procedure for estimating the Lagrange multipliers. Gill, Murray and Wright (1981) present the following algorithm:

Step (0) Select initial estimates of the Lagrange multipliers $\lambda^{k=0}$, the penalty parameter ψ, an initial point $x^{k=0}$. Set $k = k+1$ and set the maximum number of iterations as J.

Step (1) Check to see if x^k satisfies optimality conditions or if $k > J$. If so, terminate the algorithm.

Step (2) Minimize the augmented Lagrangian function,

Minimize $L_A(x, \lambda, \psi)$, in equation (3.5.1).

Procedures to consider unboundedness must be considered. The best solution is denoted as $\mathbf{x}^{k+1}$. Should this be x^{*k+1}

Step (3) Update the multiplier estimate by computing λ^{k+1}.

Step (4) Increase the penalty parameter ψ if the constraint violations at $\mathbf{x}^{k+1}$ have not decreased sufficiently from those at $\mathbf{x}^{k}$.

Step (5) Set k = k+1 and return to Step (1).

Augmented Lagrangian methods can be applied to inequality constraints. For the set of violated constraints, $\mathbf{g}(\mathbf{x})$ at $\mathbf{x}^k$, the augmented Lagrangian function has discontinuous derivatives at the solution if any of the constraints are active (Gill, Murray and Wright, 1981). Buys (1972) and Rockafellar (1973a,b, 1974) presented the augmented Lagrangian function for inequality-constrained problems

$$L_A(x, \lambda, \psi) = f(x) + \sum_{i=1}^{m} \begin{cases} \lambda_i g_i(x) + \frac{\psi}{2}[g_i(x)]^2, & \text{if } g_i(x) \le \frac{\lambda_i}{\psi} \\ -\frac{\psi}{2}\lambda_i^2 & \text{if } g_i(x) > \frac{\lambda_i}{\psi} \end{cases} \tag{3.5.2}$$

3.6 The Augmented Lagrangian Method

3.6.1 Introduction

(1) Equality Constraints Only

Initially, a nonlinear programming problem with equality constraints will be discussed and later, another problem with inequality constraints will be considered.

The nonlinear programming problem with equality constraints only can be stated as

$$\text{Minimize} \quad f(x) \tag{3.6.1}$$

$$\text{subject to} \quad c_i(x) = 0, \qquad i = 1, 2, \ldots, m \tag{3.6.2}$$

where **x** is a vector of n components and usually $n \geq m$. An optimal solution, **x***, can be obtained by solving the corresponding related set of n+m nonlinear equations,

$$\nabla_\lambda L(x^*,\lambda^*) = c_i(x^*) = 0 \tag{3.6.3}$$

$$\nabla_x L(x^*,\lambda^*) = \nabla f(x^*) - \sum_i \lambda_i^* \nabla c_i(x^*) = 0 \tag{3.6.4}$$

where λ^* is the m-dimensional Lagrange multiplier vector and is part of the entire solution vector, besides the n-dimensional **x*** vector. (3.6.3) and (3.6.4) are the first derivative vectors with respect to λ and **x**, respectively, evaluated at **x*** of the Lagrangian function

$$L(x, \lambda) = f(x) - \sum_i \lambda_i c_i(x) \tag{3.6.5}$$

In the classical sense (3.6.3) and (3.6.4) comprise the first-order necessary conditions for **x*** to be at least a local minimum of f (**x**). If the Lagrangian function (3.6.5) does not contain a saddle point, however, the solution obtained may not yield a minimum point (Lasdon, 1970).

Rewriting (3.6.1)-(3.6.3) in an exterior penalty type form

$$P(x, \theta, \sigma) = f(x) + \frac{1}{2}\sum_i \sigma_i \left[c_i(x) - \theta_i \right]^2, \qquad i = 1, 2, \ldots, m$$

$$= f(x) + \frac{1}{2}\left[c(x) - \theta\right]^T S\left[c(x) - \theta\right] \qquad (3.6.6)$$

where θ is an m-dimensional parameter vector and S is an m x m diagonal matrix whose elements are the penalty weights $\sigma_i > 0$. The solution procedure to the unconstrained minimization problem (3.6.6) involves the variation of σ_i and θ_i in such a way that $x(\sigma, \theta) \rightarrow x^*$. When $\theta = 0$ the second term in (3.6.6) is sometimes called the penalty term. At each iteration, when x becomes infeasible this penalty term is added to f (x) and σ is increased for the next iteration. Convergence is guaranteed by letting σ_i approach infinity. When these penalty weights are allowed to grow without bound, an ill-conditioned matrix may arise even before x gets close to x^*. An attractive feature of the augmented Lagrangian method is that σ_i need not approach infinity and may, in fact, be held constant. Instead, θ is varied, such that $\theta \rightarrow \theta^*$, an optimum parameter vector, while satisfying the condition

$$\theta_i^* \sigma_i = \lambda_i^*, \quad i=1, 2, \ldots, m \qquad (3.6.7)$$

If σ_i is sufficiently large each iteration needs to update only θ_i. Further increase in σ_i is only required when the rate of convergence of $x(\theta, \sigma) \rightarrow x^*$ is small. A satisfactory value of σ_i is usually obtained near the early steps of calculation and can be held constant throughout the remaining iterations (Powell, 1978). The augmented Lagrangian function, formed from the Lagrangian function (3.6.5) augmented by the penalty term defined earlier, would then be

$$L_A(x, \lambda, \sigma) = f(x) - \sum_i \lambda_i c_i(x) + \frac{1}{2}\sum_i \sigma_i \left[c_i(x)\right]^2$$

$$= f(x) - \lambda^T c(x) + \frac{1}{2} c(x)^T S\, c(x) \qquad (3.6.8)$$

By letting

$$\theta_i = \frac{\lambda_i}{\sigma_i}, \quad i=1, 2, \ldots, m \tag{3.6.9}$$

and expanding (3.6.6) we get

$$P(\mathbf{x}, \theta, \sigma) = f(\mathbf{x}) + \frac{1}{2}\left\{\sum_i \sigma_i \left[c_i^2(\mathbf{x}) - 2c_i(\mathbf{x})\frac{\lambda_i}{\sigma_i} + \frac{\lambda_i^2}{\sigma_i^2}\right]\right\}$$

$$= f(\mathbf{x}) + \frac{1}{2}\sum_i \sigma_i [\, c_i(\mathbf{x})\,]^2 - \sum_i \lambda_i [\, c_i(\mathbf{x})\,] + \frac{1}{2}\sum_i \frac{\lambda_i^2}{\sigma_i}$$

$$= L_A(\mathbf{x}, \lambda, \sigma) + \frac{1}{2}\sum_i \frac{\lambda_i^2}{\sigma_i} \tag{3.6.10}$$

Since the second term of the right-hand side of (3.6.10) is not a function of $\mathbf{x}_i$, we can say that $\mathbf{x}(\theta, \sigma) = \mathbf{x}(\lambda, \sigma)$ for any σ as long as (3.6.9) holds. For a well-scaled problem, a single scalar value, say r, can replace all σ_i's in **S** such that **S** = r**I**. As such, considerable reduction in the number of unknowns can be realized.

is this the Identity Matrix

Now, consider the nonlinear programming problem with inequality constraints only

② Inequality Constraints Only.

$$\text{Minimize } f(\mathbf{x}) \tag{3.6.11}$$

$$\text{subject to } h_i(\mathbf{x}) \geq 0, \quad i=1, 2, \ldots, m' \tag{3.6.12}$$

The constraint set (3.6.12) can be modified in the form of (3.6.2) by adding slack variable z_i

$$h_i(\mathbf{x}) - z_i = 0, \quad z_i \geq 0, \quad i=1, 2, \ldots, m' \tag{3.6.13}$$

The set $(\mathbf{x}, \mathbf{z})$ forms the new feasible space. The new augmented Lagrangian function would be (3.6.11) plus

$$\sum_i t_i(\mathbf{x}, \mathbf{z}, \mu, \sigma) = \sum_i -\mu_i [\, h_i(\mathbf{x}) - z_i\,] + \frac{1}{2}\sum_i \sigma_i [\, h_i(\mathbf{x}) - z_i\,]^2 \tag{3.6.14}$$

where i=1, 2, . . . m'. The slack variable z_i can be eliminated from the calculations by performing minimization on the function over **z** (Powell, 1978). Since only (3.6.14) depends on z, by the first order necessary conditions are

$$z_i^* = h_i(x) - \frac{\mu_i}{\sigma_i}, \quad z_i \geq 0$$

$$= \max\left[0, h_i(x) - \frac{\mu_i}{\sigma_i}\right] \tag{3.6.15}$$

(3.6.14) is transformed into

$$\sum_i t_i(x, \mu, \sigma) = \sum_i \begin{cases} -\mu_i h_i(x) + \frac{1}{2}\sigma_i\left[h_i(x)\right]^2, & \text{if } h_i(x) < \frac{\mu_i}{\sigma_i} \\ -\frac{1}{2}\frac{\mu_i^2}{\sigma_i}, & \text{if } h_i(x) \geq \frac{\mu_i}{\sigma_i} \end{cases} \tag{3.6.16}$$

or

$$\sum_i t_i(x, \mu, \sigma) = \frac{1}{2}\sum_i \sigma_i \left\{ \min[\,0, h_i(x) - \mu_i/\sigma_i\,] \right\}^2 - \frac{1}{2}\sum_i \mu_i^2/\sigma_i \tag{3.6.17}$$

When the equality constraints (3.6.2) and inequality constraints (3.6.12) occur concurrently in a nonlinear programming problem, the augmented Lagrangian function becomes

$$L_A(x, \lambda, \mu, \sigma) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x) + \frac{1}{2}\sum_{i=1}^{m} \sigma_i[\,c_i(x)\,]^2$$

$$+ \frac{1}{2}\sum_{i=1}^{m'} \sigma_i \left\{ \min[\,0, h_i(x) - \mu_i/\sigma_i\,] \right\}^2 - \frac{1}{2}\sum_{i=1}^{m'} \mu_i^2/\sigma_i \tag{3.6.18}$$

3.6.2 Optimality Results of the Lagrange Multipliers

μ, σ – inequality constraints
σ – equality constraints

Some important duality results will be discussed in this section, showing the optimum choice of the λ and μ (or σ) parameters which are determined by the maximization problem in terms of these parameters.

The first order necessary condition for x^* to be a local minimum of $L_A(x, \lambda, \sigma)$ is that ∇L_A vanishes at x^*. Deriving ∇L_A from the function in (3.6.8) produces

$$\nabla L_A(x, \lambda, \sigma) = \nabla f(x) - \sum_i \lambda_i \nabla c_i(x) + \sum_i \sigma_i c_i(x) \nabla c_i(x) \qquad (3.6.19)$$

On the other hand, the first order necessary condition for x^* to be a local minimum of the original problem (3.6.1)-(3.6.2) is that it had to satisfy (3.6.3) and (3.6.4). It follows, then, that $\nabla L_A(x^*, \lambda^*, \sigma^*) = 0$.

The next and final step would be to prove that $\nabla^2 L_A(x^*, \lambda^*, \sigma)$ is positive definite, i.e., the second order sufficiency condition of the theorem should hold. By taking the derivative of (3.6.19) with respect to x we obtain

$$\nabla^2 L_A(x, \lambda, \sigma) = \nabla^2 L(x, \lambda, \sigma) + \sum_i \sigma_i [\, c_i(x) \nabla^2 c_i(x) + \nabla c_i(x) \nabla c_i(x)^T] \qquad (3.6.20)$$

where ∇^2 is the second derivative of (3.6.5). At the optimum point x^*, the matrix (3.6.20) becomes

$$\nabla^2 L_A(x^*, \lambda^*, \sigma) = \nabla^2 L(x^*, \lambda^*, \sigma) + \sum_i \sigma_i \nabla c_i(x^*) \nabla c_i(x^*)^T \qquad (3.6.21)$$

Let y be a unit vector orthogonal to $\nabla c\,(x^*)$ then the matrix $\nabla^2 L_A{}^*$ or $\nabla^2 L_A(x^*, \lambda^*, \sigma)$ is positive definite since

$$y^T \nabla^2 L_A^* \, y = y^T \nabla^2 L^* \, y + \sigma [\, y^T \nabla c(x^*) \,]^2 \qquad (3.6.22)$$

If $\nabla^2 L^*$ is not a positive definite matrix and thus $y \nabla c(x^*) \neq 0$, then σ has to be sufficiently large, say $\sigma > \sigma' > 0$ such that the second term on the right hand side of (3.6.21) dominates the negative first term. If this is pursued, $\nabla^2 L_A{}^*$ is positive and the theorem is proved.

The augmented Lagrangian function comprising f (x) plus (3.6.16) or (3.6.17) is discontinuous in its derivatives. A remedy to this problem would be to partition the function into two parts such that

$$I_- = \{ i \mid h_i(x) < \mu_i/\sigma_i \} \tag{3.6.23}$$

$$I_+ = \{ i \mid h_i(x) < \mu_i/\sigma_i \} \tag{3.6.24}$$

where I is a general index set, i=1, 2, . . ., m', and $I = I_- \cup I_+$. The augmented Lagrangian function considering inequality constraints only would be

$$L_A(x, \mu, \sigma) = f(x) + \sum_{i=1}^{m'} \begin{cases} -\mu_i h_i(x) + \frac{1}{2}\sigma_i [h_i(x)]^2, & \text{if } i \in I_- \\ -\frac{1}{2}\mu_i^2/\sigma_i, & \text{if } i \in I_+ \end{cases} \tag{3.6.25}$$

If the second order conditions on problem (3.6.11)-(3.6.12) are satisfied and $\mu = \mu^*$, then there exists a $\sigma' > 0$ such that for all $\sigma \geq \sigma'$, x^* is a local minimum of $L_A(x, \mu, \sigma)$. Consider the first order derivatives of (3.6.25) which are

$$\nabla L_A(x, \mu, \sigma) = \begin{cases} \nabla L(x, \mu) + \sum_{i=1}^{m'} \sigma_i h_i(x) \nabla h_i(x), & \text{if } i \in I_- \\ \nabla f(x), & \text{if } i \in I_+ \end{cases} \tag{3.6.26}$$

where

$$\nabla L(x, \mu) = \nabla f(x) - \sum_{i=1}^{m'} \mu_i \nabla h_i(x), \quad \mu_i \geq 0, \quad \text{for } i \in I_- \tag{3.6.27}$$

The first order necessary conditions for x^* minimize $L_A(x, \mu^*, \sigma)$ can be proved to hold in both sides of the partition. For $i \in I_-$, it follows from (3.6.16) that $z_i^* = 0$ or $h_i(x^*) = 0$ and thus (3.6.26) gives $\nabla L_A(x^*, \mu^*, \sigma) = \nabla L(x^*, \mu^*) = 0$. For $i \in I_+$, the necessary conditions for unconstrained minimization of f (x) implies that $\nabla f(x^*) = 0$ and so does $L_A(x^*, \mu^*, \sigma)$.

Using the Kuhn-Tucker conditions and Theorem 1, $\nabla^2 L_A(x^*, \mu^*, \sigma)$ is positive definite in either of the two cases discussed above. Thus the theorem is proved. (Note that the derivatives of the function are undefined at $h_i(x) = \mu_i/\sigma_i$.) Further details on dual theorems of the function can be found in Fletcher (1975) and Rockafellar (1973).

3.6.3 Updating Formula and Convergence

The following discussion is based on the form of the Augmented Lagrangian function for inequality constraints given in (3.6.25). For the case with equality constraints, however, it can also be implied from (3.6.25) when $i \in I_-$. The general form for the updating formula for multipliers is

$$\mu^{(k+1)} = \mu^{(k)} + \Delta\mu^{(k)} \tag{3.6.28}$$

where the superscripts stand for iteration number. The second term of the equation is continuously modified such that $\mu^{(k+1)} \rightarrow \mu^*$. The first order necessary conditions at optimum for the original problem (3.6.11)-(3.6.12) gives

$$\nabla L(x^*, \mu^*) = \nabla f(x^*) - \sum_i \mu_i^* \nabla h_i(x^*) = 0 \tag{3.6.29}$$

$$\mu_i^* h_i(x^*) = 0, \quad \mu_i^* \geq 0 \tag{3.6.30}$$

The first derivatives of (3.6.25) are defined as

$$\nabla L_A(x, \mu, \sigma) = \begin{cases} \nabla f(x) - \sum_i [\mu_i - \sigma_i h_i(x)] \nabla h_i(x), & \text{if } i \in I_- \\ \nabla f(x), & \text{if } i \in I_+ \end{cases} \tag{3.6.31}$$

When $i \in I_+$, the complementary slackness conditions, $\mu_i h_i(x^*) = 0$, give $\mu_i^* = 0$. Since $\mu^{(k+1)} \rightarrow \mu^*$, substituting $\mu_i^* = 0$ in (3.6.28) for $\mu^{(k+1)}$ implies

$$\Delta\mu^{(k)} = -\mu^{(k)}, \quad \text{for } i \in I_+ \tag{3.6.32}$$

When $i \in I_-$, we can equate (3.6.29) and (3.6.31), cancel similar terms and obtain $\mu_i^* = \mu_i - \sigma_i h_i(x)$. By using (3.6.28) again, the implication becomes

$$\Delta\mu_i^{(k)} = -\sigma_i h_i(x), \quad \text{for } i \in I_- \tag{3.6.33}$$

Equations (3.6.32) and (3.6.33) are the simplest updating formulas which do not require any derivatives. Either updating formula represents a steepest ascent towards the maximum of the dual function of (3.6.26) with a linear rate of convergence.

3.8 Solution of Discrete Optimal Control Problem

3.8.1 Discrete Optimal Control Problem

Consider the following discrete optimal control problem

$$\min z = \min f(x,u) \tag{3.8.1}$$

subject to

$$x_{t+1} = g(x_t, u_t, t) \qquad t = 0, \ldots, T-1 \tag{3.8.2}$$

$$\underline{x}_t \le x_t \le \overline{x}_t \qquad t = 0, \ldots, T \tag{3.8.3}$$

$$\underline{u}_t \le u_t \le \overline{u}_t \qquad t = 0, \ldots, T \tag{3.8.4}$$

where x_t is the column vector of state variable at time t; u_t is the column vector of control variables at time t; $\overline{x}_t$ and $\overline{u}_t$ are column vectors of upper bounds; and $\underline{x}_t$ and $\underline{u}_t$ are column vectors of lower bounds. f and g are assumed continuously differentiable in (x_t, u_t) for each t. the time, t, can only take on finite number of discrete values, t = 0, 1, . . . , T. Equation (3.8.2)

represent the process or simulator equation and equations (3.8.3) and (3.8.4) represents the bound constraints on the state and control variables, respectively.

The structure of the Jacobian of (3.8.2) is shown in Figure (3.8.1). Nonzero elements are only in the unit submatrices and in the submatrices $H_0, \ldots, H_{T-1}$; $K_0, \ldots, K_{T-1}$, where

$$H_t = \frac{\partial g}{\partial x_t} \tag{3.8.5}$$

$$K_t = \frac{\partial g}{\partial u_t} \tag{3.8.6}$$

3.8.2 Reduced Objective Problem

Considering a given **u**, the system of equation (3.8.2) may be solved for a unique **x**, **x**(**u**). This function **x**(**u**) can then be used to eliminate **u** in the objective (3.8.1)to yield a new function

$$F(\mathbf{u}) = f\,(\mathbf{x}(\mathbf{u}), \mathbf{u}) \tag{3.8.7}$$

By the implicit function theorem ( ), **x**(**u**) is continuously differentiable, so that F is a differentiable function of u, referred to as the **reduced objective function.**

Solving the process simulation equations (3.8.2) for a particular set of control variables, **u**, each time these equations need to be evaluated, the reduced optimization problem takes the form

$$\text{Min } f\,(\mathbf{x}(\mathbf{u}), \mathbf{u}) = \text{Min } F(\mathbf{u}) \tag{3.8.8}$$

subject

$$\underline{x}_t \le x_t(u_t) \le \overline{x}_t \tag{3.8.9}$$

$$\underline{u}_t \le u_t \le \overline{u}_t \tag{3.8.10}$$

State (or dependent) variables and the control (or independent) variables are implicitly related through the simulator. In essence the simulator equations are used to express the states in terms the controls yielding a much smaller optimization problem. The reduced gradient $\frac{\partial F}{\partial u}$ where $F(u) = f(x(u), u)$ is required to solve the reduced problem. In order to determine the reduced gradient the following procedure can be used.

<u>Step 1</u> Use the appropriate simulation model solve the simulator (process) equations.

<u>Step 2</u> Solve the following set of linear equations

$$\pi\left[\frac{\partial g}{\partial x}\right] = \frac{\partial f}{\partial x} \tag{3.8.11}$$

or

$$\pi B = \frac{\partial f}{\partial x}$$

for the row vector Lagrange of multiplier (π). This equation is derived from the general reduced gradient equation (?).

<u>Step 3</u> Evaluate the reduced gradient

$$\frac{\partial F}{\partial u} = \frac{\partial f}{\partial u} - \pi \frac{\partial g}{\partial u} \tag{3.8.12}$$

In the above two equations all elements of $\frac{\partial g}{\partial x}$ and $\frac{\partial g}{\partial u}$ are evaluated at some model solution **u** for which the $\frac{\partial F}{\partial u}$ is evaluated. Because the simulator

equations (3.8.2) have a sequential form, $\frac{\partial g}{\partial u}$ is block lower triangular with square nonsingular blocks. The large linear system (3.8.11) decomposes into T smaller sequential systems, which are solved backwards in time. The difference equation for the multipliers are

$$\pi_T \mathbf{B}_T = \frac{\partial f}{\partial x_T} \tag{3.8.13a}$$

$$\pi_t \mathbf{B}_t = \frac{\partial f}{\partial x_t} - \pi_{t+1} \frac{\partial g_{t+1}}{\partial x_t} \qquad t = T\text{-}1, T\text{-}2, \ldots, 1 \tag{3.8.13b}$$

In these equations the matrices $\mathbf{B}_t$ and $\frac{\partial g_{t+1}}{\partial x_t}$ are evaluated using the control and state variables obtained in step 1 when solving the simulator equations and all vectors in (3.8.13 a and b) are row vectors. Then (3.8.13a) is solved for π_T and (3.8.13b) is solved sequentially for $\pi_{T-1}, \pi_{T-2}, \ldots, \pi_1$. Equations (3.8.13a and b) are derived from the general reduced gradient equation $\pi \mathbf{B} = \frac{\partial f}{\partial x}$. The components of the $\frac{\partial F}{\partial u}$ are evaluated by

$$\frac{\partial F}{\partial u_t} = \frac{\partial f}{\partial u_t} - \pi_t \frac{\partial g_t}{\partial u_t} \tag{3.8.14}$$

The dynamic structure of the simulator equations could be of the form

$$\mathbf{g}\,(\mathbf{x}_t, \ldots \mathbf{x}_{t-s}, \mathbf{u}_t, \ldots, \mathbf{u}_{t-c}) = 0 \qquad t = 1, \ldots, T \tag{3.8.15}$$

where $\mathbf{g}$ ( ) is an m-vector of function, assumed differentiable, and s and c are the maximum lags of $\mathbf{x}$ and $\mathbf{u}$, respectively. For many application g has the form $g = -\mathbf{x}_t + h\,(\mathbf{x}_t, \ldots, \mathbf{x}_{t-s}, \mathbf{u}_t, \ldots, \mathbf{u}_{t-c}) = 0$. The difference equations for the Lagrange multipliers are

$$\pi_T \mathbf{B}_T = \frac{\partial F}{\partial x_T} \tag{3.8.16}$$

$$\pi_t \mathbf{B}_t = \frac{\partial F}{\partial x_t} - \sum_{\tau=t+1}^{b} \pi_\tau \mathbf{B}_{\tau,t} \qquad t = T-1, T-2, \ldots, 1 \tag{3.8.17}$$

where

$$b = \min(t+s, T) \tag{3.8.18}$$

$$\mathbf{B}_{\tau,t} = \frac{\partial g_\tau}{\partial x_t} \tag{3.8.19}$$

$$\mathbf{B}_t = \frac{\partial g_t}{\partial x_t} \tag{3.8.20}$$

$\frac{\partial g}{\partial x}$ is nonsingular if and only if all matrices $\mathbf{B}_t$ are nonsingular. The components of $\frac{\partial F}{\partial u}$ are evaluated using

$$\frac{\partial F}{\partial u_t} = \frac{\partial f}{\partial u_t} - \sum_{\tau=t}^{a} \pi_\tau \frac{\partial g_\tau}{\partial u_t} \tag{3.8.21}$$

where

$$a = \min(t-c, T) \tag{3.8.22}$$

If the simulator equations are not simultaneous, each $\mathbf{B}_t$ is triangular so (3.8.16) and (3.8.17) can be solved quickly.

3.8.3 GRG Algorithm to Solve Optimal Control Problem

Consider an optimal control problem of the form

$$\text{Min } z = \text{Min} \sum_{t=1}^{T} f_t (x_t, \ldots, x_{t-s'}, u_t, \ldots, u_{t-c'}) \tag{3.8.23}$$

subject to

$$g_t (x_t, \ldots, x_{t-s}, u_t, \ldots, u_{t-c}) = 0 \qquad t = 1, \ldots, T \tag{3.8.24}$$

$$\underline{u}_t \leq u_t \leq \overline{u}_t \qquad t = 1, \ldots, T \tag{3.8.25}$$

For simplicity, bound constraints on the state variable have been suppressed. The state and control lags, s' and c', for the objective function may differ from s and c for the simulator equations. The vector of functions **g** and objective functions f_t may all be nonlinear and are assumed to be continuously differentiable. The recursive equations (3.8.24) are assumed to have a unique solution $\mathbf{x}_1, \ldots, \mathbf{x}_T$ for any set of control vectors $\mathbf{u}_1, \ldots, \mathbf{u}_T$ satisfying (3.8.24) and for any initial conditions.

Bounds on the state variables may be dealt with by penalty or Augmented Lagrangian methods, which required no basis changes, and consequently simplify the algorithm. Penalty or Lagrangian methods may not be as efficient as methods that deal with state bounds directly.

The algorithm presented by Mantell and Lasdon (1978) is stated as follows

<u>Step 0</u>: Given are the initial control vector $\mathbf{u}^{(i)}$ and all initial values of lagged states and control variables, set k = 0.

<u>Step 1</u>: Simulate the system with $\mathbf{u} = \mathbf{u}^k$ to determine all state variables and the objective value F ($\mathbf{u}^k$).

<u>Step 2</u>: Compute ∇F ($\mathbf{u}^k$) from (3.8.17) and (3.8.21).

Step 3: Check for convergence and stop if convergence criteria are satisfied, otherwise go to step 4.

Step 4: Compute the search direction $\mathbf{d}^k$ using an unconstrained minimization algorithm.

Step 5: Perform a one dimensional search along d^k to find β^k, the step size that minimizes $F(\mathbf{u}^k + \beta \mathbf{d}^k)$ subject to $\beta > 0$ and $\underline{\mathbf{u}} \leq \mathbf{u}^k + \beta \mathbf{d}^k \leq \overline{\mathbf{u}}$. For each value of β in the search it is required to simulate the system by solving the simulator equation (3.8.24), compute the objective, and possibly compute the reduced gradient.

Step 6: Set $\mathbf{u}^{k+1} = \mathbf{u}^k + \beta^k \mathbf{d}^k$

Step 7: Replace k by k + 1 and return to step 3 (to step 2 if the reduced gradient is not computed in the one dimensional search).

REFERENCES

Buys, J.D., Dual Algorithms for Constrained Optimization Problems, Ph.D. Thesis, University of Leiden, Netherlands, 1972.

Cooper, L.L. and M.W. Cooper, Introduction to Dynamic Programming, Pergamon Press, Elmsford, N.Y., 1981.

Chow, V.T., 1975

Denardo, E. V., Dynamic Programming Theory and Applications, Prentice Hall, Englewood Cliffs, N.J., 1982.

Dennis, J.E. and R.B. Schnable, Numerical Methods for Unconstrained Optimization, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

Dreyfus, S. and A. Law, The Art and Theory of Dynamic Programming, Academic Press, New York, 1977.

Edgar, T.F. and D. M. Himmelblau, Optimization of Chemical Processes, McGraw-Hill, Inc., New York, 1988.

Fletcher, R., Practical Methods of Optimization, Vol. 1, Unconstrained Optimization, John Wiley & Sons, New York, 1980.

Gill, P.E., W. Murray and M.H. Wright, Practical Optimization, Academic Press, London and New York, 1981.

Himmelblau, D.M., Applied Nonlinear Programming, McGraw-Hill, Inc., 1972.

Lasdon, L.S., R.L. Fox and M.W. Ratner, Nonlinear optimization using the generalized reduced gradient method, Revue Francaise d'Automatique, Informatique et Recherche Operationnelle, V-3, pp. 73-104, November 1974.

Lasdon, L.S., A.D. Waren, A. Jain, and M. Ratner, "Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Programming," ACM Transactions on Mathematical Software, Vol. 4, pp. 34-50, 1978.

Lasdon, L.S. and A.D. Waren, "Generalized Reduced Gradient Software for Linearly and Nonlinearly Constrained Problems," in Design and Implementation of Optimization Software, H.J. Greenberg (ed.), Sijthoff and Noordhoff, pp. 363-397, 1978.

Liebman, J.S., L.S. Lasdon, L. Schrage and A. Waren, Modeling and Optimization with GINO, The Scientific Press, Palo Alto, CA, 1986.

Luenberger, D.G., Introduction to Linear and Nonlinear Programming, Addison-Wesley, Reading, Massachusetts, 1984.

McCormick, G.P., Nonlinear Programming: Theory, Algorithms, and Applications, John Wiley and Sons, New York, 1983.

Murtaugh, B.A. and M.A. Saunders, "Large-Scale Linearly Constrained Optimization," Mathematical Programming, Vol. 14, pp. 41-72, 1978.

Murtaugh, B.A. and M.A. Saunders, "MINOS/AUGMENTED User's Manual," Syst. Optimiz. Lab. Tech. Rep. 80-14, 51 pp., Department of Operations Research, Stanford University, Stanford, CA, 1980.

Murtaugh, B.A. and M.A. Saunder, "MINOS 5.0 User's Guide," Syst. Optimiz. Lab. Tech. Rep. 83-20, 118 pp., Department of Operations Research, Stanford University, Stanford, CA, 1983.

Rockafellar, R.T., "A Dual Approach to Solving Nonlinear Programming Problems by Unconstrained Optimization," Math. Prog., 5:354-373, 1973a.

Rockafellar, R.T., "The Multiplier Method of Hestenes and Powell Applied to Convex Programming," SIAM, J. Control and Optimization, 12:268-285, 1973b.

Rockafellar, R.T., "Augmented Lagrangian Multiplier Functions and Duality in Nonconvex Programming," SIAM, J. Applied Math, 12:555-562, 1974.

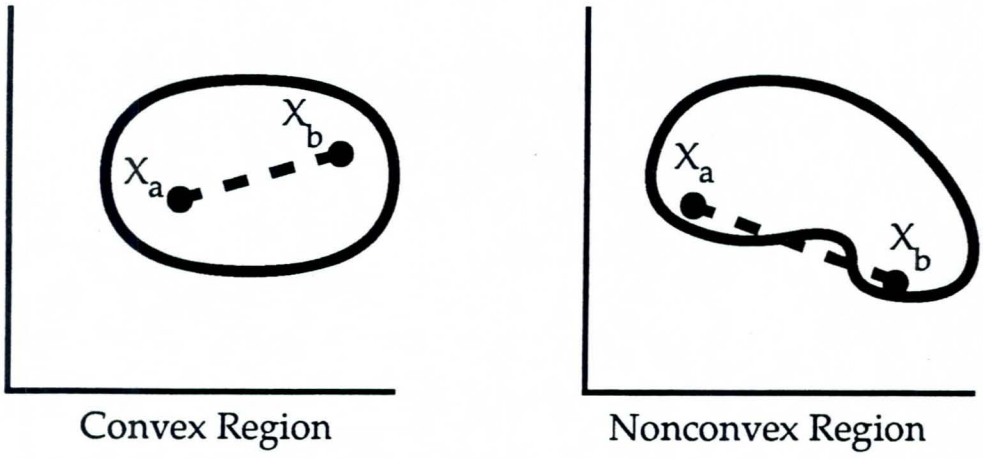


FIGURE 3.1.1
Illustration of Convex and Nonconvex Regions.

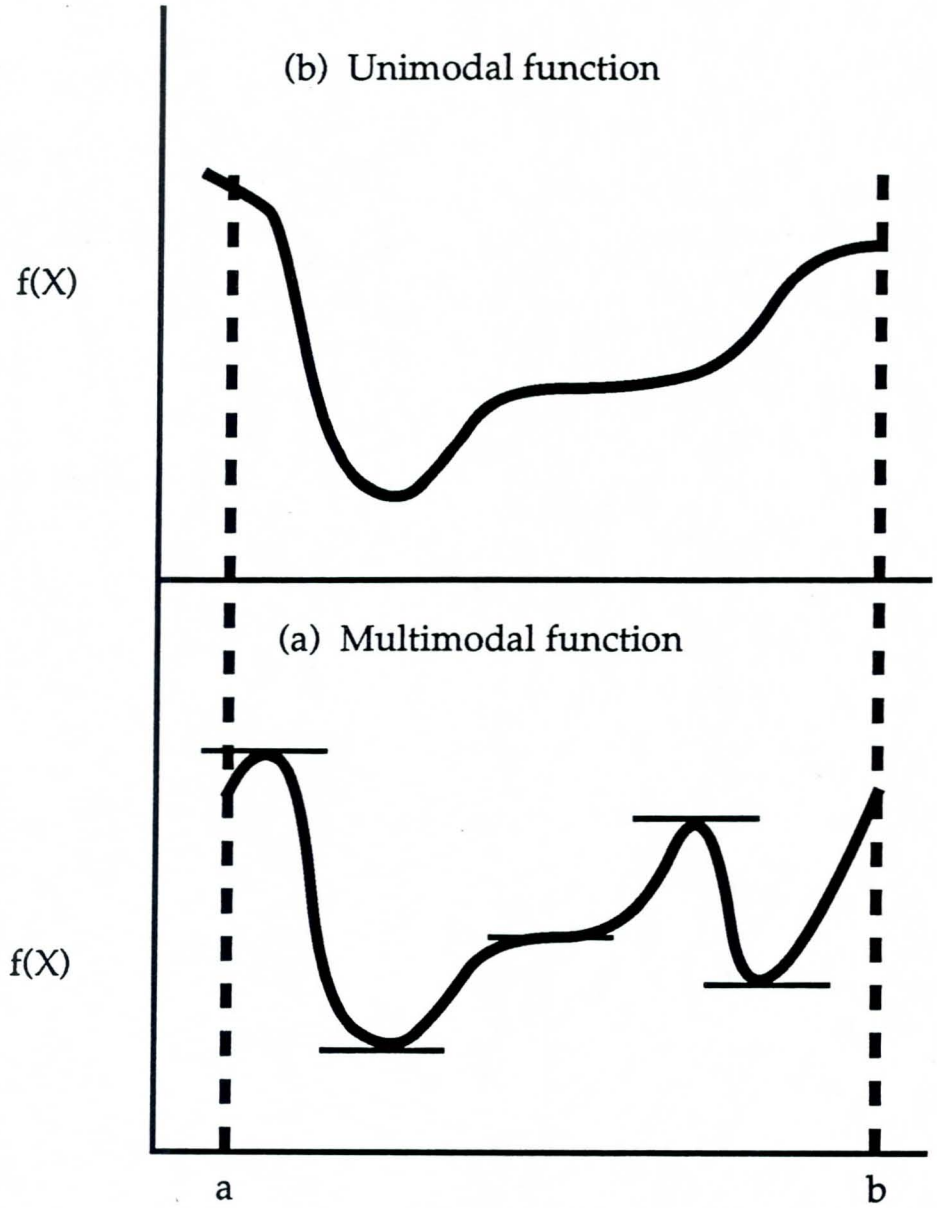


FIGURE 3.2.1
Definition of Unimodal Functions.

3.2.1

Table ~~4.4.1~~ Computation of Search Directions* (Mayp and Tang, 1992)

Search Direction	Definition of Terms
Steepest Descent	
$d^{k+1} = -\nabla f\left(x^{k+1}\right)$	
Conjugate Gradient Methods	
(1) Fletcher-Reeves	
$d^{k+1} = -\nabla f\left(x^{k+1}\right) + a_1 d^k$	$a_1 = \dfrac{\nabla^T f\left(x^{k+1}\right) \nabla f\left(x^{k+1}\right)}{\nabla^T f\left(x^k\right) \nabla f\left(x^k\right)}$ $d^0 = -\nabla f\left(x^0\right)$
(2) Polak-Ribiere	
$d^{k+1} = -\nabla f\left(x^{k+1}\right) + a_2 d^k$	$a_2 = \dfrac{\nabla^T f\left(x^{k+1}\right) Y^{k+1}}{\nabla^T f\left(x^k\right) \nabla f\left(x^k\right)}$ $Y^{k+1} = \nabla f\left(x^{k+1}\right) - \nabla f\left(x^k\right)$ $d^0 = -\nabla f\left(x^0\right)$
(3) 1-Step BFGS	
$d^{k+1} = -\nabla f\left(x^{k+1}\right) + a_3\left(a_4 S^{k+1} + aY^k\right)$	$a_3 = \dfrac{1}{\left(S^{k+1}\right)^T Y^{k+1}}$ $a_4 = -\left(1 + \dfrac{\left(Y^{k+1}\right)^T Y^{k+1}}{\left(S^{k+1}\right)^T Y^{k+1}}\right)\left(S^{k+1}\right)^T \nabla f\left(x^{k+1}\right) + \left(Y^{k+1}\right)^T \nabla f\left(x^{k+1}\right)$ $a_5 = \left(S^{k+1}\right)^T \nabla f\left(x^k\right)$ $S^{k+1} = x^{k+1} - x^k$ $d^0 = -\nabla f\left(x^0\right)$
Quasi-Newton Methods **	
(1) Davidon-Fletcher-Powell DFP Method (Variable Metric Method)	
$d^{k+1} = G^{k+1} \nabla f\left(x^{k+1}\right)$	$G^{k+1} = G^k + \dfrac{S^k\left(S^k\right)^T}{\left(S^k\right)^T Y^k} - \dfrac{G^k Y^k\left(G^k Y^k\right)^T}{\left(Y^k\right)^T G^k Y^k}$
(2) Broyden-Fletcher-Goldfarb-Shanno (BFGS) Method)	

$$d^{k+1} = G^{k+1} \nabla f\left(x^{k+1}\right)$$

$$G^{k+1} = G^{k} + \left(\frac{1 + \left(Y^{k}\right)^{T} G^{k} Y^{k}}{\left(Y^{k}\right)^{T} S^{k}} \right) \frac{S^{k}\left(S^{k}\right)^{T}}{\left(S^{k}\right)^{T} Y^{k}} - \frac{Y^{k}\left(S^{k}\right)^{T} G^{k} + G^{k} S^{k}\left(Y^{k}\right)^{T}}{\left(Y^{k}\right)^{T} S^{k}}$$

(3) <u>Broyden Family</u>

$$G^{\phi} = (1 - \phi)\, G^{DFP} + \phi\, G^{BFGS}$$

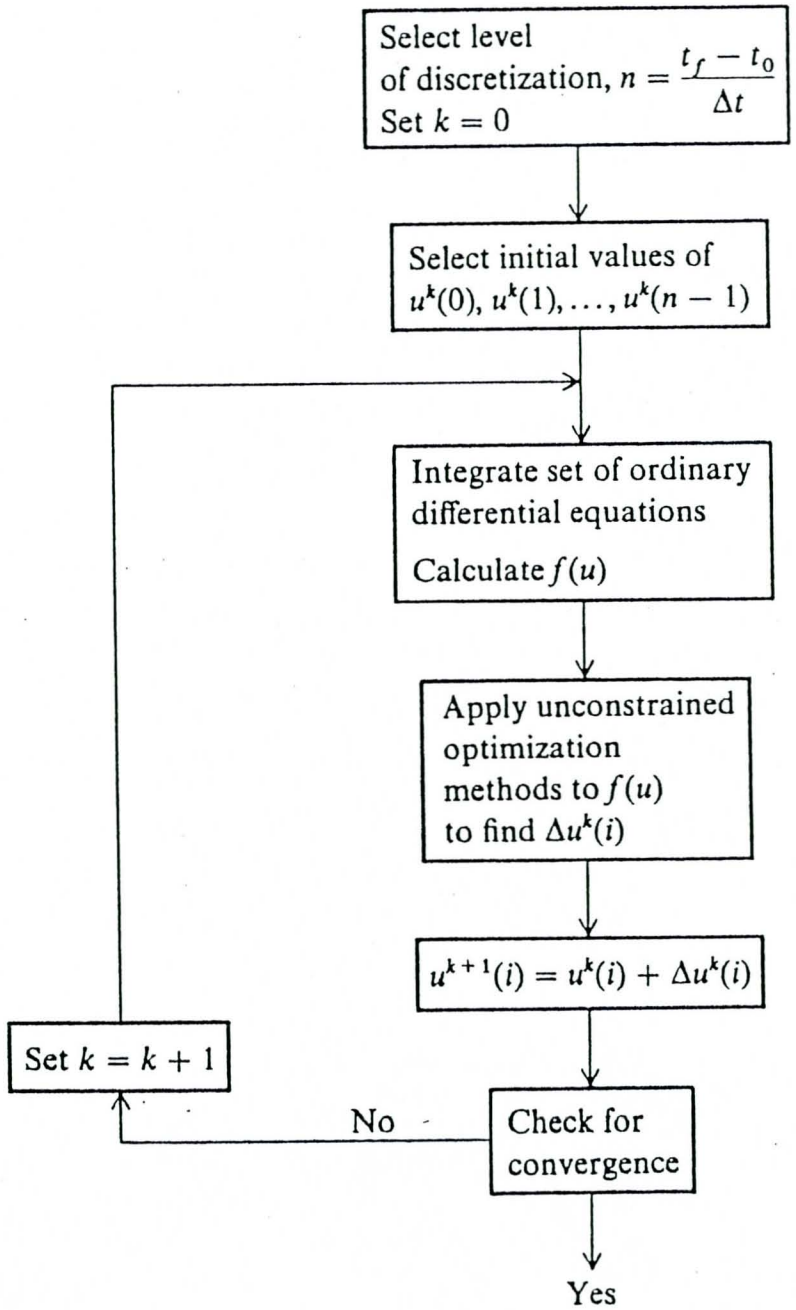
<u>Newton Method</u>

$$d^{k+1} = -H^{-1}\left(x^{k}\right) \nabla f\left(x^{k}\right)$$

---

*Formulas for other search directions can be found in Luenberger (1984).

** $G^0 = 0$



3.7.1

Figure 8.10 Calculation of optimal control via unconstrained optimization combined with iterative solution of the differential equations comprising the model. (Edgar and Himmelblau, 1988)

3-48

$$
\begin{array}{cccccccccccc}
x^0 & x^1 & x^2 & x^3 & & x^{T-1} & x^T & u^0 & u^1 & & u^{T-2} & u^{T-1} \\
\hline
H_0 & \begin{matrix}1 & & \\ & 1 & \\ & & \ddots & \\ & & & 1\end{matrix} & & & & & & K_0 & & & & \\
\hline
 & H_1 & \begin{matrix}1 & & \\ & 1 & \\ & & \ddots & \\ & & & 1\end{matrix} & & & & & K_1 & & & \\
\hline
 & & H_2 & \begin{matrix}1 & & \\ & 1 & \\ & & \ddots & \\ & & & 1\end{matrix} & & & & & & \ddots & & \\
\hline
 & & & & \ddots & & & & & & \ddots & \\
\hline
 & & & & & \begin{matrix}1 & & \\ & 1 & \\ & & \ddots & \\ & & & 1\end{matrix} & & & & & K_{T-2} & \\
\hline
 & & & & & H_{T-1} & \begin{matrix}1 & & \\ & 1 & \\ & & \ddots & \\ & & & 1\end{matrix} & & & & & K_{T-1}
\end{array}
$$

Fig. ~~2.~~ 3.8.1

CHAPTER 4

GROUNDWATER MANAGEMENT SYSTEMS

4.1 Problem Identification

4.2 Problem Formulation

4.2.1 Aquifer Model

4.2.2 Constraints

4.2.3 Objective Function

4.3 Problem Solution

4.3.1 Overview

4.3.2 The Reduced Problem

4.3.3 Computing the Reduced Gradient

4.3.4 Satisfying the Head Bounds Using an Augmented Lagrangian Function

4.3.5 Solution Using the Code GRG2

4.4 Application

References

Appendix

4.A Computation of Basis Matrix Elements

CHAPTER 4

GROUNDWATER MANAGEMENT SYSTEMS

4.1 Problem Identification

Aquifer simulation models have been used to examine the effects of various groundwater management strategies. Use has primarily been of the "case study" or "what if" type. The analyst specifies certain quantities and the model predicts the technical and perhaps economic consequences of this choice. The analyst evaluates these consequences and uses his judgement and intuition to specify the next case.

Optimization methods have been used in groundwater management for more than a decade with some success. Most uses focused on explicitly combining simulation and optimization, resulting in so-called simulation-management models. Gorelick [1983] reviewed these models and classified hydraulic management models into two major approaches: embedding and use of a unit response matrix or an "algebraic technological function" (ATF). Embedding incorporates the equations of the simulation model(represented as a set of difference equations) directly into the optimization problem to be solved. This method has limited applications and is mostly used in groundwater hydraulic management, since the optimization problem quickly becomes too large to solve by available algorithms when a large scale aquifer, especially unconfined, is considered. Previous work based on this approach includes Aguado et al. [1974], Aguado and Remson [1980], Willis and Newman [1977], Aguado et al. [1977], Remson and Gorelick [1980], and Willis and Liu [1984].

The ATF approach generates a unit response matrix by solving the simulation model several times, each with unit pumpage at a single pumping node. Superposition is used to determine the total drawdowns.

This yields a smaller optimization problem, but the method has two major limitations. It is exact only for a confined aquifer but has good accuracy for an unconfined aquifer with relatively small drawdowns compared to the aquifer thickness. A drawdown correction method may be used to improve accuracy for an unconfined aquifer with larger drawdowns, but acceptable accuracy can be guaranteed. In addition, the response matrix must be recomputed when exogenous factors such as aquifer boundary conditions or potential well locations change. An alternative is to treat these factors as decision variables and constraints are included in the optimization problem. Work stemming from this approach includes that by Maddock [1972, 1974], Maddock and Haimes [1975], Morel-Seytoux [1975], Morel-Seytoux and Daly [1975], Morel-Seytoux et al. [1980], Illangasakare and Morel-Seytoux [1982], Heidari [1982], and Willis [1984].

Another approach has been to solve an optimal control problem by interfacing a simulation with an optimizer. The simulator essentially solves the simulator implicitly for the optimizer. Gorelick el al. [1984] applied this method to an aquifer reclamation design to overcome the nonlinearities incurred by the contaminant transport equations. In effect, the dynamic Jacobian matrix, required by the projected Lagrangian method in solving the optimization problem, was determined via forward or central finite differencing, with the contaminant transport simulation used to provide the function values needed in the differencing. This is closely related to the approach described here. We use an analytic rather than differencing approach for computing these same partial derivatives. The possibility of doing this is mentioned in the above reference. However, the hydraulic response was handled by the ATF method.

The work described here attempts to obtain the generality of the hydraulic simulation-management model in combining simulation and optimization to solve the optimal control problem. The overall problem is viewed as one of discrete time optimal control where variables describing the aquifer system are divided into the system state (head) and control (pumpage). By expressing head as an implicit function of pumpage, the model constraints are conceptually eliminated, yielding a smaller reduced problem involving only the pumpage variables. Head bounds are incorporated into the objective using an augmented Lagrangian algorithm as described in Chapter 3. This requires the solution of a set of linear difference equations backward in time and has major speed and accuracy advantages over finite differencing.

4.2 Problem Formulation

4.2.1 Aquifer Model

For nonsteady state heterogeneous anisotropic groundwater flow in saturated media the partial differential equation governing the two-dimensional case is

$$\frac{\partial}{\partial x_i}\left(T_{ij}\frac{\partial h}{\partial x_j}\right) = S\frac{\partial h}{\partial t} + W \qquad i, j = 1, 2 \qquad (4.2.1)$$

where T_{ij} = transmissivity tensor; h = hydraulic head; W = volume flux per unit area; S = storage coefficient; x_i, x_j = Cartesian coordinates; and t = time. For numerical solution using finite difference methods, the aquifer is divided into T periods which need not be of equal length. For the finite difference

grid shown in Figure 4.2.1. The discretization used here leads to the following system of difference equations:

$$A_{ij}h_{ij-1,t} + B_{ij}h_{i,j+1,t} + C_{ij}h_{i-1,j,t} + D_{ij}h_{i+1,j,t}$$
$$- (A_{ij} + B_{ij} + C_{ij} + D_{ij} + F_{ij} + R_{ij})h_{i,j,t} + F_{ij}h_{i,j,t-1}$$
$$- q_{ijt} + R_{ij}RD_{ij} = 0 \text{ for all } i,j \qquad t = 1, \ldots, T \tag{4.2.2}$$

In the above, h_{ijt} = head at cell (i, j) at the end of time period t, q_{ijt} = pumpage (if positive) or recharge (if negative), R_{ij} = spring water constant, RD_{ij} = minimum head for spring water to occur, F_{ij} = a coefficient which depends on storativity or specific yield, and A_{ij}, B_{ij}, C_{ij}, D_{ij} = coefficients which depend on the transmissivity for cells adjacent to (j,j). Coefficients A, B, C and D can be expressed in terms of aquifer permeability as follows:

$$A_{i,j} = (TH_{i,j}\Delta x_{j-1} + TH_{i,j-1}\Delta x_j)\frac{2\Delta y_i PX_{i,j-1}}{(\Delta x_{j-1} + \Delta x_j)^2} \tag{4.2.3}$$

$$B_{i,j} = (TH_{i,j}\Delta x_{j+1} + TH_{i,j+1}\Delta x_j)\frac{2\Delta y_i PX_{i,j}}{(\Delta x_{j+1} + \Delta x_j)^2} \tag{4.2.4}$$

$$C_{i,j} = (TH_{i,j}\Delta y_{j-1} + TH_{i-1,j}\Delta y_i)\frac{2\Delta x_j Py_{i-1,j}}{(\Delta y_{j-1} + \Delta y_i)^2} \tag{4.2.5}$$

$$D_{i,j} = (TH_{i,j}\Delta y_{j+1} + TH_{i+1,j}\Delta y_i)\frac{2\Delta x_j Py_{i,j}}{(\Delta y_{j+1} + \Delta y_i)^2} \tag{4.2.6}$$

where

$PX_{i,j}$ = aquifer permeability between node (i,j) and (i, j+1);

$PY_{i,j}$ = aquifer permeability between node (i,j) and (i+1, j);

$TH_{i,j}$ = aquifer thickness for node (i,j) at time step t, and

Δx_j, Δy_i = the grid size of the cell (i, j).

The expressions (4.2.3) - (4.2.6) are valid for both artesian and water table conditions, only the thickness terms are defined differently. For a cell (i,j) with water table conditions, the thickness can be computed from,

$$TH_{i,j} = h_{i,j,t} - BOT_{i,j} \tag{4.2.7}$$

and for artesian conditions

$$TH_{i,j} = TOP_{i,j} - BOT_{i,j} \tag{4.2.8}$$

where $TOP_{i,j}$ and $BOT_{i,j}$ are the average elevations at the top and bottom of the aquifer at cell (i, j), respectively. Similarly, the coefficient F is given as

$$F = S_{i,j} \Delta x_j \Delta y_i / \Delta t \tag{4.2.9}$$

where $S_{i,j}$ is either the storage coefficient or the specific yield depending upon the condition of the cell (i, j), and Δt is the time step increment. Under water table conditions, the thickness terms defined in (4.2.7) will cause the system of equations to be nonlinear in terms of the hydraulic head.

The alternating direction implicit (ADI) method to solve the system of equations (4.2.2). The method involves iteratively solving the simultaneous equations by first, for a given time increment, reducing a large set of the equations down to a number of small sets. This is done by solving the node equations using Gauss elimination of an individual column of the model while all terms related to the node in adjacent columns are held constant. The set of column equations is then implicit in the direction along the column and explicit in the direction orthogonal to the column alignment. The solution of the set of column equations is then a straight forward process of back substitution.

After all column equations have been processed column by column, attention is focused on solving the node equations again by Gauss

elimination of an individual row while all items related to adjacent rows are held constant. Finally, after all equations have been solved row by row, an iteration has been completed. The above process is repeated a sufficient number of times to achieve convergence, and this completes the computations for the given time step. The solution is said to converge if the differences between row and column solutions is not greater than the tolerance limit set forth. The computed heads are then used as the initial conditions for the next time step. This total process is repeated for successive time increments with unconditional stability regardless of the size of the time increment. More details on how to rearrange the variables and equations can be found in Prickett and Lonnquist (1971).

The coefficients A_{ij}, B_{ij}, C_{ij}, and D_{ij} are linear functions of the thickness of cell (i, j) and the thickness of one of the adjacent cells. For artesian conditions, this thickness is a known constant, so if cell (i, j) and its neighbors are artesian, the (i, j) equation of (4.2.2) is linear for all t. For water table conditions, the thickness of cell (i, j) is $h_{ijt} - BOT_{ij}$, where BOT_{ij} is the average elevation of the bottom of the aquifer at cell (i, j). Then (4.2.2) involves products of heads and is nonlinear.

4.2.2 Constraints

Demand Schedule. It is assumed that the flow rates over specified time periods from all wells must either equal specified values or lie within a specified range. If ω is the set of all cells with pumpage, the former restriction is expressed as

$$\sum_{(i,j) \in \omega} q_{ijt} = d_t \qquad t = 1, \ldots, T \qquad (4.2.10)$$

while the latter one is

$$\underline{d}_t \le \sum_{(i,j)\in\omega} q_{ijt} \le \bar{d}_t \qquad t = 1, \ldots, T \qquad (4.2.10)$$

where $\underline{d}_t$ represents the lower bound on demand for time period t and d_t is the upper bound.

Flow Bounds. The flow bound constraints for recharge and pumpage have the form

$$\underline{q}_{ijt} \le q_{ijt} \le \bar{q}_{ijt} \qquad (i,j) \in \omega \qquad (4.2.11)$$

where the barred quantities are specified limits on the pumpage or recharge. If $\underline{q}_{ijt}$ is zero, this permits no recharge, while a positive lower limit forces pumpage to at least to this level.The expression $\bar{q}_{ijt}$ represents pumpage capacity if positive, while $\bar{q}_{ijt} = 0$ and $\underline{q}_{ijt} < 0$ provide for a limited recharge capability.

Head Bounds : The head bounds are expressed as

$$\underline{h}_{ijt} \le h_{ijt} \le \bar{h}_{ijt} \qquad (i,j,t) \in S \qquad (4.2.13)$$

where S is a subset of cells and time periods where the head is to be controlled. Examples include reducing heads below specified levels in dewatering problems, for maintaining heads above certain levels at springs in acquifer management problems, or insuring that computed heads do not exceed the ground surface for water table conditions.

Groundwater Flow Equations. The difference equations (4.2.2) relating the heads and the well flows in the aquifer are also constraints of the optimization. In the solution approach described in the next section, these equations are used to solve for the heads given the well flows, eliminating the heads and reducing the problem to one involving only the well flows as decision variables. Constraints of this reduction problem will be the demands (4.2.10) or (4.2.11), flow bounds (4.2.12) and head bounds (4.2.13).

4.2.3 Objective Function

Any continuous function of **h** and **q** can be used as an objective function. However for demonstration purposes, two objective functions are presented here. One is to maximize the sum of the heads at all pumping nodes over all time periods, i.e.,

$$\text{Maximize sumh} = \sum_{(i,j)\in\omega} \sum_{t=1}^{T} h_{ijt} \tag{4.2.14}$$

In conjunction with the demand constraints (3) or (4) this objective meets demands while maintaining maximum acquifer potential. The second, used in dewatering problems, is to minimize the pumpage,

$$\text{Minimize sumq} = \sum_{(i,j)\in\omega} \sum_{t=1}^{T} q_{ijt} \tag{4.2.15}$$

4.3 <u>Problem Solution</u>

4.3.1 Overview

The solution methods described here were designed to work with existing aquifer simulation programs. This is a desirable feature in making maximal use of existing technology, and any improvements or changes in the simulation model are automatically incorporated into the optimization scheme.

Aquifer simulators solve for heads and perhaps pollutant concentrations given certain controllable variables. In the simulator used here, the head is computed given the well flows. This allows the constraint and objective function of any aquifer model problems to be viewed as functions of only these controllable variables. Since there are relatively few controllable variables, the resulting problem is easier to solve. The major

remaining difficulty is to compute first partial derivatives of the objective and constant functions with respect to the controllable variables. These derivatives can be computed in significantly less time than is required to perform a simulation. Once they are determined, several efficient nonlinear optimization routines are available to solve the problems. These ideas are general and can be applied to any aquifer, modelling both water quantity and quality.

4.3.2 The Reduced Problem

The system of nonlinear difference equations (2) can be solved for the heads h_{ijt} given well flows q_{ijt} (and initial and boundary conditions). Let q be the vector of all well flows in all time periods, and define $h_{ijt}(q)$ as the heads which satisfy these difference equations when the well flows have the values given by q. For purposes of illustration, let the objective function be the sum of the heads at the pumping nodes, hsum, given by (4.2.14). Since each head h_{ijt} is a function of q, hsum is a function of q also, expressed as hsum(q) :

$$\overline{\text{hsum}}(q) = \sum_{(i,j)\in\omega} \sum_{t=1}^{T} h_{ijt}(q) \tag{4.3.1}$$

Similarly, the head bounds (4.2.13) are functions of q also, rewritten as

$$\underline{h}_{ijt} \le h_{ijt}(q) \le \overline{h}_{ijt} \qquad (i,j,t) \in S \tag{4.3.2}$$

Again, for purposes of illustration, let the demand constraints be equalities as in (4.2.3). These involve only the well flows q_{ijt} and are rewritten here along with the head bounds:

$$\sum_{(i,j)\in\omega} q_{ijt} = d_t \qquad t = 1,\ldots,T \tag{4.3.3}$$

$$\underline{q}_{ijt} \le q_{ijt} \le \overline{q}_{ijt} \qquad (i,j) \in \omega \quad t = 1,\ldots,T \tag{4.3.4}$$

The problem of maximizing $\overline{\text{hsum}}$, (4.3.1), subject to the head bounds (4.3.2), demand constraints (4.3.3), and flow bounds (4.3.4) is called the reduced problem. It involves only the well flows, and is much smaller than the original problem. Many head variables have been eliminated, as have the aquifer model equations (4.2.2). However, the remaining heads $h_{ijt}(q)$ for $\{ (i, j) \in \omega\} \cup \{ (i, j, t) \in S\}$ in (4.3.1) and (4.3.2) are implicit, possibly nonlinear functions of the well flows, q. The simulation model is used to solve for the implicit function value. Optimization methods require values of the objective and constraint functions and their derivatives with respect to each well flow variable. We focus now on how these derivatives can be computed efficiently.

4.3.3 Computing the Reduced Gradient

Consider the computation of the gradient of the head sum, $\overline{\nabla \text{hsum}}(q)$, which is called a reduced gradient. The function hsum is an implicit function of q through the groundwater simulation equations (4.2.2). However, the time-staged structure of these equations leads to an efficient procedure for computing $\overline{\nabla \text{hsum}}$. Procedures of this type have been used to compute the reduced gradients of objective functions defined in econometric models, which are also systems of implicit nonlinear difference equations. Details are given by Mantell and Lasdon [1978] and Norman et al. [1982]. To apply these results to the problem at hand, some additional notation is needed.

Let h_t be the vector of all heads at time t, with components h_{ijt}, and write the aquifer model difference equations (2) in vector form as

$$g_t(h_t, h_{t-1}, q_t) = 0 \qquad t = 1, \ldots, T \tag{4.3.5}$$

Also defined are the following matrices of partial derivatives of the model equations with respect to current and lagged heads :

$$B_t = \partial g_t / \partial h_t \qquad t = 1, \ldots, T \tag{4.3.6}$$

$$C_{t+1,t} = \partial g_{t+1} / \partial h_t \qquad t = 1, \ldots, T+1 \tag{4.3.7}$$

Finally, let π_t be a row vector of Lagrange multipliers for the model equations (4.3.5). Each π_t has as many components as there are grid blocks in the aquifer discretization. Then the procedure for computing $\overline{\nabla hsum}$ for a given vector of well flows q^+ is

Step 1

Solve the simulator equation (4.3.5) forward in time with $q_t = q_t^+$, yielding heads h_t^+ for t = 1, . . ., T.

Step 2

Solve the following system of linear difference equations backwards in time for the Lagrange multiplier vectors π_t:

$$\pi_T B_T = \partial(hsum) \ / \ \partial h_T \tag{4.3.8}$$

$$\pi_T B_T = \partial(hsum) \ / \ \partial h_T - \pi_{t+1} C_{t+1,T} \qquad t = T\text{-}1, T\text{-}2, \ldots, 1 \tag{4.3.9}$$

In these equations, the matrices B_t and $C_{t+1,\,t}$ must be evaluated using the well flows and heads obtained in step 1, and all vectors in (4.3.8) and (4.3.9) are row vectors. Then (4.3.8) is solved for π_T and (4.3.9) is solved sequentially for $\pi_{T-1}, \pi_{T-2}, \ldots, \pi_1$. Equations (4.3.8) and (4.3.9) are derived from the general reduced gradient equation $\pi B = \partial f / \partial y$ in which π is a (row) vector of Lagrange multipliers, B is the basis matrix, f is the objective function, and y is the vector of basic variables (see Ladson et al. [1978] or Lueberger for a derivation). If all variables h_t are basic and all q_t nonbasic, the time staged structure of (4.3.5) implies a sequential structure for these Lagrange multiplier equations as well.

Step 3

Evaluate the components of $\overline{\nabla hsum}$ by

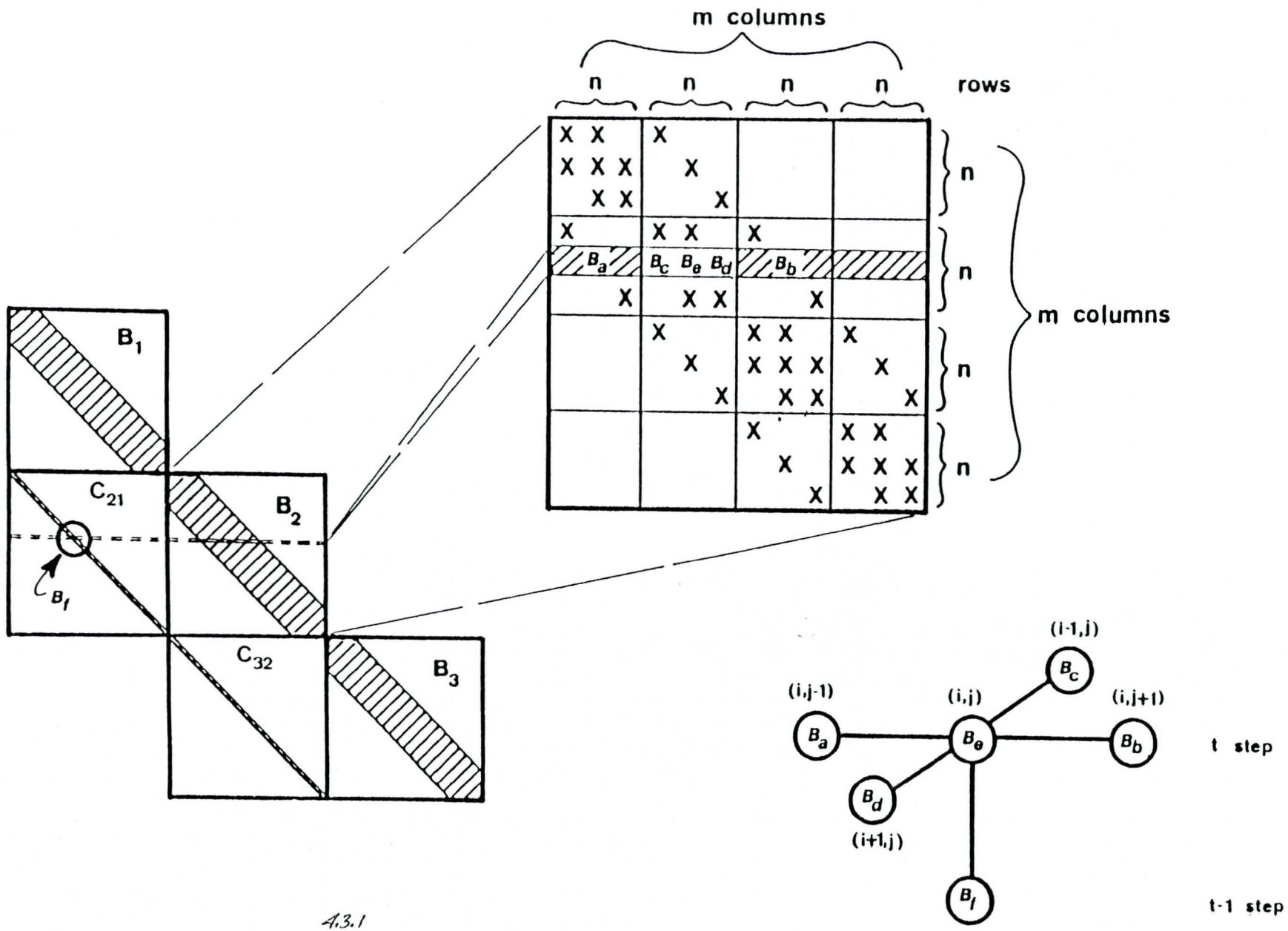
$$\partial(\overline{hsum}) / \partial q_{ijt} = \partial(hsum) / \partial q_{ijt} - \pi_t \partial g_t / \partial q_{ijt} \qquad (4.3.10)$$

The most time consuming part of these computations (apart from the groundwater simulation) is computing the partial derivative matrices B_t and $C_{t+1,t}$ and solving the linear equations (4.3.8) and (4.3.9). For the difference equations (4.2.2), the structure of the matrices B_t and $C_{t+1,t}$ is shown in Figure 4.3.1. B_t is a pentadiagonal matrix, while $C_{t+1,t}$ is diagonal, so the right-hand side of (4.3.~~15~~9) is easy to compute. $C_{t+1,t}$ is constant, and B_t is constant if the entire aquifer is artisian. If some portion has water table conditions, some elements of B_t are linear functions of head. Hence, for the artesian aquifer, the reduced problem is linear, so the reduced gradient of any problem function (either $\overline{hsum}$ or one of the heads $h_{ijt}(q)$) is constant and need be computed only once. Otherwise, the above computations must be performed each time the well flows are changed.

In addition, since each well flow q_{ijt} appears in only one simulator equation (the one for block (i, j) in period t), $\partial g_t / \partial q_{ijt}$ is the negative of a unit vector and $\partial(\overline{hsum}) / \partial q_{ijt}$ is zero, so (4.3.10) becomes

$$\partial(\overline{hsum}) / \partial q_{ijt} = \pi_{ijt} \qquad (4.3.11)$$

Summarizing , while (for water table conditions) the aquifer simulator solves a system of nonlinear difference equations forward in time, yielding the heads, the reduced gradient computation solves the linear system of difference equations (4.3.8) and (4.3.9) backward in time, yielding the Lagrange multipliers π_t. Both systems contain the same number of equations and involve matrices of the same form.



4.3.1

Fig. ~~1~~. Matrix of the constraint coefficients for the n by m finite difference cells with three time steps.

The Lagrange multipliers π_t are more than just an artifice which is useful in computing the reduced gradient. When evaluated at an optimal solution, they supply valuable sensitivity information. For the problem of maximizing hsum considered here, the optimal value of π_{ijt} is equal to the change in the optimal hsum value caused by an additional thousand gallons of water flowing out of cell (i, j) in period t. This applies whether or not there is a well in cell (i, j). Hence, these multipliers could serve to show where new wells should be located, either for pumping or recharge.

4.3.4 Satisfying The Head Bounds Using An Augmented Lagrangian Function

If a portion of the aquifer has water table conditions, the head bounds (4.3.2) is nonlinear. These constraints involve the implicit functions $h_{ijt}(q)$. Computing the reduced gradient of each of these functions requires performing steps 2 and 3, previously discussed. Hence, to compute reduced gradients of all head bound constraints, $N_b * T$ systems of linear equations, arising from (4.3.8) and (4.3.9), must be solved, where N_b is the number of head bound constraints. Instead, an approach which computes only one reduced gradient, requiring the solution of only T linear systems, was chosen. This approach combines the head bounds and the objective into a penalty-like function called an augmented Lagrangian. The procedure is well established in nonlinear programming and is described by Rockfellar [1973], and Fletcher [1975].

Let

$$c_{ijt}(q) = \min\left\{ h_{ijt}(q) - \underline{h}_{ijt}, \overline{h}_{ijt} - h_{ijt}(q) \right\} \tag{4.3.12}$$

Then the head bounds are equivalent to the constraint that $c_{ijt}(q)$ be nonnegative. The appropriate augmented Lagrangian function is

$$L(q,\mu,\sigma) = \overline{hsum}(q) + \tfrac{1}{2}\sigma \sum_{(ijt)\in S} \left[\min\left(0, c_{ijt}(q) - \mu_{ijt}/\sigma\right)\right]^2$$

$$- \tfrac{1}{2} \sum_{(ijt)\in S} (\mu_{ijt})^2 / \sigma \qquad (4.3.13)$$

The parameters μ_{ijt} are Lagrange multipliers for the head bounds, while σ is a positive penalty weight. Consider the Lagrangian problem

$$\text{Maximize } L(q, \mu, \sigma) \qquad (4.3.14)$$

subject to constraints (4.3.3) and (4.3.4)

where the maximization is over q and μ, σ are fixed. If a σ is larger than some threshold value $\bar{\sigma}$ and μ is set equal to the optimal multipliers for the head bounds, μ^*, then any optimal solution for this Lagrangian problem solves the reduced problem (4.3.1) - (4.3.4). This suggests an algorithm (Figure 4.3.1) in which the Lagrangian problem is solved, the parameters μ and σ are adjusted convergence is checked, and the steps are repeated. The multiplier update rule used is

$$\begin{aligned} \mu^+_{ijt} &= \mu_{ijt} - \sigma c_{ijt} \quad \text{if } c_{ijt} \le \mu_{ijt}/\sigma \\ \mu^+_{ijt} &= 0 \qquad c_{ijt} > \mu_{ijt}/\sigma \end{aligned} \qquad (4.3.15)$$

Convergence is tested by checking if the maximum violation of the head bounds is less than a user-supplied tolerance. In general, these violations will be the largest at the start and will diminish as the algorithm proceeds.

If the maximum bound violation has increased over its value at the previous iteration, σ is replaced by 10σ and μ are not updated. If the current largest bound violation is larger than 1/4 of its previous value, σ is replaced by 10σ and the μ are updated. Otherwise, σ is left at its current value when the updating rule (4.3.15) is applied. The algorithm for the methodology is outlined in Figure 4.3.2.

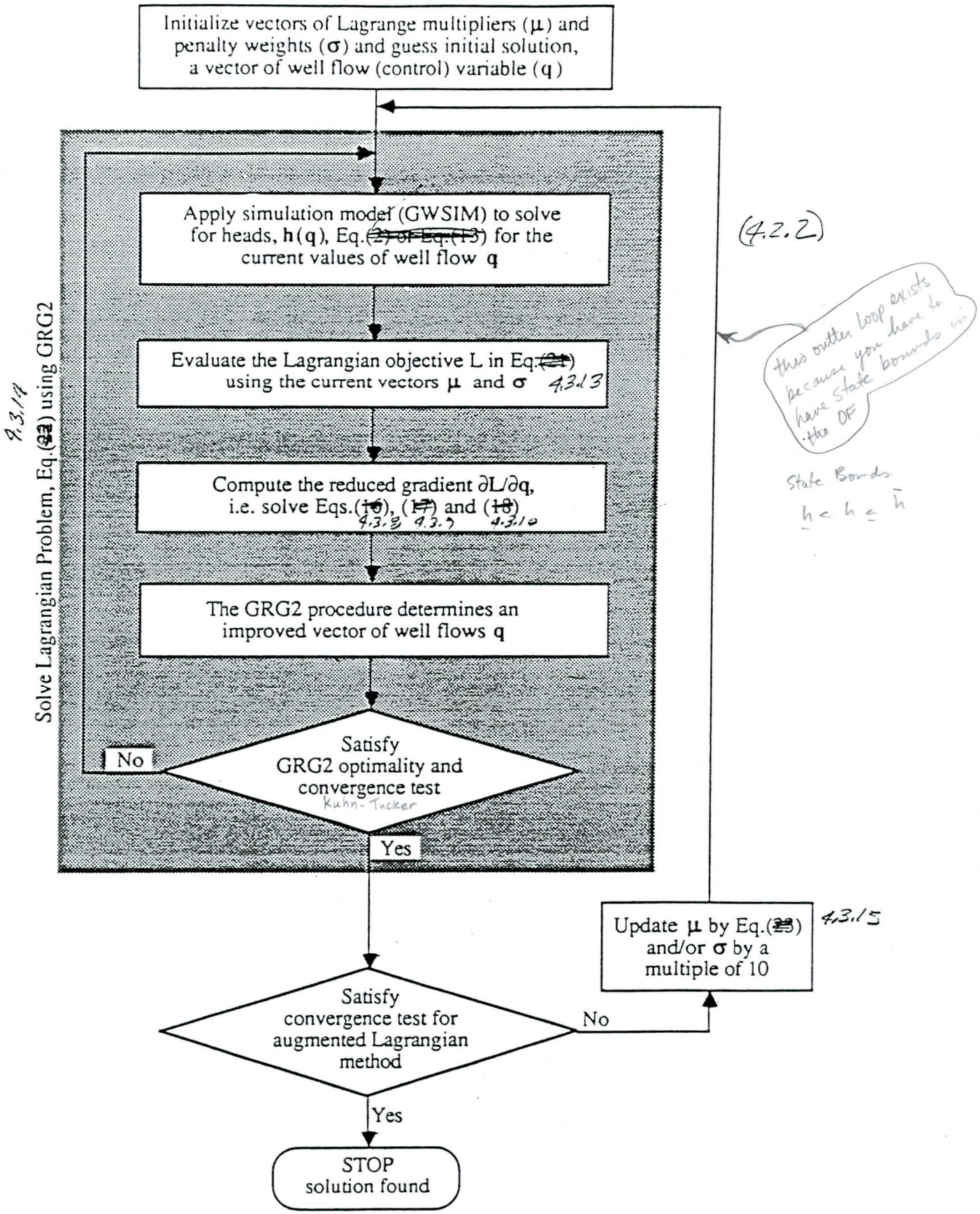


Fig. ~~2~~ Algorithm flow chart.

4.3.2

4.3.5 Solution Using the Code GRG2

The Lagrangian problem (4.3.14) has a nonlinear objective L and linear demand and flow bound constraints. The reduced gradient of L is computed using the previously discussed procedure. To solve the Lagrangian problem (4.3.14), a program called GRG2, described by Ladson et al. [1978], can be used. The algorithm used in this code is of the reduced gradient type, and such methods are particularly effective for linearly constrained problems. GRG2 uses the T demand constraints to eliminate T dependent well flows in terms of the remaining independent ones. These independent flows are varied by the most efficient algorithm available, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method [Fletcher, 1981]. This algorithm uses the gradient of the augmented Lagrangian function to estimate the matrix of second partial derivatives of this function and uses this matrix to compute an efficient search direction. A one-dimensional search procedure using quadratic interpolation is used to determine the distance to move along this direction. The procedure is repeated until one of several stopping criteria, described by Ladson et al. [1978] is met. Of course, if there are logical optima distinct from the global optimum, GRG2 cannot guarantee convergence to the global optimum.

The optimization-groundwater simulation system is referred to as GWMAN. It contains GRG2, the generalized reduced gradient model by Ladson et al. [1978], and GWSIM, a groundwater simulation model developed by the Texas Water Development Board [1974]. GWSIM is a finite difference simulation model which uses the alternating direction implicit method to solve the finite difference equations.

4.4 Application

Wanakule et al. (1985, 1986) developed a model (GWMAN) for determining optimal pumping and recharge for large scale artesian and/or non-artesian aquifers. The model methodology was closely related to the approach used by Gorelick et al. (1984). The overall problem was viewed as one of discrete time optimal control, where variables describing the aquifer system were divided into system state (head) and control (pumpage). By expressing head as an implicit function of pumpage, the model constraints were conceptually eliminated, yielding a smaller reduced problem involving only the pumpage variables. Head bounds were incorporated into the objective using an augmented Lagrangian algorithm. The major contribution of their work was an analytic scheme to compute the reduced gradient needed for optimization. This requires the solution of a set of linear difference equations backwards in time, and has major speed and accuracy advantages over finite differencing.

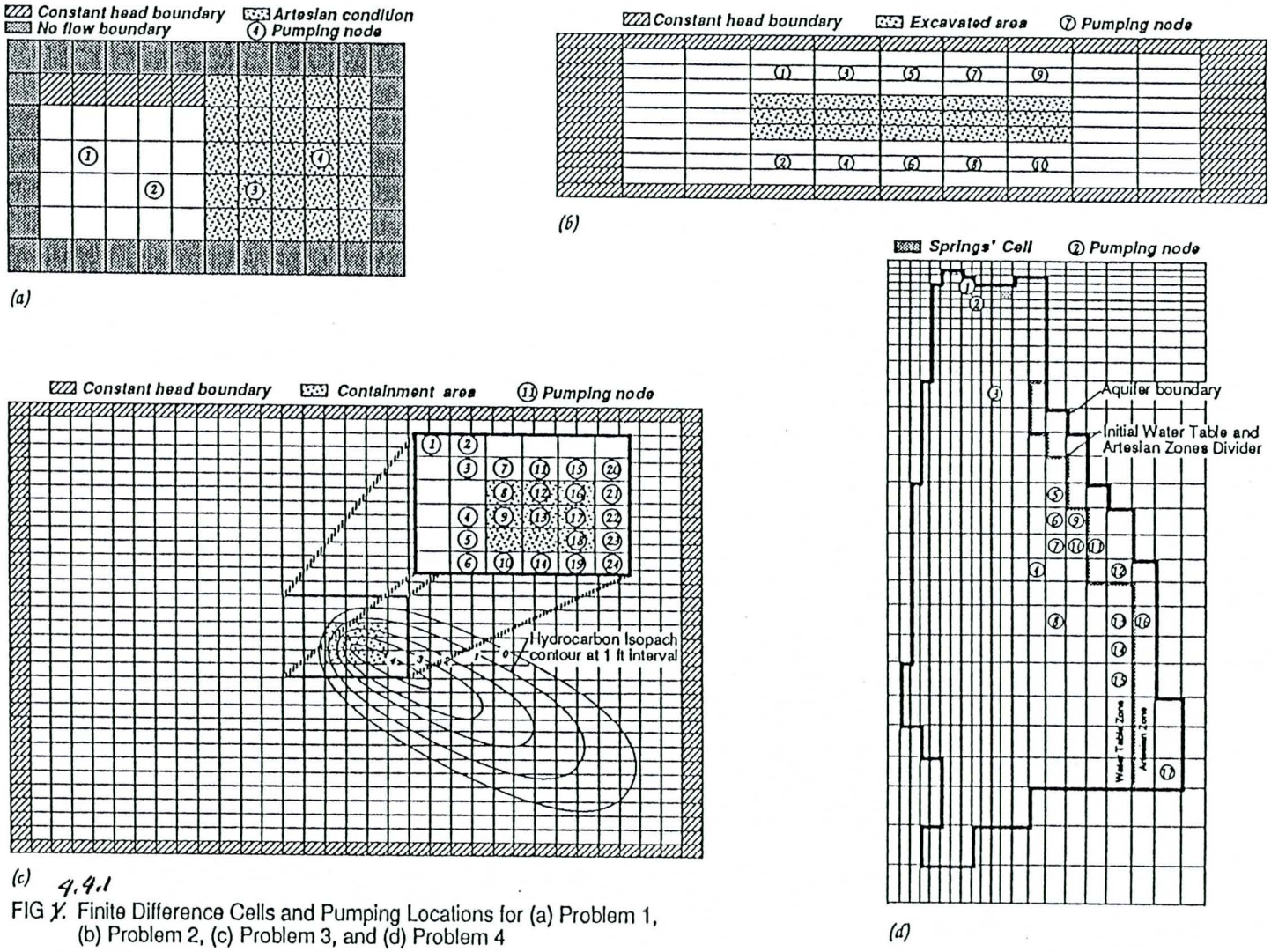
The following paragraphs describe four groundwater management problems used for comparison.

Problem 1 has a grid system configured as shown in Fig. ~~4?~~ 4.4.1a where the grid size is 0.2 mi each side. The bottom elevation is at 150 ft while the thickness in the water table and artesian portions are 100 ft and 50 ft, respectively. Other physical properties are K_x = 600 gal/day/ft^2, K_y = 300 gal/day/ft^2, S_y = 0.1, and S = 0.001. The problem requires maximizing the sum of the heads at the pumping nodes over a period of 5 years, a surrogate objective function for minimizing pumping cost, subject to 2,000 acre-ft/yr demand constraints, lower head bounds at 200 ft and lower flow bound of 200 acre-ft/yr. The problem has 5 one year time intervals with four pumping nodes for each interval.

Problem 2 is the steady state dewatering example taken from Aguado et al. (1977). The grid system (Fig. ~~16~~ 4.4.1b) consists of 121 rectangular cells of 40 m by 10 m in size. The problem is to determine minimum total pumpage that will maintain the water level in a rectangular evacuation area located in the center of the homogeneous isotropic unconfined aquifer at 21 m. The bottom of the aquifer elevation is at 0 m and surrounding constant head elevation is at 36 m. The value of hydraulic conductivity is 10.81 m/d.

Problem 3 is a hypothetical example of a hydrocarbon recovery site where the strategy is to create a containment depression near the center of the hydrocarbon plume. The problem is to determine the optimal water pumpage so that the hydrocarbon plume which is floating on the water layer will be confined to the containment area. The finite difference scheme, setup as shown in Fig. ~~4?~~ 4.4.1c, has a total of 1089 active cells whose dimensions are 180 ft by 120 ft. The aquifer is isotropic nonhomogeneous with an average hydraulic conductivity of about 100 gal/day/ft^2.

Problem 4 is a field application to the Barton Springs - Edwards aquifer in Austin, Texas. It is a limestone aquifer where its main recharge openings were created by steep-angle normal faulting across the stream beds. The problem is set up to determine the optimal yields under long-term average recharge conditions subject to maintaining the spring flows at 25 cfs (0.708 cu m/sec). The finite difference grid system contains 330 active cells whose dimensions are varied from 0.379 by 0.283 mi^2 to 0.95 by 1.51 mi^2 (Fig. ~~4.?~~ 4.4.1d). The total aquifer area includes approximately 150 mi^2. The hydraulic conductivity values vary greatly from 0.1-2.0 ft/day in the outcrop area to 50-1,150 ft/day in the eastern side of the aquifer or the confined zone where the main underground flow channels are located. The groundwater flow



4.4.1

FIG 1. Finite Difference Cells and Pumping Locations for (a) Problem 1, (b) Problem 2, (c) Problem 3, and (d) Problem 4

generally is to the east in the outcrop area and then bends to the north toward the Barton Springs.

Table 4.4.1 compares the results between ~~the~~ a VAX and ~~the~~ a Mac II. The execution time on the Mac II is about 7 time slower while the objective values at optimum obtained from VAX, in almost all cases, are better than those from the Mac II. The improvement of objective values on the Mac II can be achieved by tightening the convergence limit on the optimizer and/or adjusting the magnitude of penalty weights and the initial estimates of Lagrangian multipliers. This, off course, will increase the number of simulation calls and execution time.

The results clearly indicate the potential for implementing GWMAN on microcomputers. Even though it is slow in execution, the advantages in accessibility and low cost computing time can compensate for slowness in most medium size problems.

TABLE 4.3c1. —Comparison of Four Problem Results between the VAX and Mac II Computers.

	Problem 1		Problem 2		Problem 3		Problem 4	
	VAX	Mac II	VAX	Mac II	VAX	Mac II	VAX	Mac II
No. of simulation calls	23	41	207	174	243	140	60	44
Exec. time (minutes)	2.13	18.72	3.45	28.52	224.60	1217.28	2.48	17.08
Objective values	4,490	4,490	105,769	105,765	171.99	172.57	35.826	42.940
Pumpage values	acre-ft/yr		m^3/day		cu ft/day		acre-ft/yr	
Pumping node No. 1	1,400	1,400	13,947.0	13,950.0	0.000	0.000	1.673	1.390
2	200	200	13,658.0	13,655.0	0.000	0.000	1.677	2.688
3	200	200	8,246.7	8,235.2	0.000	0.000	1.720	2.957
4	200	200	8,327.9	8,335.3	0.000	0.000	2.134	2.607
5	1,400	1,400	8,711.3	8,712.1	0.000	0.000	1.759	2.650
6	200	200	8,698.3	8,701.5	0.000	0.000	1.703	2.468
7	200	200	8,286.2	8,287.0	10.029	0.000	1.746	2.368
8	200	200	8,284.7	8,284.8	16.141	15.781	1.791	1.933
9	1,400	1,400	13,805.0	13,802.0	16.141	16.141	1.723	2.454
10	200	200	13,804.0	13,802.0	2.523	4.675	1.726	2.316
11	200	200			0.000	5.898	2.560	3.104
12	200	200			16.141	16.141	1.788	2.176
13	1,400	1,400			16.141	16.141	2.067	2.195
14	200	200			16.141	16.141	3.396	3.384
15	200	200			1.970	12.576	3.396	3.266
16	200	200			16.141	16.141	2.047	2.177
17	1,400	1,400			16.141	16.141	2.920	2.809
18	200	200			16.141	16.141		
19	200	200			16.141	16.141		
20	200	200			0.000	0.000		
21					0.000	0.000		
22					12.203	3.381		
23					0.000	1.131		
24					0.000	0.000		

APPENDIX 4. A

COMPUTATION OF BASIS MATRIX ELEMENTS

This Appendix presents the equations for computing elements of basis matrix and a portion of the Jacobian matrix. Elements of the matrix are the partial derivatives of the groundwater flow system of equations (4.2.2) with respect to the state variable h. Each element is evaluated at the current point where h and q are known. The equations are divided into five groups depending upon the aquifer conditions of a cell under consideration. Investigation of equation (4.2.2) reveals that each row of the matrix should contain utmost six elements. Represent cells (i, j-1), (i, j+1), (i-1, j), (i+1, j), (i, j) at time step t by small letters a, b, c, d, e, respectively, and (i, j) at time (t+1) by f. The configuration of the finite difference scheme for the system of equations (4.2.2) can be viewed as shown in Figure 4.A.1. The figure also shows the positions of the elements in the structured matrix.

CASE I

When all six nodes are under water table conditions, all terms in equations (4.2.2) are nonlinear. The elements of the matrix in a row can be computed from the following expressions;

$$B_a = [(h_e - BOT_e)\Delta x_a + (2h_a - BOT_a)\Delta x_e - h_e \Delta x_e]\frac{2\Delta y_e PX_a}{(\Delta x_a + \Delta x_e)^2} \quad (4.A.1)$$

$$B_b = [(h_e - BOT_e)\Delta x_b + (2h_b - BOT_b)\Delta x_e - h_e \Delta x_e]\frac{2\Delta y_e PX_e}{(\Delta x_b + \Delta x_e)^2} \quad (4.A.2)$$

$$B_c = [(h_e - BOT_e)\Delta y_c + (2h_c - BOT_c)\Delta y_e - h_e \Delta y_e]\frac{2\Delta x_e PY_c}{(\Delta y_c + \Delta y_e)^2} \quad (4.A.3)$$

$$B_d = [(h_e - BOT_e)\Delta y_d + (2h_d - BOT_d)\Delta y_e - h_e \Delta y_e]\frac{2\Delta x_e PY_c}{(\Delta y_d + \Delta y_e)^2} \quad (4.A.4)$$

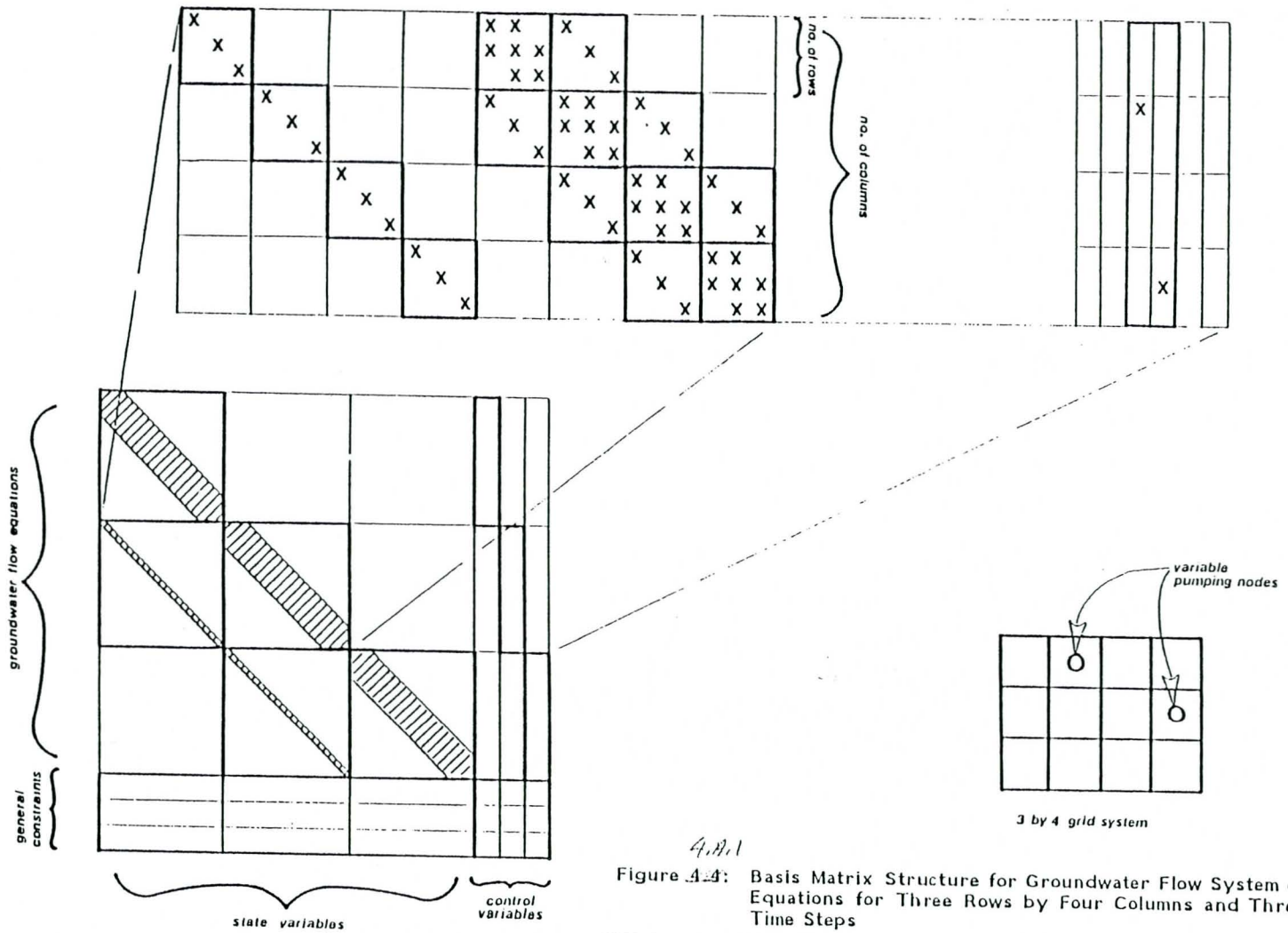


Figure 4.4.1: Basis Matrix Structure for Groundwater Flow System of Equations for Three Rows by Four Columns and Three Time Steps

$$B_e(a) = [h_a \Delta x_a - (2h_e - BOT_e)\Delta x_a + (h_a - BOT_a)\Delta x_e] \frac{2\Delta y_e PX_a}{(\Delta x_a + \Delta x_e)^2} \quad (4.A.5)$$

$$B_e(b) = [h_b \Delta x_b - (2h_e - BOT_e)\Delta x_b + (h_b - BOT_b)\Delta x_e] \frac{2\Delta y_e PX_e}{(\Delta x_b + \Delta x_e)^2} \quad (4.A.6)$$

$$B_e(c) = [h_c \Delta y_c - (2h_e - BOT_e)\Delta y_c - (h_b - BOT_b)\Delta y_e] \frac{2\Delta x_e PY_c}{(\Delta y_c + \Delta y_e)^2} \quad (4.A.7)$$

$$B_e(d) = [h_d \Delta y_d - (2h_e - BOT_e)\Delta y_d - (h_b - BOT_b)\Delta y_e] \frac{2\Delta x_e PY_e}{(\Delta y_d + \Delta y_e)^2} \quad (4.A.8)$$

$$B_e = B_e(a) + B_e(b) + B_e(c) + B_e(d) - B_f - R_e \quad (4.A.9)$$

$$B_f = S_e \Delta x_e \Delta y_e / \Delta t \quad (4.A.10)$$

CASE II

When the center node e is under water table condition but some of the neighbouring nodes are artesian, the equations 4.2.2 are partially nonlinear. The equations are the same as in CASE I, except that the elements corresponding to the artesian nodes are replaced by the following expressions;

$$B_a = [(h_e - BOT_e)\Delta x_a + (TOP_a - BOT_a)\Delta x_e] \frac{2\Delta y_e PX_a}{(\Delta x_a + \Delta x_e)^2} \quad (4.A.11)$$

$$B_b = [(h_e - BOT_e)\Delta x_b + (TOP_b - BOT_b)\Delta x_e] \frac{2\Delta y_e PX_e}{(\Delta x_b + \Delta x_e)^2} \quad (4.A.12)$$

$$B_c = [(h_e - BOT_e)\Delta y_c + (TOP_c - BOT_c)\Delta y_e] \frac{2\Delta x_e PY_c}{(\Delta y_c + \Delta y_e)^2} \quad (4.A.13)$$

$$B_d = [(h_e - BOT_e)\Delta y_d + (TOP_d - BOT_d)\Delta y_e] \frac{2\Delta x_e PY_c}{(\Delta y_d + \Delta y_e)^2} \quad (4.A.14)$$

$$B_e(a) = [h_a \Delta x_a - (2h_e - BOT_e)\Delta x_a - (TOP_a - BOT_a)\Delta x_e] \frac{2\Delta y_e PX_a}{(\Delta x_a + \Delta x_e)^2} \quad (4.A.15)$$

$$B_e(b) = [h_b \Delta x_b - (2h_e - BOT_e)\Delta x_b - (TOP_b - BOT_b)\Delta x_e] \frac{2\Delta y_e PX_e}{(\Delta x_b + \Delta x_e)^2} \quad (4.A.16)$$

$$B_e(c) = [h_c \Delta y_c - (2h_e - BOT_e)\Delta y_c - (TOP_b - BOT_b)\Delta y_e] \frac{2\Delta x_e PY_c}{(\Delta y_c + \Delta y_e)^2} \quad (4.A.17)$$

$$B_e(d) = [h_d \Delta y_d - (2h_e - BOT_e)\Delta y_d - (TOP_b - BOT_b)\Delta y_e] \frac{2\Delta x_e PY_e}{(\Delta y_d + \Delta y_e)^2} \quad (4.A.18)$$

CASE III

When all six nodes are artesian, equations 4.2.2 are linear and their elements in a row can be computed as follows

$$B_a = [(TOP_e - BOT_e)\Delta x_a + (TOP_a - BOT_a)\Delta x_e] \frac{2\Delta y_e PX_a}{(\Delta x_a + \Delta x_e)^2} = -B_e(a) \quad (4.A.19)$$

$$B_b = [(TOP_e - BOT_e)\Delta x_b + (TOP_b - BOT_b)\Delta x_e] \frac{2\Delta y_e PX_e}{(\Delta x_b + \Delta x_e)^2} = -B_e(b) \quad (4.A.20)$$

$$B_c = [(TOP_e - BOT_e)\Delta y_c - (TOP_c - BOT_c)\Delta y_e] \frac{2\Delta x_e PY_c}{(\Delta y_c + \Delta y_e)^2} = -B_e(c) \quad (4.A.21)$$

$$B_d = [(TOP_e - BOT_e)\Delta y_d - (TOP_d - BOT_d)\Delta y_e] \frac{2\Delta x_e PY_e}{(\Delta y_d + \Delta y_e)^2} = -B_e(d) \quad (4.A.22)$$

$$B_e = B_e(a) + B_e(b) + B_e(c) + B_e(d) - B_f - R_e \quad (4.A.23)$$

$$B_f = S_e \Delta x_e \Delta y_e / \Delta t \quad (4.A.24)$$

CASE IV

When the center node e is artesian but some neighboring nodes are under water table conditions, the nonlinear terms appear corresponding to the water table nodes. The computation is the same as CASE III except for the elements at the water table nodes are substituted by the following;

$$B_a = [(TOP_e - BOT_e)\Delta x_a + (2h_a - BOT_a)\Delta x_e - h_e \Delta x_e]\frac{2\Delta y_e PX_a}{(\Delta x_a + \Delta x_e)^2} \quad (4.A.25)$$

$$B_b = [(TOP_e - BOT_e)\Delta x_b + (2h_b - BOT_b)\Delta x_e - h_e \Delta x_e]\frac{2\Delta y_e PX_e}{(\Delta x_b + \Delta x_e)^2} \quad (4.A.26)$$

$$B_c = [(TOP_e - BOT_e)\Delta y_c + (2h_c - BOT_c)\Delta y_e - h_e \Delta y_e]\frac{2\Delta x_e PY_c}{(\Delta y_c + \Delta y_e)^2} \quad (4.A.27)$$

$$B_d = [(TOP_e - BOT_e)\Delta y_d + (2h_d - BOT_d)\Delta y_e - h_e \Delta y_e]\frac{2\Delta x_e PY_e}{(\Delta y_d + \Delta y_e)^2} \quad (4.A.28)$$

$$B_e(a) = -[(TOP_e - BOT_e)\Delta x_a + (h_a - BOT_a)\Delta x_e]\frac{2\Delta y_e PX_a}{(\Delta x_a + \Delta x_e)^2} \quad (4.A.29)$$

$$B_e(b) = -[(TOP_e - BOT_e)\Delta x_b + (h_b - BOT_b)\Delta x_e]\frac{2\Delta y_e PX_e}{(\Delta x_b + \Delta x_e)^2} \quad (4.A.30)$$

$$B_e(c) = -[(TOP_e - BOT_e)\Delta y_c + (h_c - BOT_c)\Delta y_e]\frac{2\Delta x_e PY_c}{(\Delta y_c + \Delta y_e)^2} \quad (4.A.31)$$

$$B_e(d) = -[(TOP_e - BOT_e)\Delta y_d + (h_d - BOT_d)\Delta y_e]\frac{2\Delta x_e PY_e}{(\Delta y_d + \Delta y_e)^2} \quad (4.A.32)$$

CASE V

When the middle node e is a constant head cell, the matrix elements in the corresponding row are set to constant as follows;

$$B_a = B_b = B_c = B_d = B_f = 0; B_e = 1 \tag{4.A.33}$$

This is a consequence of taking derivative of the equation

$$h_e - H = 0 \tag{4.A.34}$$

in place of the groundwater flow equations (4.2.1), where H is the constant head elevation. It follows that the elements $B_a, B_b, B_c,$ and B_d are also zeroes in all four cases above if they represent a constant head node.

REFERENCES

Aguado, E., and I. Remson, Ground-water management with fixed charges, *J. Water Resour. Plann. Manage. Div. Am. Soc. Civ. Eng., 106,* 375-382, 1980.

Aguado, E., I. Remson, M. F. Pikul, and W. A. Thomas, Optimum pumping for aquifer dewatering, *J. Hydraul. Div. Am. Soc. Civ. Eng., 100,* 860-877, 1974.

Aguado, E., N. Sitar, and I. Remson, Sensitivity analysis in aquifer studies, *Water Resour. Res., 13,* 733-737, 1977.

Fletcher, R., An ideal penalty function for constrained optimization, *J. Inst. Math. Its Appl., 15,* 319-342, 1975.

Fletcher, R., *Practical Methods of Optimization,* vol. 1, John Wiley, New York, 1981.

Gorelick, S. M., A review of distributed parameter groundwater management modeling methods, *Water Resour. Res., 19,* 305-319, 1983.

Gorelick, S. M., C. I. Voss, P. E. Gill, W. Murray, M. A. Saunders, and M. M. Wright, Aquifer reclamation design: The use of contaminant transport simulation combined with nonlinear programming, *Water Resour. Res., 20,* 415-427, 1984.

Guyton, W. F., and Associates, Geohydrology of Comal, San Marcos, and Hueco Springs, *Rep. Tex. Dep. Water Resour., 234,* June 1979.

Heidari, M., Application of linear systems theory and linear programming to groundwater management in Kansas, *Water Resour. Bull., 18,* 1003=1012, 1982.

Klemt, W. B., T. R. Knowles, G. R. Elder, and T. W. Sieh, Groundwater resources and model application for the Edwards (Balcones Fault Zone)

Aquifer in the San Antonio Region, Texas, *Rep. Tex. Dep. Water Resour.*, *239*, October 1979.

Knowles, T., GWSIM III -- Groundwater simulation program, Program Document and User's Manual, Tex. Dep. of Water Resour., Austin, 1981.

Lasdon, L. S., A. D. Warren, A. Jain, and M. Ratner, Design and testing of a generalized reduced gradient code for nonlinear programming, *Assoc. Comput. Mach. Trans. Math. Software*, *4*, 34-50, 1978.

Maddock, T., III, Algebraic technological function for a simulation model, *Water Resour. Res.*, *8*, 129-134, 1972.

Maddock, T., III, Nonlinear technological functions for aquifers whose transmissivities vary with drawdown, *Water Resour. Res.*, *10*, 877-881, 1974.

Maddock, T., III, and Y. Y. Haimes, A tax system for groundwater management, *Water Resour. Res.*, *11*, 7-14, 1975.

Mantell, J. and L. S. Lason, A GRG algorithm for econometric control problems, *Ann. Econ. Soc. Manage.*, *6*, 581-597, 1978.

Morel-Seytoux, H. J., A simple case of conjunctive surface-ground water management, *Ground Water*, *13*, 505-515, 1975.

Morel-Seytoux, H. J., and C. J. Daly, A discrete kernel generator for stream-aquifer studies, *Water Resour. Res.*, *11*, 253-260, 1975.

Morel-Seytoux, H. J., G. Peters, R. Young, and T. Illangasekare, Groundwater modeling for management, Paper presented at the International Symposium on Water Resource Systems, Water Resour. Dev. and Training Cent., Univ. of Roorkee, Roorkee, India, 1980.

Norman, A. L., L. S. Lason, and J. K, Hsin, A comparison of methods for solving and optimizing a large nonlinear econometric model, discussion paper, Cent. for Econ. Res., Univ. of Tex., Austin, 1982.

Prickett, T. A., and C. G. Lonnquist, Selected digital computer techniques for groundwater resource evaluation, *Bull. Ill. State Water Surv., 55,* 1971.

Reeves, R. D., R. W. McClay, and M. S. Davis, Records of groundwater recharge, discharge, water levels, and chemical quality of water for the Edwards Aquifer in the San Antonio area, Texas, 1934-80, *Bull. 40,* Edwards Underground Aquifer Dist., U. S. Geol. Surv. and Tex. Dep. of Water Resour., San Antonio, Tex., October 1982.

Remson, I., and S. M. Gorelick, Management models incorporating groundwater variables, in *Operation Research in Agriculture and Water Resources,* edited by D. Yaron, and C. S. Tapiero, North-Holland, Amsterdam, 1980.

Rockafellar, R. T., A dual approach to solving nonlinear programming problems by unconstrained optimization, *Math. Programming, 5,* 354-373, 1973.

Texas Water Development Board, GWSIM -- Groundwater simulation program, Program Document and User's Manual, *UM S7405,* Location, 1974.

Trescott, P. C., G. F. Pinder, and S. P. Larson, Finite-difference model for aquifer simulation in two-dimensionals with results of numerical experiments, in *U. S. Geological Survey Techniques of Water Resources Investigations,* book 7, chap. C1, U. S. Geological Survey, Reston, Va., 1976.

U. S. Bureau of Reclamation, San Antonio--Guadalupe River Basins Study, Special report, Tex. Basin Project, Southwest Regional Office, Amarillo, Tex., November 1978.

Wanakule, N., A model for determining optimal pumping and recharge of large-scale aquifers, Ph.D. Dissertation. Dept. of Civil Eng., Univ. of Tex., Austin, 1984.

Willis, R., A unified approach to regional groundwater management, in *Groundwater Hydraulics, Water Resour. Monogr. 9,* edited by J. S. Rosenshein and G. D. Bennett, AGU, Washington, D. C., 1984.

Willis, R., and P. Liu, Optimization model for ground-water planning, J. *Water Resour. Plann. Manage. Div., Am. Soc. Civ. Eng., 110,* 333-347, 1984.

Willis, R., and B. A. Newman, Management model for groundwater development, *J. Water Resour. Plann. Manage. Div., Am. Soc. Civ. Eng., 13,* 159-171, 1977.

CHAPTER 5

REAL-TIME OPERATION OF RIVER-RESERVOIR SYSTEMS

5.1 Problem Identification

5.2 Problem Formulation

5.2.1 Simulator Equations

5.2.2 Constraints

5.2.3 Objective Functions

5.3 Problem Solution

5.3.1 Overview

5.3.2 The Reduced Problem

5.3.3 Solution of Reduced Problem

5.4 Application

References

5.A Solution of Simulation Model

5.B Computation of Basis Matrix Elements

CHAPTER 5

REAL-TIME OPERATION OF RIVER-RESERVOIR SYSTEMS

5.1 Problem Identification

Real-time operation of multireservoir systems involves various hydrologic, hydraulic, operational, technical, and institutional considerations. For efficient operation, a monitoring system is essential that provides the reservoir operator with the flows and water levels at various points in the river system including upstream extremities, tributaries and major creeks as well as reservoir levels, and precipitation data for the watersheds whose outputs (runoff from rainfall) are not gauged. A flow routing procedure is needed to predict the impacts of observed and/or predicted inflow hydrographs on the downstream parts of the river system. A reservoir operation policy or a methodology is another component which reflects the flood control objectives of the system, the operational and institutional constraints on flood operations, and other system-related considerations. A integral part of these components is the reservoir operation model that predicts the results of a given operation policy for forecasted flood hydrographs.

Flood forecasting in general, and real-time flood forecasting in particular, have always been an important problem in operational hydrology, especially when the operation of storage reservoirs is involved. The forecasting problem, as in most hydrological problems, can be viewed as a system with inputs and outputs. The system output is related to its causative input through a process, either linear or nonlinear. In the reservoir management problem, the system is the river system that includes the main

river and its tributaries, catchments, and natural and manmade structureson the path of the flood waters.The system inputs are inflow hydrographs at the upstream ends of the river system, and runoff from the rainfall (and snowmelt, where applicable) In the intervening catchments. The system outputs are flow rates and/or water levels at control points of the river system. The operations involved are the operations of the reservoir(s) in order to control flood waters. The term '**forecasting**' refers to the prediction of the discharges and water surface elevations at various points of a river system as a result of the observed portion of flood hydrograph.

Multireservoir operation can be characterized by the integrated operation of multiple facilities on river systems for multiple objectives. Flood control is one of the major purposes of many reservoirs in the U.S. Many reservoirs were built several years ago and operation policies were established. However, many of these reservoirs cannot be operated in the manner that they were initially intended to be operated. One of the major reasonsis the uncontrolled urbanization into the floodplains of the rivers and reservoirs. Other reasons are due to inadequate spillway for passing floods, legal constraints, and reduced downstream conveyance capacities.

Many of the reservoir systems are characterized by conditions that result in significant backwater conditions due to gate operation, tributary flows, hurricane surge flows, and tidal conditions, flow constrictions in the rivers. These conditions cannot be described by the use of hydrologic routing methods, and as a result must be described by more accurate hydraulic routing models such as DWOPER, which is based upon a finite difference solution of the Saint Venant equations. Also, flows through reservoirs having considerable length are not properly predicted by the simple hydrologic

methods, particularly when the inflow hydrograph is a flash flood, that is, has a short time base.

There have been many reservoir operation models reported in the literature but only a few have been directed at reservoir operations under flooding conditions. Jamieson and Wilkinson (1972) developed a DP model for flood control with forecasted inflows being the inputs to the model. Windsor (1973) employed a recursive linear programming procedure for the operation of flood control systems, using the Muskingum method for channel routing and the mass balance equation for reservoir computations.

The U.S. Army Corps of Engineers (1973, 1979) developed HEC-5 and HEC-5C for reservoir operation for flood control, where releases are selected by applying a fixed set of heuristic rules and priorities that are patterned after typical operation studies. These models are based upon hydrologic routing techniques and provide no optimal stragety for operation. One application of these models was to the Kanawha River Basin (U.S. Army Corps of Engineers, 1983) which contributes which contributes flow to the Ohio River at Pt. Pleasant, West Virginia. Figure5.1.1 illustrates observed and forecasted hydrographs at Kanawha Falls for the March 1967 event. The vertical dashed line represents the time of the forecast.

The Tennessee Valley Authority (1974) devoloped an incremental dynamic programming and successive approximations technique for real-time operations with flood control and hydropower generation being the objectives. Can and Houck (1984) developed a goal programming model for the hourly operations of a multireservoir system and applied it to the Green River basin in Indiana. The model objective is defined by a hierarchy of goals, with the best policy being a predetermined rule curve.

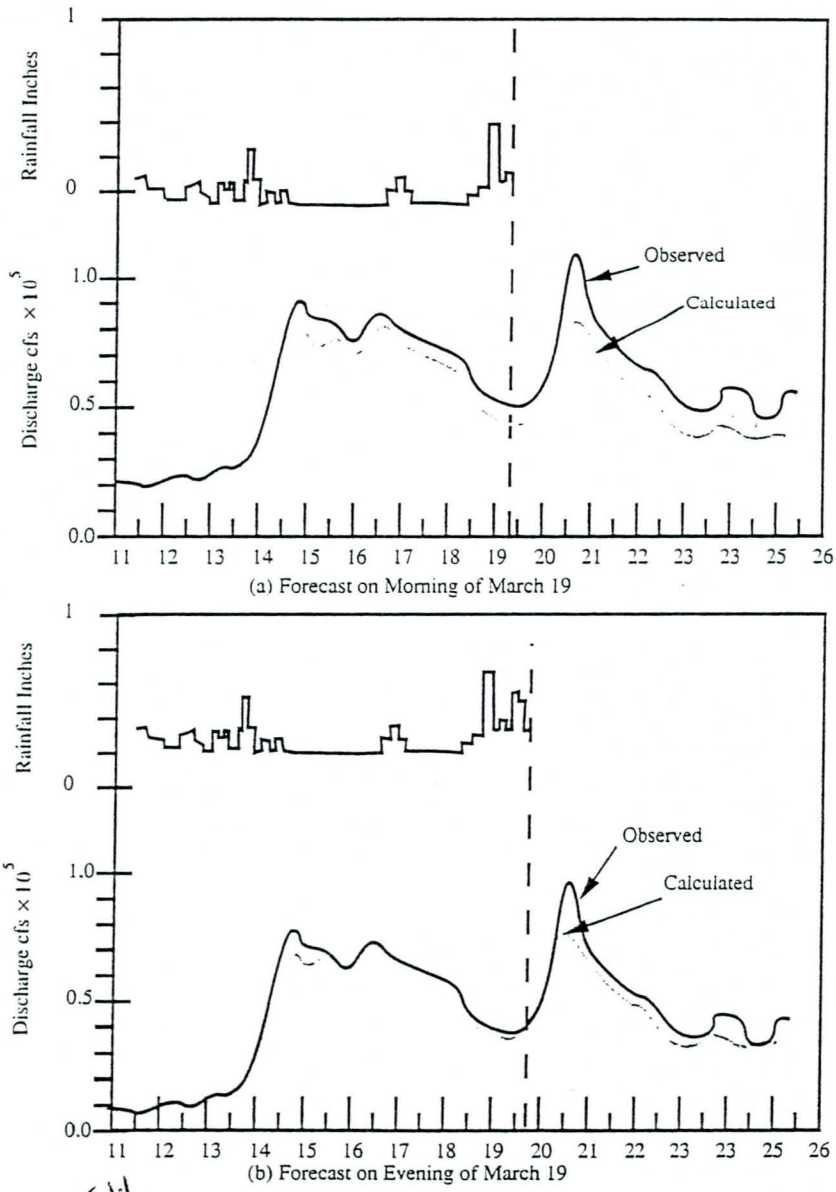


FIGURE ~~13.2.1~~ 5.1.1

Observed and forecasted hydrographs at Kanawha Falls, resulting from a forecast of the March 1967 flood event. (U.S. Army Corps of Engineers, 1983)

Wasimi and Kitanidis (1983) developed an optimization model for the daily operations of a multireservoir system during floods which combines linear quadratic gaussian optimization and a state-space mathematical model for flow forecasting. Yazicigil (1980) developed an LP optimization model for the daily real-time operations of the Green River basin in Indiana, a system of four multipurpose reservoirs. The model inputs are deterministic. The objective of operation is to follow a set of target states, deviations from which are penalized. The channel routing is performed using a linear routing procedure similar to the Muskingum method, called multi-input linear routing. The reservoir calculations are based on mass-balance equations which take into account precipitation input.

The flood forecasting model for the Lower Colorado River-Highland Lakes system in Texas developed by Unver et al. (1987) was developed for a real-time framework to make decisions on reservoir operations during flooding. This model is an integrated computer program with components for flood routing, rainfall-runoff modeling, and graphical display, and is controlled by interactive software. Input to the model includes automated real-time precipitation and stream flow data from various locations in the watershed.

The real-time reservoir operation problem involves the operation of a reservoir system by making decisions on reservoir releases as information becomes available, with relatively short time intervals which may vary between several time intervals and several hours. A new methodology is presented for operating the reservoir system under flooding conditions that incorporates: (a) a simulation model that adequately simulates the hydraulics of the system for a given flood hydrograph and a set of operating decisions, and (b) a systematic way that will improve the trial decisions made previously

and generate a set of operating decisions that would cause the least damage to the protected areas.

5.2 Problem Formulation

The optimization problem for the operation of multireservoir systems under flooding conditions can be stated as

(1) Objective:

$$\text{Minimize } z = f(\mathbf{h}, \mathbf{Q}). \tag{5.2.1}$$

(2) Constraints:

(a) Hydraulic constraints defined by the Saint-Venant equations for one-dimensional gradually varied unsteady flow and other relationships such as upstream, downstream, and internal boundary conditions and initial conditions that describe the flow in the different components of the river-reservoir system,

$$\mathbf{G}(\mathbf{h}, \mathbf{Q}, \mathbf{r}) = \mathbf{0}. \tag{5.2.2}$$

(b) Bounds on discharges defined by minimum and maximum allowable reservoir releases and flow rates at specific locations,

$$\underline{\mathbf{Q}} \le \mathbf{Q} \le \overline{\mathbf{Q}} \tag{5.2.3}$$

(c) Bounds on elevations defined by minimum and maximum allowable water surface elevations at specified locations (including reservoir levels).

$$\underline{\mathbf{h}} \le \mathbf{h} \le \overline{\mathbf{h}} \tag{5.2.4}$$

(d) Physical and operational bounds on gate operations,

$$\mathbf{0} \le \underline{\mathbf{r}} \le \mathbf{r} \le \overline{\mathbf{r}} \le \mathbf{1} \tag{5.2.5}$$

(e) Other constraints such as operating rules, target storages, storage capacities, etc.

$$\mathbf{W}(\mathbf{r}) \leq \mathbf{0}. \tag{5.2.6}$$

The objective z is defined by minimizing the total flood damage or deviations from the target levels or water surface elevations in flood areas or spills from reservoirs or maximizing storage in reservoirs. The variables **h** and **Q** are, respectively, the water surface elevations and the discharge at the computational points and **r** is the gate setting, all given in matrix form to consider the time and space dimensions of the problem. Bars above and balow a variable denote the upper and lower bounds for that variable, respectively.

5.2.1 Simulator Equations

The governing equations for one-dimensional unsteady flow are the Saint-Venant equations defined in conservation form are as follows:

Continuity:

$$\frac{\partial Q}{\partial x} + \frac{\partial (A + A_o)}{\partial t} - q = 0 \tag{5.2.7}$$

Momentum:

$$\frac{\partial Q}{\partial t} + \frac{\partial (\beta Q^2 / A)}{\partial x} + gA\left(\frac{\partial h}{\partial x} + S_f + S_e\right) - \beta q v_x + W_f B = 0 \tag{5.2.8}$$

where

x = longitudinal distance along the channel or river

t = time

A = cross-sectional area of flow

A_O= cross-sectional area of off-channel dead storage (contributes to continuity, but not momentum)

q = lateral inflow per unit length along the channel

h = water surface elevation

v_x= velocity of lateral flow in the direction of clannel flow

S_f= friction slope

S_e= eddy loss slope

B = width of the channel at the water surface

W_f=wind shear force

β = momentum correction factor

g = acceleration due to gravity

Weighted four-point finite difference approximations are used for dynamic routing with the Saint-Venant equations. The spatial derivatives $\partial Q / \partial x$ and $\partial h / \partial x$ are estimated between adjacent time lines,

$$\frac{\partial Q}{\partial x} = \theta \frac{Q_{i+1}^{j+1} - Q_i^{j+1}}{\Delta x_i} + (1-\theta)\frac{Q_{i+1}^{j} - Q_i^{j}}{\Delta x_i} \tag{5.2.9}$$

$$\frac{\partial h}{\partial x} = \theta \frac{h_{i+1}^{j+1} - h_i^{j+1}}{\Delta x_i} + (1-\theta)\frac{h_{i+1}^{j} - h_i^{j}}{\Delta x_i} \tag{5.2.10}$$

and the time derivatives are estimated using

$$\frac{\partial (A + A_o)}{\partial t} = \frac{(A + A_o)_i^{j+1} + (A + A_o)_{i+1}^{j+1} - (A + A_o)_i^{j} - (A + A_o)_{i+1}^{j}}{2\Delta t_j} \tag{5.2.11}$$

$$\frac{\partial Q}{\partial t} = \frac{Q_i^{j+1} + Q_{i+1}^{j+1} - Q_i^{j} - Q_{i+1}^{j}}{2\Delta t_j} \tag{5.2.12}$$

The nonderivative terms, such as q and A are estimated between adjacent time lines using

$$\begin{aligned} q &= \theta \frac{q_i^{j+1} + q_{i+1}^{j+1}}{2} + (1-\theta)\frac{q_i^{j} + q_{i+1}^{j}}{2} \\ &= \theta \bar{q}_i^{j+1} + (1-\theta)\bar{q}_i^{j} \end{aligned} \tag{5.2.13}$$

$$A = \theta\frac{A_i^{j+1} + A_{i+1}^{j+1}}{2} + (1-\theta)\frac{A_i^j + A_{i+1}^j}{2}$$
$$= \theta\overline{A}_i^{j+1} + (1-\theta)\overline{A}_i^j \tag{5.2.14}$$

where $\overline{q}_i$ and $\overline{A}_i$ indicate the lateral flow and cross-sectional area averaged over the reach Δx_i.

The finite-difference form of the continuity equation is produced by substituting Eqs. (5.2.9), (5.2.11), and (5.2.13) into (5.2.7) and rearranging to obtain

$$\theta\left(Q_{i+1}^{j+1} - Q_i^{j+1} - \overline{q}_i^{j+1}\Delta x_i\right) + (1-\theta)\left(Q_{i+1}^j - Q_i^j - \overline{q}_i^j\Delta x_i\right)$$
$$+ \frac{\Delta x_i}{2\Delta t_j}\left[(A + A_o)_i^{j+1} + (A + A_o)_{i+1}^{j+1} - (A + A_o)_i^j - (A + A_o)_{i+1}^j\right] = 0 \tag{5.2.15}$$

Similarly, the momentum equation in finite difference form is

$$\frac{\Delta x_i}{2\Delta t_j}\left(Q_i^{j+1} + Q_{i+1}^{j+1} - Q_i^j - Q_{i+1}^j\right)$$
$$+ \theta\left\{\left(\frac{\beta Q^2}{A}\right)_{i+1}^{j+1} - \left(\frac{\beta Q^2}{A}\right)_i^{j+1} + g\overline{A}_i^{j+1}\left[h_{i+1}^{j+1} - h_i^{j+1} + \left(\overline{S}_f\right)_i^{j+1}\Delta x_i + \left(\overline{S}_e\right)_i^{j+1}\Delta x_i\right]\right.$$
$$\left. - \left(\overline{\beta q v_x}\right)_i^{j+1}\Delta x_i + \left(\overline{W}_f\overline{B}\right)_i^{j+1}\Delta x_i\right\}$$

$$+ (1-\theta)\left\{\left(\frac{\beta Q^2}{A}\right)_{i+1}^{j} - \left(\frac{\beta Q^2}{A}\right)_i^{j} + g\overline{A}_i^{j}\left[h_{i+1}^{j} - h_i^{j} + \left(\overline{S}_f\right)_i^{j}\Delta x_i + \left(\overline{S}_e\right)_i^{j}\Delta x_i\right]\right.$$
$$\left. - \left(\overline{\beta q v_x}\right)_i^{j}\Delta x_i + \left(\overline{W}_f\overline{B}\right)_i^{j}\Delta x_i\right\} = 0 \tag{5.2.16}$$

where the average values (marked with $^-$) over a reach are defined as

$$\overline{\beta}_i = \frac{\beta_i + \beta_{i+1}}{2} \tag{5.2.17}$$

$$\overline{A}_i = \frac{A_i + A_{i+1}}{2} \tag{5.2.18}$$

$$\overline{B}_i = \frac{B_i + B_{i+1}}{2} \tag{5.2.19}$$

$$\overline{Q}_i = \frac{Q_i + Q_{i+1}}{2} \tag{5.2.20}$$

Also,

$$\overline{R}_i = \frac{\overline{A}_i}{\overline{B}_i} \tag{5.2.21}$$

for use in Manning's equation. Manning's equation may be solved for S_f and written in the form shown below, where the term $|Q|Q$ has magnitude Q^2 and sign positive or negative depending on whether the flow is downstream or upstream, respectively:

$$(\overline{S}_f)_i = \frac{\overline{n}_i^2 |\overline{Q}_i| \overline{Q}_i}{2.208 \overline{A}_i^2 \overline{R}_i^{4/3}} \tag{5.2.22}$$

The minor head losses arising from contraction and expansion of the channel are proportional to the difference between the squares of the downstream and upstream velocities, with a contraction/expansion loss coefficient K_e:

$$(\overline{S}_e)_i = \frac{(K_e)_i}{2g\Delta x_i}\left[\left(\frac{Q}{A}\right)^2_{i+1} - \left(\frac{Q}{A}\right)^2_i\right] \tag{5.2.23}$$

The velocity of the wind relative to the water surface, V_r, is defined by

$$(\overline{V}_r)_i = \left(\frac{Q_i}{A_i}\right) - (\overline{V}_w)_i \cos\omega \tag{5.2.24}$$

where ω is the angle between the wind and the water directions. The wind shear factor is then given by

$$(\overline{W}_f)_i = (C_w)_i |(\overline{V}_r)_i| (\overline{V}_r)_i \tag{5.2.25}$$

where C_w is the friction drag coeffecient.

The terms having superscript j in Eqs. (5.2.15) and (5.2.16) are known either from initial conditions, or from a solution of the Saint-Venant equations for a previous time line. The terms g, Δx_i, β_i, K_e, C_w, and V_w are known and must be specified independently of the solution. The unknown terms are Q_i^{j+1}, Q_{i+1}^{j+1}, h_i^{j+1}, h_{i+1}^{j+1}, A_i^{j+1}, A_{i+1}^{j+1}, B_i^{j+1}, and B_{i+1}^{j+1}. However all the terms can be expressed as functions of the unknowns, Q_i^{j+1}, Q_{i+1}^{j+1}, h_i^{j+1}, and h_{i+1}^{j+1}, so there are actually four unknowns. The unknowns are raised to powers other than unity, so (5.2.15) and (5.2.16) are nonlinear equations.

The continuity and momentum equations are considered at each of the N-1 rectangular grids shown in figure 5.2.1, between the upstream boundary at i = 1 and the downstream boundary at i = N. This yields 2N - 2 equations. There are two unknowns at each of the N grid points (Q and h), so there are 2N unknowns in all. The two additional equations required to complete the solution are supplied by the upstream and downstream boundary conditions. The upstream boundary condition is usually specified as a known inflow hydrograph, while the downstream boundary condition can be specified as a known stage hydrograph, a known discharge hydrograph, or a known relationship between stage and discharge, such as a rating curve.

5.2.2 Constraints

The constraints of the model can be divided into two groups: the hydraulic constraints and the operational constraints. The hydraulic constraints are equality constraints consisting of the equations that describe the flow in the system. These are (a) the Saint-Venant equations for all the computational reaches except internal boundary reaches, (b) relationship to

describe the upstream and downstream boundary conditions in addition to the Saint-Venant equations for the extremities, and (c) internal boundary conditions including the continuity equation and a flow relationship.

Internal boundary conditions describe cannot be described by the Saint-Venant equations such as critical flow resulting from flow over a spillway or waterfall. The operational constraints are basically greater-than or less-than type constraints that define variable bounds, operational targets, structural limitations, capacities etc. Options for the operator to set or limit the values of certain variables are also classified under this category. The solution methodology used in this study seperately solves the hydraulic and operational constraints. The hydraulic constraints are solved implicitly by the simulation model, DWOPER, whereas the operational constraints are solved by the optimization model, GRG2. The DWOPER model performs the unsterady flow computations.

Bound constraints are used to impose operational or optimization-related requirements. Nonnegativity constraints on discharges are not used because discharges are allowed to take on negative values in order to be able to realistically represent the reverse flow phenomena (backwater effects) due to a rising lake or due to large tributary inflows into a lake.Nonnegativity of water surface elevations is always satisfied since the system hydraulics are solved implicitly by the simulation model, DWOPER. The lower limits on elevations and discharges can be used to impose water quality considerations, minimum required reservoir releases, and other policy requirements. The upper bounds on elevations and discharges can be used to set the maximum allowable levels (values beyond are either catastrophic or physically impossible) such as the overtopping elevations for major structures, spillway capacities, etc. When the objective function, Equation (5.2.23) or (5.2.24) is

used, the damaging elevations and/or discharges must be given to the model through the constraints, as neither objective function has any terms to control them.

The third model variable, gate openings, are allowed to vary between zero and one, which corresponds to zero and one hundred percent opening of the total available gate area, respectively. The upper and lower bounds on the model variables are expressed mathematically as

$$\underline{Q}_i^j \le Q_i^j \le \overline{Q}_i^j, \quad \forall\ i, j, \tag{5.2.26}$$

$$\underline{h}_i^j \le h_i^j \le \overline{h}_i^j, \quad \forall\ i, j, \tag{5.2.27}$$

$$0 \le \underline{r}_i^j \le r_i^j \le \overline{r}_i^j \le .1, \quad \forall\ i, j: i \in l_r \tag{5.2.28}$$

where variables with a bar above them denote upper limits; those with a bar below them denote lower limits; i and j are respectively the time and location index; and l_r is the set containing the reservoir locations. Q, h, and r denote the discharge, water surface elevation, and gate opening, respectively.

The bounds on gate settings are intended primarily to reflect the physical limitations on gate operations as well as to enable the operator to prescribe any portion(s) of the operation for any reservoir(s). Operational constraints other than bounds can be imposed for various purposes.The maximum allowable rates of change of gate openings, for instance, for a given reservoir, can be specified through this formulation, as a time-dependent constraint. This particular formulation may be very useful, especially for cases where sharp changes in gate operations, i.e. sudden openings and closures, are not desirable or physically impossible. It is handled by setting an upper bound to the change in the percentage of gate opening from one time step to the next. This constraint can also be used to model another important aspect

to gate operationsd for very short time intervals, i.e.the gradual settings that have to be followed when opening or closing a gate. For this case, the gate cannot be opened (or closed) by more than a certain percentage during a given time interval. This can be expressed in mathematical terms as follows:

$$-r_{ci}^{j} \le r_{i}^{j+1} - r_{i}^{j} \le r_{oi}, \quad i \in I_r \tag{5.2.29}$$

where r_c and r_o are the maximum allowable (or possible) percentages by which to open and close the gate. This constraint can be used to model manually operated gates, for example, for all or a portion of the time intervals. The same constraint can be used, for example, to incorporate an operational rule that ties the operations of a reservoir to those of the upstream reservoir such as a multi-site constraint.

5.2.3 Objective Functions

The model can be based upon any of a number of objective functions reflecting various approaches to real-time reservoir operation for flood control. The first objective function is based on minimizing total flood damages which are defined as a function of water surface elevations in flood-prone areas.A damage-elevation relationship is provided to the modelfor each location where flood damage potential exists. The overall damage to be minimized is the summation of the total damages at each location. The mathematical expression for this objective function is:

$$\min\ z = \sum_{i}\sum_{j} c_i h_i^j, \quad i \in I_c, j \in T, \tag{5.2.30}$$

where z is the objective function value; i is the location index; I_c is the set that contains flood control locations; j is the time index; T is the time horizon; c is the unit flood damage defined as a function of the water surface elevation;

h_i^j. The unit flood damage, c, is expressed in terms of the water surrface elevation at flood control locations. It must be noted that, unlike the more common approach to damage functions (e.g. Windsor, 1973), the damage is not a function of the maximum water surface elevation for any given location, but rather a function of all elevations that are individually damaging. This approach was chosen to keep all water surface elevations in the nondamaging range individually and when this is not possible to minimize the number of times a damaging elevation occurs. The total damage cost, however, may not have a real meaning in dollar value due to the nature of this formulation.

The second objective function is basically the same as the first one except that flood damages are expressed in terms of discharges instead of water surface elevations, given as:

$$\min\ z = \sum_i \sum_j c'_i Q_i^j, \quad i \in I_p, j \in T, \tag{5.2.31}$$

where c' is the unit flood damage as a function of discharge, Q_j. The unit flood damages, c', are expressed in terms of the discharge at the flood control locations. This objective function is provided for cases where it is more convenient to express damages in terms of flow rates for certain locations, or the available data is in this form. It must be noted that this objective function would normally be used for natural channels as the damages in lakes are almost always a function of flood stages.

The third objective function is a combination of the first two for cases where both discharges and water surface elevations are used to define the flood damages given as

$$\min z = \sum_{i \in I_c} \sum_j c_i h_i^j + \sum_{i \in I_p} \sum_j c'_i Q_i^j, j \in T, \tag{5.2.32}$$

where l_c is the set that contains locations where damage is a function of water surface elevation and l_c is the set that contains locations where damage is a function of discharge. The myopic nature of short-term operation is usually handled by constraints that represents the end-of-the-period, or medium-term targets or goals. For example, the possiblity of ending up with an empty reservoir is usually prevented by defining a lower limit for the water surface elevation of the headwater location for time step T. An alternative to this is given by the fourth objective function. The objective of operation is defined as the maximization of the total reservoir storages while keeping the water stages and/or flow rates within nondamaging ranges through the constraint set. The fourth objective function is

$$\max z = \sum_i \sum_j Q_i^j, j \in T, \tag{5.2.33}$$

where all terms are as defined earlier.

Zoning is another very common approach used in modeling the real-time operation objectives (e.g. Yazicigil, 1982; Can and Houck, 1984;Wasimi and Kitanidis, 1983). In order to use this approach, operation targets (or ideal levels) are defined prior to operation and deviations from these are penalized through a penalty function. Zones are identified for different levels deviations and a unit penalty (or a penalty coefficient) is assigned to each, almost always in such a way that the resulting function is convex. Although the solution methodology presented in the next section has provisions for violated bounds on discharges and water surface elevations, a penalty-type objective function is presented here as the sixthe

objective function, for cases where data are already available or the reservoir operator opts to use a penalty function. The mathematical expression for the sixth objective is:

$$\min\ z = \sum_i \sum_j c_i h_i^j + \sum_i \sum_j c'_i Q_i^j \ , i \in l_s, j \in T, \tag{5.2.34}$$

where l_s is the set that contains locations for which a target is specified and c and c' are the unit penalties associated with water surface elevation, h, and discharge, Q. It must be noted that water surface elevations in this formulation replace the deviations used in most penalty functions. However, this is justified by the fact that the inclution of the target into the objective function contributes a constant to the objective value, which does not affect the optimization, within the given range of unit penalties. Different unit penalties for different locations are used to reflect the relative importance of each location.

5.3 Problem Solution

5.3.1 Overview

The optimization problem stated above is a large mathematical programming problem for most real-world situations. In modeling a river system, computational points are used to discretize the river channels and reservoirs. Each computational point, for each time-step of the operation, contributes two flow variables (water surface elevation and discharge) and two hydraulic constraints (the Saint-Venant equationsa or other flow relations) to the problem. In addition, each reservoir contributes another variable(the setting of the equivalent gate) per time step. The external boundaries each contribute an additional hydraulic relationship. Thus a

typical 24 h operation horizon with 1 h time steps for a river system with five reservoirs and 150 computational pointswould give rise to a problem with more than 7200 flow equations (two times the product of the number of time steps and computational nodes) and over 7200 flow variables. This is beyond the capacity of existing nonlinear programming codes. A logical approach in solving a problem this large would be to reduce its size. Traditionally, the problem size has been reduced by replacing the unsteady flow equations by more simplistic relationships. In this work, a different approach is taken to alleviate the dimensionality problem. The optimum control model presented here leads to an efficient algorithm to solve the optimization problem without sacrificing the hydraulic model accuracy.

The basic idea is to solve the hydraulic constraints (Saint-Venant equations) using an unsteady flow routing model such as the U.S. National Weather Service Dynamic Wave Operational (DWOPER) model. For each iteration of the optimization model,, the simulator (DWOPER) solves for the water surface elevations, h, and the flow rates, Q, given the gate operations which are the control variables. This allows the constraints and the objective function of the reservoir optimization problem to be viewed as a function of only the controllable variables. Since there are relatively few controllable variables, the resulting reduced problem is easier to solve. The major remaining difficulty is to compute the first partial derivatives of the objective and constraint functions with respect to the controllable variables. Once the derivatives are determined, several efficient nonlinear optimization routines could be used to solve the reduced optimization problem.

5.3.2 The Reduced Problem

The operations problem (Equations (5.2.1) - (5.2.6)), referred to as the general operations model (GOM) has certain characterstics that can be used in reducing it to a smaller problem. The GOM has the general structureof a discrete time control with three basic groups of constraints: those concerning the state of the system (hydraulic constraints) and those describing the system controls (bound and operational constraints). The GOM yields to an efficient solution algorithm when the state variables (discharges and water surface elevations) and the control variables (gate settings) are treated seperately, in a coordinated manner. The hydraulic constraints (Equation (5.2.2)) can be solved sequentially forward in time for water surface elevations, **h** and the flow rates, **Q** by using the DWOPER simulation model, once the gate settings, **r** are specified. The general optimal control approach to the real-time reservoir operation problem is shown in Figure 5.3.1. Through this simulator-optimizer formulation, the problem is solved efficiently by incorporating the simulation model into a procedure when a set of gate operations, **r**, (control vector) is chosen, the simulation model is run subject to the selected control vector, to solve the hydraulic constraint set, **g**, for the elevations and discharges (state vector). Then the objective function is evaluated, the bound constraints are checked for any violations and the procedure is repeated with an updated set of gate operations until a convergance criterion is satisfied and no bound constraints are violated.

It must be noted that the optimization is performed only on the gate settings in this procedure. The new optimization problem, called the reduced operations model (ROM) has N_r*T variables compared to the (2N*T + N_r*T) variables of the GOM, where N, T, and N_r are the total number of computational points, time steps, and reservoirs, respectively. The number of constraint equations has also been reduced by the same amount, (2N*T),

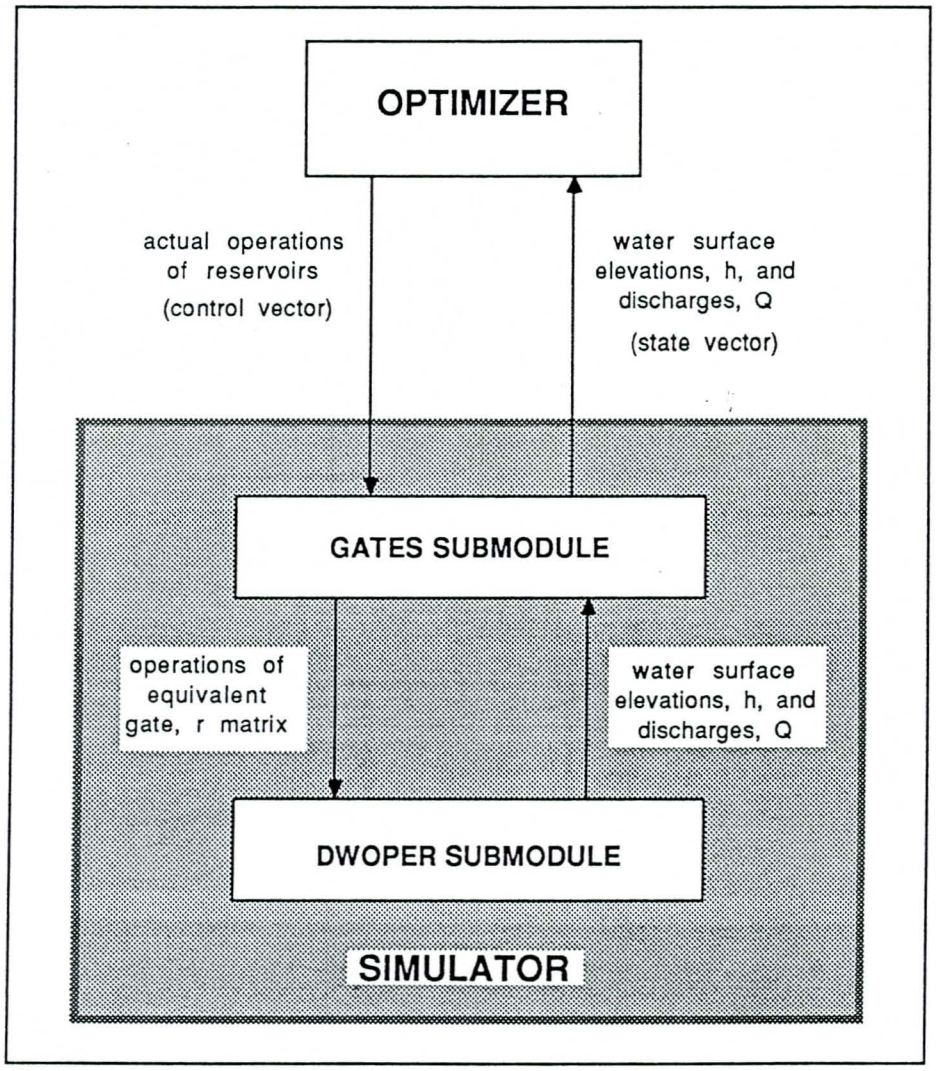


Figure 5.3.1. Optimal Control Approach to Operations Problem

with the elimination of the hydraulic constraints, **g**. The transformation of the operations problem is shown in Figure 5.3.2, along with the problem size at each step of the transformation for an example system. The problem size for an example with 100 computational points, 5 reservoirs, and 48 time steps is drastically reduced, from over 9000 variables and constraints to 120 variables and 120 bound constraints because of the simulator-optimizer formulation.

The hydraulic constraint set, **g**, has a special staircase banded structure that can be exploited to construct an efficient overall algorithm. The model presented herein combines the simulation model, DWOPER, and the optimization model, GRG2, within the framework of an optimum control formulation. The transformation of the original problem into the reduced one is similar to the generalized reduced gradient approach, which is also used to solve the reduced (transformed) problem.

The original problem, GOM, can be converted into a reduced problem as suggested by the implicit function theorem (Luenberger, 1973). The implicit function theorem states that if some of the problem variables can be solved in terms of the remaining variables, then a reduced problem can be devised which can be mainipulated more easily. The approach is applied to the problem given by Equations 5.2.1 – 5.2.6 in such a way that the hydraulic constraints (Equation 5.2.2) are handled separately by the simulator and the other constriants by the optimizer. The simulation model computes the values of the state variables, **h** and **Q** for given values of the control variables **r** and the optimization model seeks the optimal values of **r** that will minimize the objective function. The implicit function theorem states that **h(r)** and **Q(r)** exist if and only if the basic matrix (the Jacobian of the system of equations given by (Equation 5.2.2)) is nonsingular. This condition is always

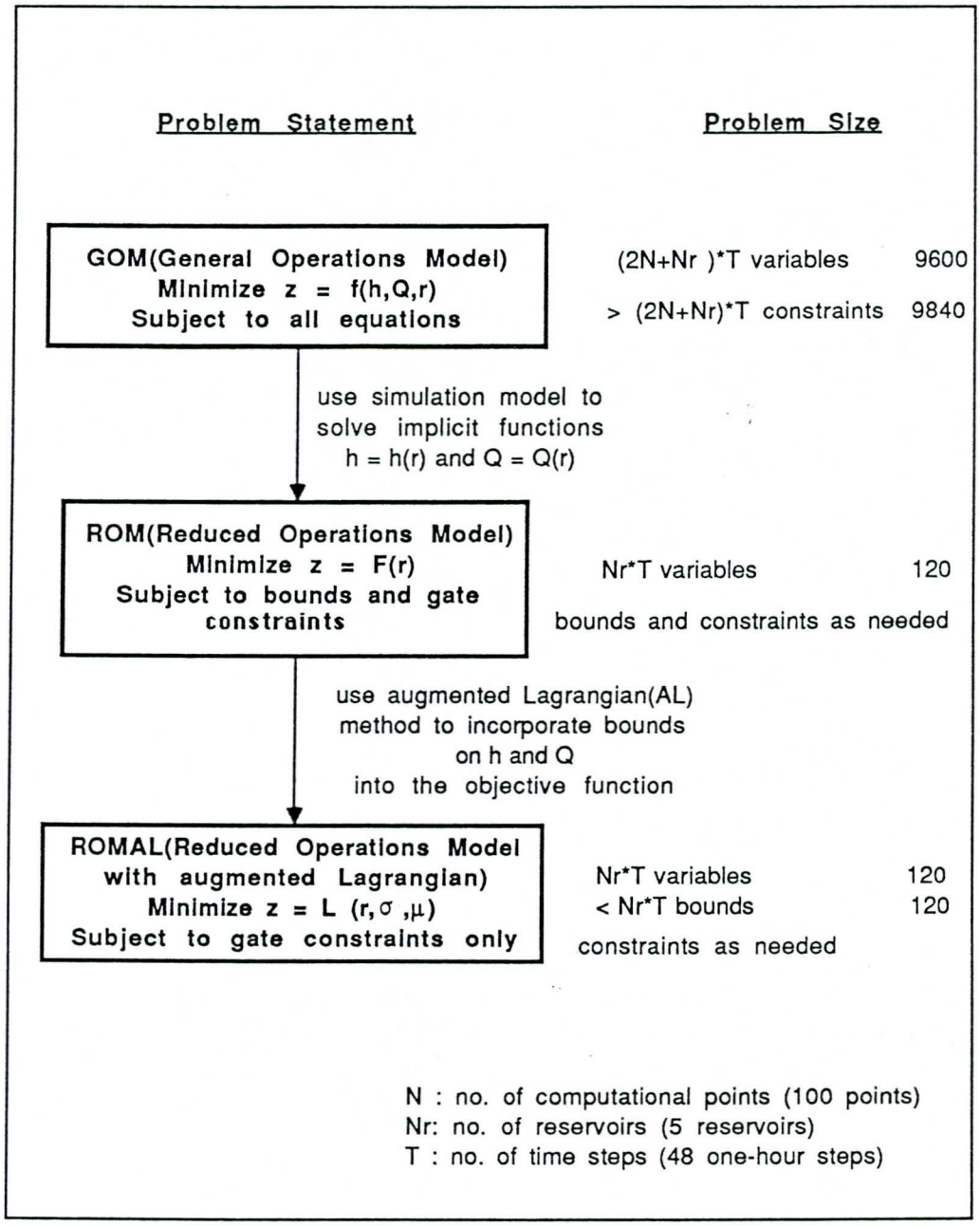


Figure ~~2~~ 5.3.2. Transformation of Operations Problem

satisfied when a solution is possible, as the simulator (DWOPER), uses the same matrix for the finite-difference unsteady flow computations.

Expressing the water surface elevation and discharge as a function of the control variable, **r**,

$$\mathbf{h} = \mathbf{h}(\mathbf{r}) \tag{5.3.1}$$

and

$$\mathbf{Q} = \mathbf{Q}(\mathbf{r}), \tag{5.3.2}$$

then, the objective function, now called the reduced objective function is expressed as

$$\text{Minimize } z = \mathbf{F}(\mathbf{r}) = f\,[\,\mathbf{h}(\mathbf{r}), \mathbf{Q}(\mathbf{r})\,]. \tag{5.3.3}$$

The objective function can be evaluated once the state variables, **h** and **Q** , are computed for the given set of control variables, **r**.

The reduced problem, which is called the reduced operations model (ROM), is now expressed by the reduced objective function, Equation (~~19~~ 5.3.3), subject to Equation (~~3)-(6~~ 5.2.3-5.2.6). The ROM is much smaller in size than the GOM with the simulator determining the implicit functions **h(r)** and **Q(r)**, by performing the unsteady flow computations thus eliminating the constraint matrix **g** that describes the hydraulics.

In solving the ROM by a nonlinear programming algorithm, the Jacobian of the matrix **g(h, Q, r)** will be required as well as the gradients of the functions F(**r**), **h(r)**, and **Q(r)**, which are also called the reduced gradients. The Jacobian matrix is defined as

$$\mathbf{J}\,(\mathbf{h}, \mathbf{Q}, \mathbf{r}) = [\,\partial \mathbf{g}\,/\partial \mathbf{h}, \partial \mathbf{g}/\partial \mathbf{Q}, \partial \mathbf{g}/\partial \mathbf{r}\,] = [\mathbf{B}, \mathbf{C}] \tag{5.3.4}$$

or

$$\mathbf{J}\,(\mathbf{y}, \mathbf{r}) = [\,\partial \mathbf{g}\,/\partial \mathbf{y}, \partial \mathbf{g}/\partial \mathbf{r}\,] = [\mathbf{B}, \mathbf{C}]\,, \tag{5.3.5}$$

where **y** denotes the state variable (**h**, **Q**) and **B** is the basic matrix. The basis matrix of the optimal control problem is the same as the Jacobian matrix used

in the Newton-Raphson solution procedure in the simulation model (DWOPER). Thus, the two elements of the Jacobian matrix **J** are available (with the basis **B** explicitly computed, and terms in **C** already available) after a simulation run. The basis matrix is a banded sparse matrix with at most four nonzero elements in each row around the matrix's main diagonal.

The reduced gradients can be calculated by applying the two-step scheme used by Lasdon and Mantell (1978) and also by Wanakule et al. (1986). Letting $B_t = \partial g_t / \partial y_t$ denote the basis matrix for time step t, the following scheme is adapted for the ROM:

(i) Solve the system of finite difference equations for the last time step T to find the values of the Lagrange multipliers π_T

$$\pi_T B_T = \partial f / \partial y_T, \tag{5.3.6}$$

then solve for the π_T backward in time

$$\pi_t B_t = \partial f / \partial y_t - \pi_{t+1}(\partial g_{t+1} / \partial y_t), \text{ for } t = T-1, T-2, \ldots, 2, 1 \tag{5.3.7}$$

(ii) Calculate the value of the reduced gradient

$$\partial F / \partial r_t = \partial f / \partial r_t - \pi_t(\partial g_t / \partial r_t), \text{ for } t = 1, 2, \ldots, T \tag{5.3.8}$$

The Lagrange multipliers, π_t can be used in a sensitivity analysis as they show the effect of a small change in the corresponding term in the objective value.

5.3.3 Solution of Reduced Problem

The reduced problem, ROM, can be solved by a nonlinear programming algorithm. As the reduced problem still contains bound-type constraints on the state variables **h** and **Q** , the algorithm adopted should have provisions to assure the feasibility of the simulation model solutions for the state variables. An augmented Lagrangian (AL) algorithm that incorporates the bounds on the state variablesinto the objective function is used for this purpose. An application of this type can be found in Hsin (1981)

where the bounds on the state variables are violated until the solution converges. The reduced problem with AL terms is

$$\min L_A(\mathbf{r}, \mu, \sigma) = F(\mathbf{r}) + 0.5\sum_i \sigma_i \min\left[0, (b_i - \mu_i/\sigma_i)\right]^2 + 0.5\sum_i \mu_i^2/\sigma_i \quad (5.3.9)$$

where i denotes the constraint set which is formed of the bounds on the state variables, i.e. the water surface elevations and discharges, and σ_i and μ_i are, respectively, the penalty weight and the Lagrange multiplier associated with the ith bound. The term b_i is the violation term defined as

$$b_i = \min\left[(y_i - \underline{y}_i), (\overline{y}_i - y_i)\right] \quad (5.3.10)$$

The constraints of the new problem are the bounds on the control variables and the operating constraints.

A reduced gradient approach is adopted to solve the reduced problem with AL terms. This new problem, which will be referred to as the reduced operations model with augmented Lagrangian (ROMAL) can be expressed as

$$\text{Minimize } L_A(\mathbf{r}, \sigma, \mu) \quad (5.3.11)$$

subject to Equations (5.2.5) and (5.2.6).

The solutions to this is a two-step procedure with an inner and an outer problem that must be solved. The objective function of this inner-outer problem combination is

$$\min_{\sigma, \mu}\left[\min_{\mathbf{r} \in S} L_A(\mathbf{r}, \sigma, \mu)\right] \quad (5.3.12)$$

where **r** is selected from S, the set of feesible gate settings defined by Equation (5.2.5). The inner problem involves the optimization of the augmented Lagrangian objective by using GRG2 to determine optimal values of **r** while keeping μ and σ fixed. Then outer problem is iterated by updating the values of μ and σ for the next solution run of the inner problem. The overall

optimization is attained when μ and σ need no further updating, within a given tolerance level. The updating formula used for μ is

$$\mu_i^{k+1} = \begin{cases} \mu_i^{(k)} - \sigma_i b_i, & \text{if } c_i < \mu_i / \sigma_i, \\ 0, & \text{otherwise}, \end{cases} \tag{5.3.13}$$

where k is the number of the current iteration. The value of σ is normally adjusted once during early iterations and then kept constant (Powell, 1978).

In applying the generalized reduced gradient approach to the ROMAL formulation the gradient of the new objective function is evaluated as

$$\nabla L_A(\mathbf{r}, \mu, \sigma) = \partial L_A / \partial r_i - \pi(\partial g / \partial r_i), \text{ for all } i = 1 \text{ to } 2N \tag{5.3.14}$$

The solution of the inner problem, i.e. finding the optimal **r** for fixed μ and σ is accomplished by GRG2 (Lasdon and Waren, 1983), which is based on generalized reduced gradient technique. The basic steps of the optimal control algorithm are shown in Figure ?

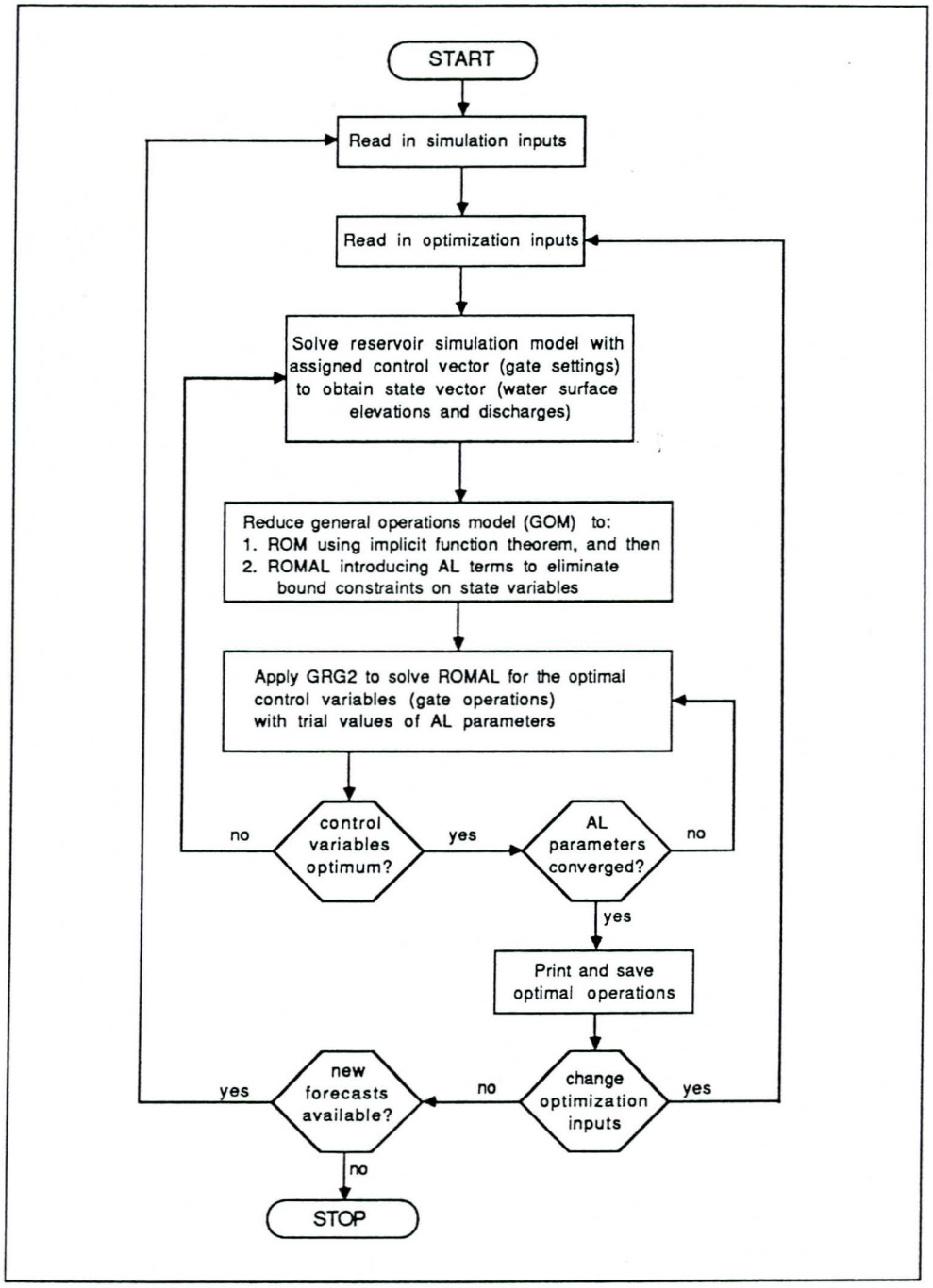


Figure 5.3.3. Basic Steps of Optimal Control Algorithm

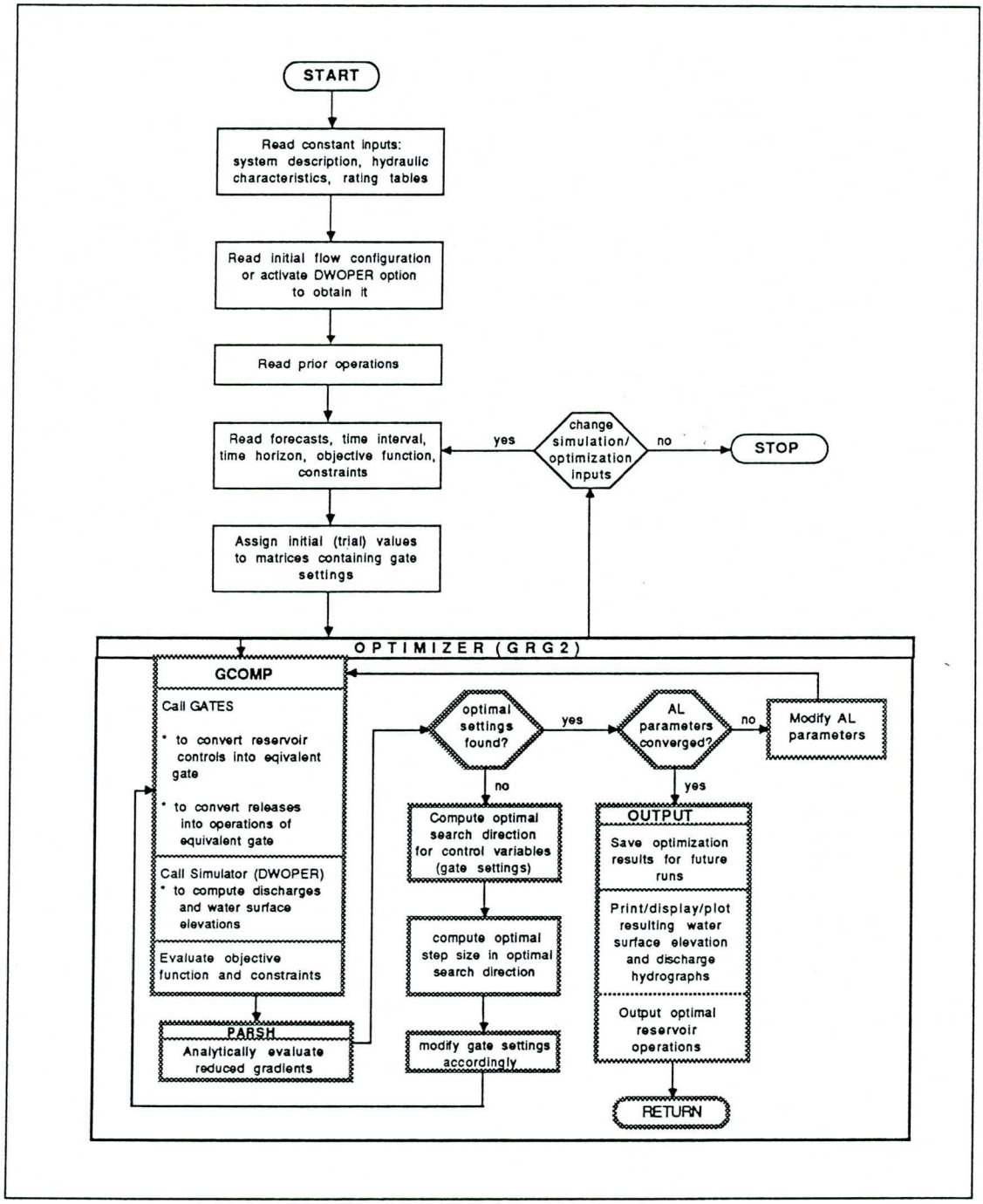


Figure ~~3~~ 5.3.4. Block Diagram of Optimal Control Algorithm

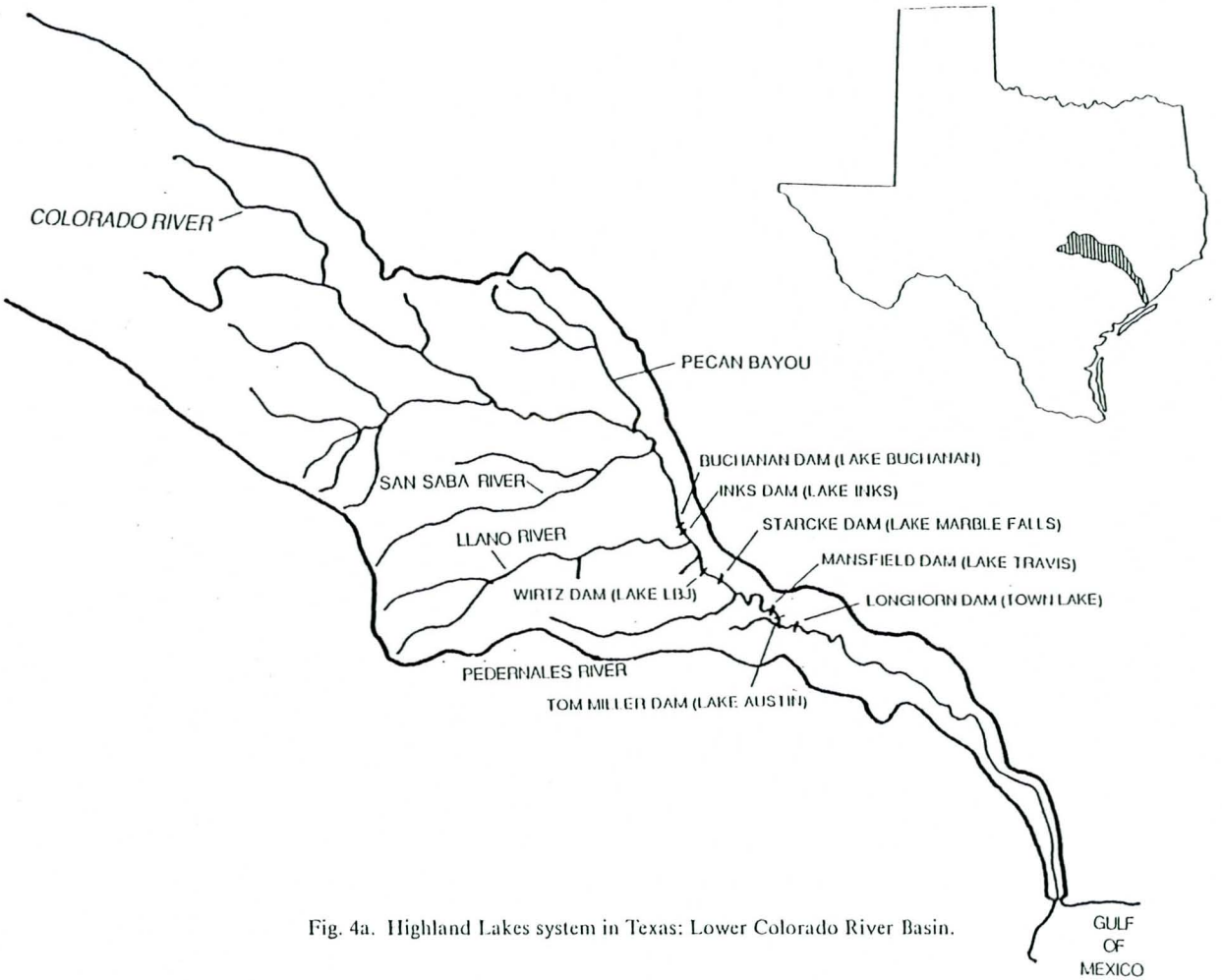


Fig. 4a. Highland Lakes system in Texas: Lower Colorado River Basin.

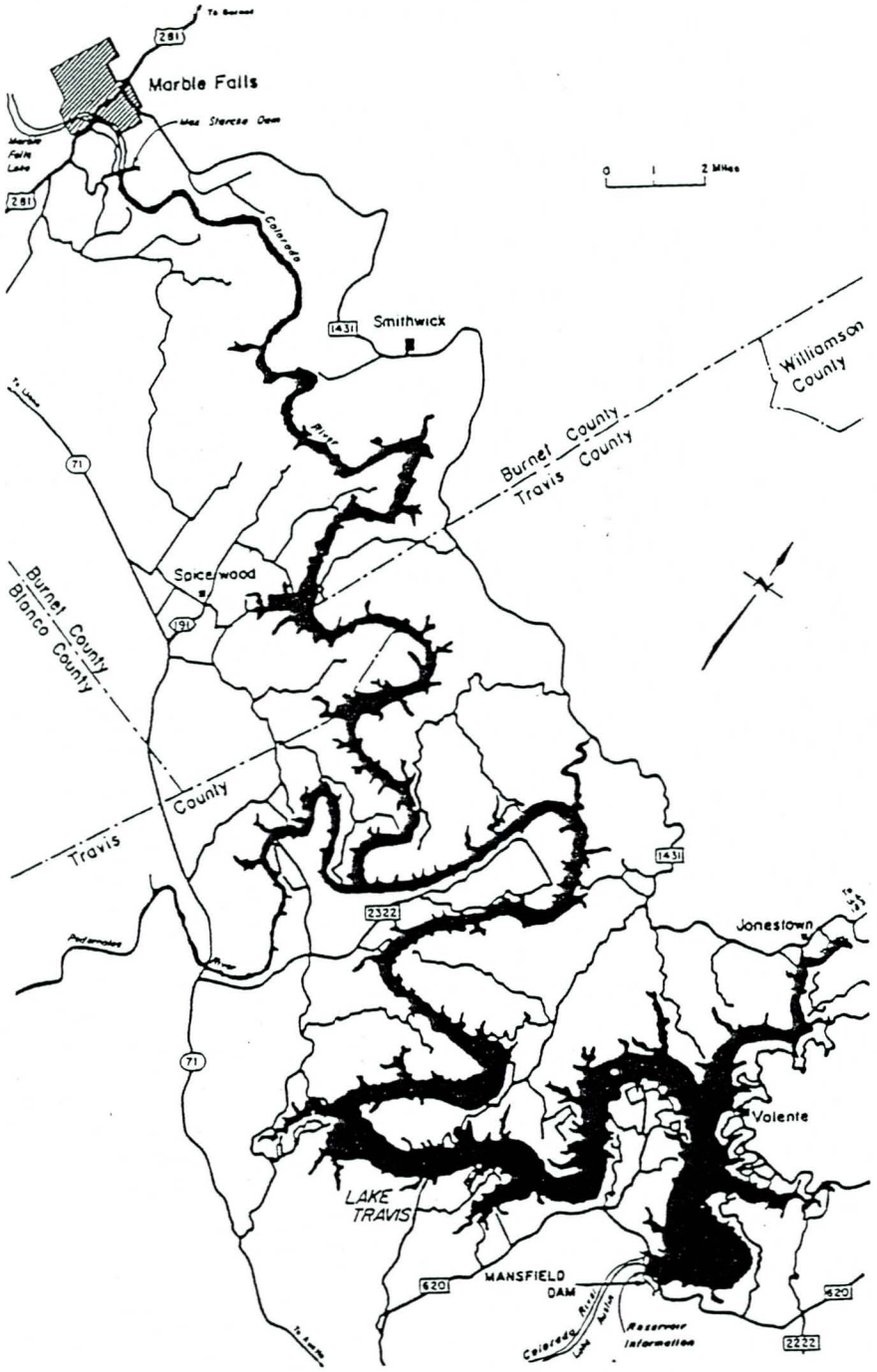


Fig. 4b. Highland Lakes system in Texas: Lake Travis (Texas Water Development Board, 1971).

APPENDIX 5A

Solution of Simulation Model

The system of nonlinear equations can be expressed in functional form in terms of the unknowns **h** and **Q** at time level j+1, as follows:

$UB(h_1, Q_1) = 0$	upstream boundary conditions	
$C_1(h_1, Q_1, h_2, Q_2) = 0$	continuity for grid I	
$M_1(h_1, Q_1, h_2, Q_2) = 0$	momentum for grid I	
.		
.		
.		
$C_i(h_i, Q_i, h_{i+1}, Q_{i+1}) = 0$	continuity for grid i	(5.A.1)
$M_i(h_i, Q_i, h_{i+1}, Q_{i+1}) = 0$	momentum for grid i	
.		
.		
.		
$C_{N-1}(h_{N-1}, Q_{N-1}, h_N, Q_N) = 0$	continuity for grid N-1	
$M_{N-1}(h_{N-1}, Q_{N-1}, h_N, Q_N) = 0$	momentum for grid N-1	
$DB(h_N, Q_N) = 0$	downstream boundary condition	

This system of 2N nonlinear equations in 2N unknowns is solved for each time step by the Newton-Raphson method (See Figure 5.A.1). The computational procedure for each time j+1 starts by assigning trial values to the 2N unknowns at that time. These trial values of **Q** and **h** can be the values known at time j from the initial condition (if j=1) or from calculations during the previous time step. Using the trial values in the system (~~10.4.~~5.A.1) results in 2N residuals. For the kth iteration these residuals can be expressed as

$UB(h_1^k, Q_1^k) = RUB^k$	residual for upstream boundary condition
$C_1(h_1^k, Q_1^k, h_2^k, Q_2^k) = RC_1^k$	residual for continuity at grid I

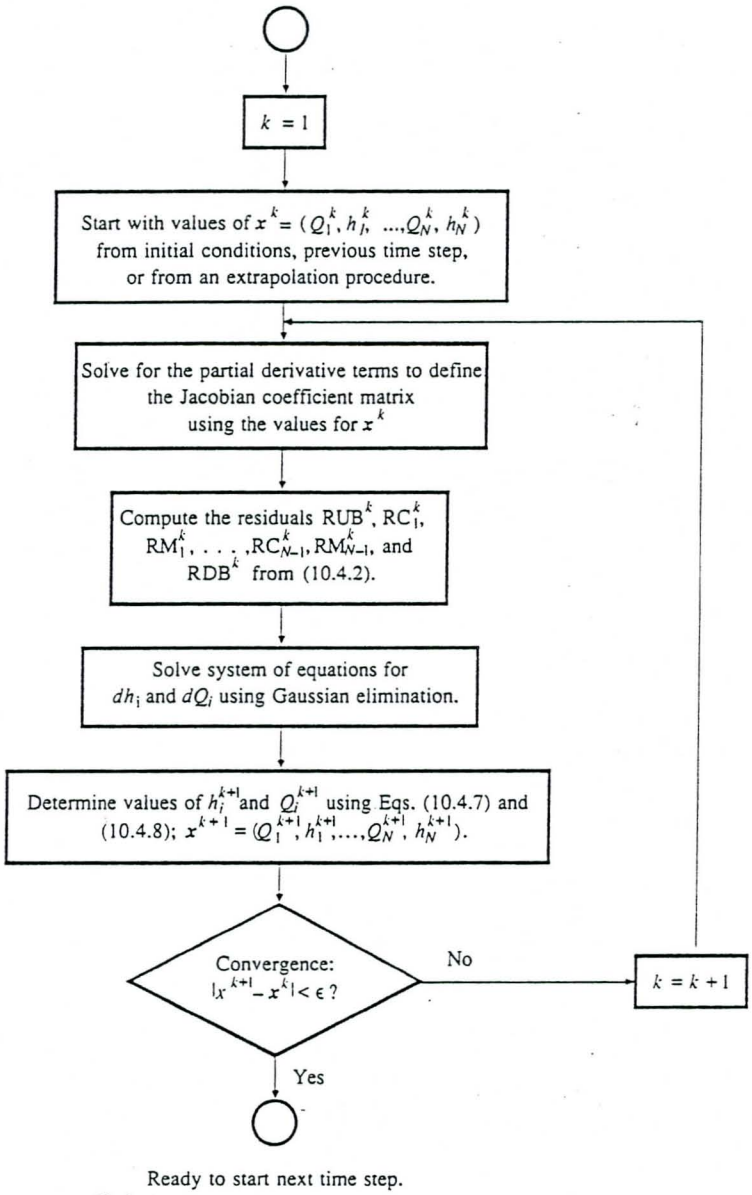


FIGURE ~~10.4.2~~ 5.17.1

Procedure for solving a system of difference equations at one time step using the Newton-Raphson method. (Chow, Maidment & Mays, 1988)

$M_1(h_1^k, Q_1^k, h_2^k, Q_2^k) = RM_1^k$ residual for momentum at grid I

.
.
.

$C_i(h_i^k, Q_i^k, h_{i+1}^k, Q_{i+1}^k) = RC_i^k$ residual for continuity at grid i (5.A.2)

$M_i(h_i^k, Q_i^k, h_{i+1}^k, Q_{i+1}^k) = RM_i^k$ residual for momentum at grid i

.
.
.

$C_{N-1}(h_{N-1}^k, Q_{N-1}^k, h_N^k, Q_N^k) = RC_{N-1}^k$ residual for continuity at grid N-1

$M_{N-1}(h_{N-1}^k, Q_{N-1}^k, h_N^k, Q_N^k) = RM_{N-1}^k$ residual for momentum at grid N-1

$DB(h_N^k, Q_N^k) = RDB_N^k$ residual for downstream boundary condition

The solution is approached by finding values of the unknowns **Q** and **h** so that the residuals are forced to zero or very close to zero.

The Newton-Raphson method is an iterative technique for solving a system of nonlinear algebraic equations. ~~It uses the same idea as was presented in Chap. 5 for the determination of flow depth in Manning's equation, except that here the solution is for a vector of variables rather than for a single variable.~~ Consider the system of equations (5.A.2) denoted in vector form as

$$f(x) = 0 \qquad (5.A.3)$$

where $x = (Q_1, h_1, Q_2, h_2, \ldots, Q_N, h_N)$ is the vector of unknown quantities and for iteration k, $x^k = (Q_1^k, h_1^k, Q_2^k, h_2^k, \ldots, Q_N^k, h_N^k)$. The nonlinear system can be linearized to

$$f(x^{k+1}) \approx f(x^k) + J(x^k)(x^{k+1} - x^k) \tag{5.A.4}$$

where $J(x^k)$ is the Jacobian, which is a coefficient matrix made up of the first partial derivatives of f(x) evaluated at x^k. The right-hand side of Eq. (5.A.4) is the linear vector function of $\bar{x}^k$. Basically, an iterative procedure is used to determine x^{k+1} that forces the residual error $f(\bar{x}^{k+1})$ in Eq. (5.A.4) to zero. This can be accomplished by setting $f(\bar{x}^{k+1}) = 0$ rearranging (5.A.4) to read

$$J(x^k)(x^{k+1} - x^k) = -f(x^k) \tag{5.A.5}$$

This system is solved for $(x^{k+1} - x^k) = \Delta x^k$, and the improved estimate of the solution, x^{k+1}, is determined knowing Δx^k. The process is repeated until $(x^{k+1} - x^k)$ is smaller than some specified tolerance.

The system of linear equations represented by (5.A.5) involves $J(x^k)$, the Jacobian of the set of equations (5.A.1) with respect to **h** and **Q** , and $-f(x^k)$, the vector of the negatives of the residuals in (5.A.2). The resulting system of equations is

$$\frac{\partial UB}{\partial h_1} dh_1 + \frac{\partial UB}{\partial Q_1} dQ_1 = -RUB^k$$

$$\frac{\partial C_1}{\partial h_1} dh_1 + \frac{\partial C_1}{\partial Q_1} dQ_1 + \frac{\partial C_1}{\partial h_2} dh_2 + \frac{\partial C_1}{\partial Q_2} dQ_2 = -RC_1^k$$

$$\frac{\partial M_1}{\partial h_1} dh_1 + \frac{\partial M_1}{\partial Q_1} dQ_1 + \frac{\partial M_1}{\partial h_2} dh_2 + \frac{\partial M_1}{\partial Q_2} dQ_2 = -RM_1^k$$

$$\vdots$$

$$\frac{\partial C_i}{\partial h_i} dh_i + \frac{\partial C_i}{\partial Q_i} dQ_i + \frac{\partial C_i}{\partial h_{i+1}} dh_{i+1} + \frac{\partial C_i}{\partial Q_{i+1}} dQ_{i+1} = -RC_i^k \tag{5.A.6}$$

$$\frac{\partial M_i}{\partial h_i} dh_i + \frac{\partial M_i}{\partial Q_i} dQ_i + \frac{\partial M_i}{\partial h_{i+1}} dh_{i+1} + \frac{\partial M_i}{\partial Q_{i+1}} dQ_{i+1} = -RM_i^k$$

$$\vdots$$

$$\frac{\partial C_{N-1}}{\partial h_{N-1}} dh_{N-1} + \frac{\partial C_{N-1}}{\partial Q_{N-1}} dQ_{N-1} + \frac{\partial C_{N-1}}{\partial h_N} dh_N + \frac{\partial C_{N-1}}{\partial Q_N} dQ_N = -RC_{N-1}^k$$

$$\frac{\partial M_{N-1}}{\partial h_{N-1}} dh_{N-1} + \frac{\partial M_{N-1}}{\partial Q_{N-1}} dQ_{N-1} + \frac{\partial M_{N-1}}{\partial h_N} dh_N + \frac{\partial M_{N-1}}{\partial Q_N} dQ_N = -RM_{N-1}^k$$

$$\frac{\partial DB}{\partial h_N} dh_N + \frac{\partial DB}{\partial Q_N} dQ_N = -RDB^k$$

APPENDIX 5B

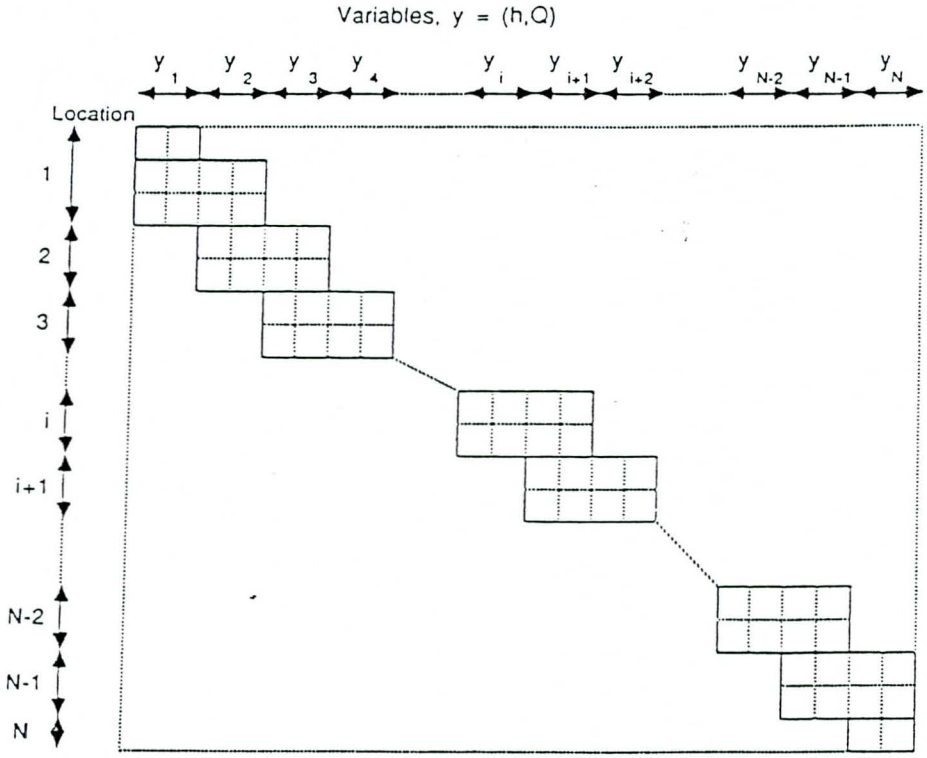
Computation of Basis Matrix Elements

For any given time step and a given set of reservoir gate settings, the system of equations has a banded structure, as shown in Figure 5.B.1, with at most four elements around the main diagonal. This structure is exploited in the simulation model by an efficient solution algorithm. The system of equations (5.A.1) constitutes the **g** matrix of the optimization model when written for all time steps. Each equation in the matrix has at most four nonzero terms that belong to the current time step, and another four nonzero terms that belong to the previous time step. The **g** matrix thus has a special banded staircase structure as shown in Figure 5.B.2, that can be exploited. The partial derivatives of the **g** matrix with respect to the problem variables have to be computed to be used in the optimization model. The mathematical expressions for the partial derivatives are given in the next two sections. Section 5.B.1 gives the partial derivatives with respect to the variables at the current time step (i.e. time step j+1), and section 5.B.2 gives the expressions with respect to the previous time step.

5.B.1 Partial Derivatives for Current Time Step

The partial derivatives for the Saint-Venant equations, external boundary conditions, and internal boundary relationships with respect to the problem variables water surface elevations **h**, discharges **Q** , and gate setting **r** at the current time step are given below under seperate headings. In the following, the subscripts denote the time step, and the subscripts denote the location of the variables.

5.B.11 Partial Derivatives for Saint-Venant Equations [Fread, 1978]



5.B.1

Figure ~~D.1.~~ Structure of g matrix for one time step

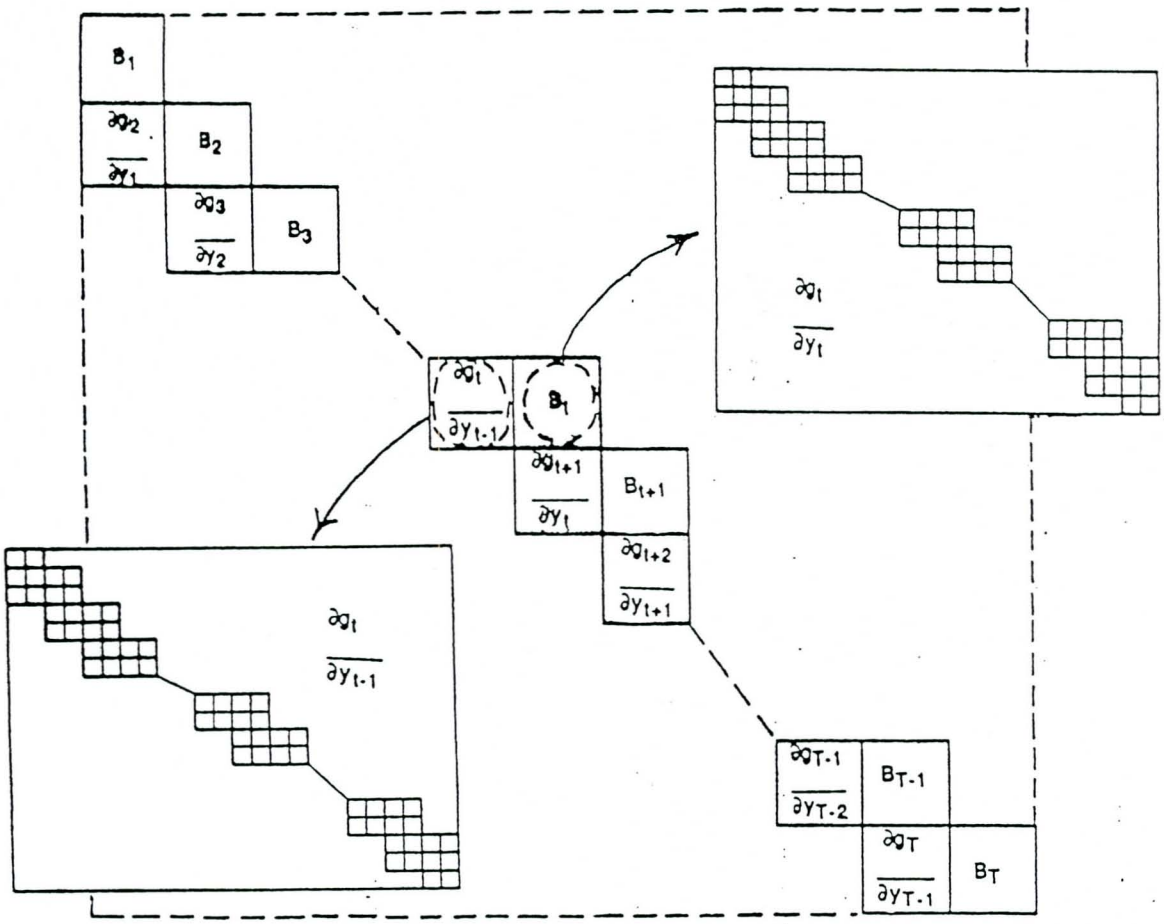


Figure ~~D.2.~~ 5.B.2 Matrix of partial derivatives with respect to y=(h,Q) for T time steps

$$\frac{\partial C_i}{\partial h_i} = \frac{\Delta x_i}{2\Delta t_j}(B + B_o)_i^{j+1} \tag{5.B.1}$$

$$\frac{\partial C_i}{\partial Q_i} = -\theta \tag{5.B.2}$$

$$\frac{\partial C_i}{\partial h_{i+1}} = \frac{\Delta x_i}{2\Delta t_j}(B + B_o)_{i+1}^{j+1} \tag{5.B.3}$$

$$\frac{\partial C_i}{\partial Q_{i+1}} = \theta \tag{5.B.4}$$

$$\frac{\partial M_i}{\partial h_i} = \theta\left[\left(\frac{\beta Q^2 B}{A^2}\right)_i^{j+1} + g\overline{A}_i^{j+1}\left(-1 + \frac{\partial \overline{S}_f}{\partial h_i}\Delta x_i + \frac{\partial \overline{S}_e}{\partial h_i}\Delta x_i\right)\right.$$
$$\left. + \frac{gB_i^{j+1}}{2}\left(h_{i+1}^{j+1} - h_i^{j+1} + \overline{S}_{f_i}^{j+1}\Delta x_i + \overline{S}_{e_i}^{j+1}\Delta x_i\right) + \frac{\overline{W}_f B}{2}_i^{j+1}\Delta x_i\right] \tag{5.B.5}$$

$$\frac{\partial M}{\partial Q_i} = \left(\frac{\Delta x_i}{2\Delta t_j}\right) + \theta\left[-2\left(\frac{\beta Q}{A}\right)_i^{j+1} + g\overline{A}_i^{j+1}\left(\frac{\partial \overline{S}_f}{\partial Q_i}\Delta x_i + \frac{\partial \overline{S}_e}{\partial Q_i}\Delta x_i\right)\right] \tag{5.B.6}$$

$$\frac{\partial M}{\partial h_{i+1}} = \theta\left[-\left(\frac{\beta Q^2 B}{A^2}\right)_{i+1}^{j+1} + g\overline{A}_i^{j+1}\left(1 + \frac{\partial \overline{S}_f}{\partial h_{i+1}}\Delta x_i + \frac{\partial \overline{S}_e}{\partial h_{i+1}}\Delta x_i\right)\right.$$
$$\left. + \frac{gB_{i+1}^{j+1}}{2}\left(h_{i+1}^{j+1} - h_i^{j+1} + \overline{S}_{f_i}^{j+1}\Delta x_i + \overline{S}_{e_i}^{j+1}\Delta x_i\right) + \frac{\overline{W}_f B}{2}_i^{j+1}\Delta x_i\right] \tag{5.B.7}$$

$$\frac{\partial M}{\partial Q_{i+1}} = \left(\frac{\Delta x_i}{2\Delta t_j}\right) + \theta\left[2\left(\frac{\beta Q}{A}\right)_{i+1}^{j+1} + g\overline{A}_i^{j+1}\left(\frac{\partial \overline{S}_f}{\partial Q_{i+1}}\Delta x_i + \frac{\partial \overline{S}_e}{\partial Q_i}\Delta x_i\right)\right] \tag{5.B.8}$$

in which

$$\frac{\partial \overline{S}_f}{\partial h_i} = 2\overline{S}_{f_i}\left(\frac{d\overline{n}/dh_i}{\overline{n}_i} - \frac{5B_i}{6\overline{A}} + \frac{dB_i/dh_i}{3\overline{B}}\right) \tag{5.B.9}$$

$$\frac{\partial \bar{S}_f}{\partial h_{i+1}} = 2\bar{S}_{f_i}\left(\frac{d\bar{n}/dh_{i+1}}{\bar{n}_i} - \frac{5B_{i+1}}{6\bar{A}} + \frac{dB_{i+1}/dh_{i+1}}{3\bar{B}}\right) \tag{5.B.10}$$

$$\frac{\partial \bar{S}_f}{\partial Q_i} = 2\bar{S}_{f_i}\left(\frac{d\bar{n}/dQ_i}{\bar{n}_i} + \frac{1}{2Q}\right) \tag{5.B.11}$$

$$\frac{\partial \bar{S}_f}{\partial Q_{i+1}} = 2\bar{S}_{f_i}\left(\frac{d\bar{n}/dQ_{i+1}}{\bar{n}_i} + \frac{1}{2Q}\right) \tag{5.B.12}$$

$$\frac{\partial \bar{S}_e}{\partial h_i} = \left(\frac{2\bar{S}_{e_i} B_i V_i^2}{A_i\left(V_{i+1}^2 - V_i^2\right)}\right) \tag{5.B.13}$$

$$\frac{\partial \bar{S}_e}{\partial h_{i+1}} = \left(\frac{-2\bar{S}_{e_i} B_{i+1} V_{i+1}^2}{A_{i+1}\left(V_{i+1}^2 - V_i^2\right)}\right) \tag{5.B.14}$$

$$\frac{\partial \bar{S}_e}{\partial Q_i} = \left(\frac{-2\bar{S}_{e_i} V_i}{\left(V_{i+1}^2 - V_i^2\right)A_i}\right) \tag{5.B.15}$$

$$\frac{\partial \bar{S}_e}{\partial Q_{i+1}} = \left(\frac{2\bar{S}_{e_i} V_{i+1}}{\left(V_{i+1}^2 - V_i^2\right)A_{i+1}}\right) \tag{5.B.16}$$

$$dB_i / dh_i = \Delta B_i / \Delta h_i \tag{5.B.17}$$

$$V_i = Q_i / A_i \tag{5.B.18}$$

$$V_{i+1} = Q_{i+1} / A_{i+1} \tag{5.B.19}$$

$$d\bar{n}/dh_i = \frac{\Delta\bar{n}}{\Delta\hat{h}}\frac{d\hat{h}}{dh_i} \tag{5.B.20}$$

$$d\bar{n}/dQ_i = \frac{\Delta\bar{n}}{\Delta\hat{Q}}\frac{d\hat{Q}}{dQ_i} \tag{5.B.21}$$

$$\hat{h} = \frac{h_m + h_{m+1}}{2} \tag{5.B.22}$$

$$\hat{Q} = \frac{Q_m + Q_{m+1}}{2} \tag{5.B.23}$$

and

$$\left.\begin{aligned} d\hat{h} / dh_i &= 0 \quad \text{if } i \neq m \\ d\hat{h} / dh_i &= 1/2 \quad \text{if } i = m \end{aligned}\right\} \tag{5.B.24}$$

$$\left.\begin{aligned} d\hat{Q} / dQ_i &= 0 \quad \text{if } i \neq m \\ d\hat{Q} / dQ_i &= 1/2 \quad \text{if } i = m \end{aligned}\right\} \tag{5.B.25}$$

References

Can, E. K. and Houck, M. H., 1984. Real-time reservoir operations by goal programming, *J. Water Resour. Plan. Manage.*, ASCE, **110**, 297-309.

Fread, D. L., 1982, National Weather Service operational dynamic wave model, Hydrologic Research Laboratory, U.S. National Weather Service, Silver Spring, Maryland.

Hsin, J. K., 1980, The optimal control of deterministic econometric planning models, PhD dissertation, Dept. of General Business, The University of Texas, Austin.

Jamieson, D. G. and Wilkinson, J. C., 1972, River Dee research program, 3, a short-term control strategy for multipurpose reservoir systems, *Water Resour. Res.* **8**, 911-920.

Lasdon, L. S. and Mantell, J., 1978, A GRG algorithm for econometric control problems, *Ann. Econ. Social Mange.* 6:51.

Lasdon, L. S. and Waren, A. D., 1983, GRG2 user's guide, Dept. of General Business, The University of Texas, Austin.

Luenberger, D. G., 19~~73~~84, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Menlo Park.

Powell, M. J. D., 1978, Algorithms for nonlinear constraints that use Lagrangian functions, *Mathmatical Programming* 14:2.

Tennessee Valley Authority, 1974, Development of a comprehensive TVA water resource management program, Technical Report, Div. of Water Cont. Plan., Tenn. Valley Auth., Knoxville.

Texas Water Development Board, 1971, Engineering data on dams and reservoirs in Texas, part III, Report No. 126, Austin, Texas.

U.S. Army Corps of Engineers, Hydrologic Engineering Center, 1973a, HEC-5, Simulation of Flood Control and Conservation Systems, Davis, California, 1973a.

U.S. Army Corps of Engineers, Hydrologic Engineering Center, 1973b, HEC-5C, A Simulation Model for System Formulation and Evaluation, Davis, California, 1973b.

Unver, O., Mays, L. W., and Lansey, K., 1987, Real-time flood management model for Highland Lake system, *J Water Resour. Plan. Mange.* ASCE **WR5**(3), 620-638.

Wanakule, N., Mays, L. W., and Lasdon, L., 1986, Optimal management of large-scale aquifers: methodology and applications, *Water Resour. Res.* **22**: 4, 447-466.

Wasimi, S. A. and Kitanidis, P. K., 1983, Real-time forecasting and daily operation of a multireservoir system during floods by linear quadratic Gaussian control, *Water Resour. Res.* **19**, 1511-1522.

Windsor, J. S., 1973, Optimization model for the operation of flood control systems, *Water Resour. Res.* **9** [HY5], 1219-1226.

Yazicigil, H., 1982, Optimal operation of a reservoir system using forecasts, PhD Dissertation, Purdue University, West Lafayette.

CHAPTER 6

WATER DISTRIBUTION SYSTEM OPERATION

6.1 Problem Identification

6.2 Problem Formulation

6.3 Problem Solutions

- 6.3.1 Overview
- 6.3.2 The Reduced Problem
- 6.3.3 Solution of the Reduced Problem
- 6.3.4 Computation of Reduced Gradient

6.4 Application

References

6.A Simulation Model

6.B Computation of Basic Elements

6.C Program Structure

Chapter 6

WATER DISTRIBUTION SYSTEM OPERATION

6.1 Problem Identification

A methodology based upon solving a large scale nonlinear programming (NLP) problem is presented for the optimal operation of pumping stations in water distribution systems. Optimal operation refers to the scheduling of pump operation that results in the minimum operating cost for a given set of operating conditions. The methodology is based upon an optimal control framework which interfaces a nonlinear optimization model with a hydraulic simulation model. The objective function is to minimize pumping cost over a planning horizon and the constraint set includes system constraints, which account for the hydraulics involved in a water distribution system, bound constraints on decision variables, and other constraints that may reflect operator preferences or system limitations.

There are a myriad of reasons why pumping stations operate inefficiently: including (Amabee, 1992) 1) pumps which were incorrectly selected; 2) pumps which have worn out; 3) limited capacity in the transmission/distribution system; 4) limited storage capacity; 5) inefficient operation of pressure (hydropneumatic) tanks; 6) inadequate or inaccurate telemetry equipment; 7) inability to automatically or remotely control pumps and valves; 8) penalty due to time-of-day or seasonal energy pricing; 9) lack of understanding of demand or capacity power charges; 10) operator error; and 11) suboptimal control strategies. The optimal control problem for a water distribution system is complicated by the fact that the mathematical problem can be very large in the number of constraints, many

of which are nonlinear, and the large number of decision variables that are non-linear. This is complicated even further by the fact that the controls (pumps on and off) are discrete. Several approaches using dynamic programming (DP) have been proposed (Solanas and Vergés, 1974; Solanas and Montolio, 1987; Cohen, 1982; Joalland and Cohen, 1980; Carpentier and Cohen, 1985; Coulbeck and Orr, 1985; Sabet and Helweg, 1985; Zessler and Shamir, 1985; and Ormsbee, et al., 1987). All of these DP approaches suffer from the curse of dimensionality limiting the size of problems (number of pumps, storage facilities, and size of network) that can be considered; and as a result the DP approaches are only applicable to very small systems. Other previous techniques (Fallside and Perry, 1975 and Coulbeck and Sterling, 1978) that were not based upon dynamic programming were also not very successful. Chase and Ormsbee (1989) proposed a nonlinear programming approach based upon using a nonlinear programming optimizer and a hydraulic simulator to solve the hydraulic constraints of the optimizer. Brion ( ) and Brion and Mays (1991) presented a methodology to solve the problem on a discrete time optimal control problem. This methodology is presented in this chapter.

6.2 Problem Formulation

Consider a water distribution system composed of J nodes, M pipes, P pumps, K primary loops, F fixed grade nodes, and S storage tanks. The mathematical statement of the optimal pump operation problem considering T time periods is to minimize the energy cost, Z_p, given as

$$\text{Min } Z_p = \text{Minimize} \sum_{t}^{T} \sum_{P}^{P} \frac{1}{EFF_{pt}} UC_t \frac{0.746\, \gamma Q_{pt} H_{pt}}{550} D_{pt} \tag{6.2.1}$$

where EFF_{pt} is the efficiency of pump p in time period t; UC_t is the unit pumping cost (\$/KWH) during time period t; γ is the specific weight of water (lb/ft^3); and D_{pt} is the length of time pump p operates during time period t (hr).

The constraints that have to be satisfied at all time periods include the conservation of mass at nodes,

$$\sum_{i} (q_{i,j})_t = Q_{jt} \qquad j = 1,\ldots, J \text{ and } t = 1,\ldots,T \tag{6.2.2}$$

where $(q_{i,j})_t$ is the flowrate in the pipe connecting nodes i and j during time step t; and Q_{jt} is the external demand at node j during time period t. This constraint, which is linear in $(q_{i,j})_t$, assumes that the fluid is incompressible and is written for each node j in the network.

The conservation of energy for primary loops is

$$\sum_{i,j \in k} h_{kt} - \sum_{p \in k} H_{pt} = 0 \qquad k = 1,\ldots, K \text{ and } t = 1,\ldots,T \tag{6.2.3}$$

where h_{kt} is the head loss in the pipe connecting nodes i and j contained in primary loop k at time t; and H_{pt} is the pumping head delivered by pump p in primary loop k at time t.

The conservation of energy for paths between two points of known total grade (fixed grade nodes) is

$$\sum_{i,j \in f} h_{ft} - \sum_{p \in f} H_{pt} = \Delta E_f \qquad f = 1,.., F\text{-}1 \text{ and } t = 1,...,T \qquad (6.2.4)$$

where h_{ft} is the head loss in the pipe connecting nodes i and j contained in path f at time t; H_{pt} is the pumping head delivered by pump p in path f at time t; and ΔE_f is the difference in total grade, expressed as elevation plus gage pressure, between two fixed grade nodes (FGN's) located at both ends of path f. This constraint, which is nonlinear in pipe flowrate, is written for F-1 paths where F is the number of fixed grade nodes in the network for all time periods. Constraint equation (4) is a special case of constraint equation (3). In fact, energy conservation by (3) and (4) apply to primary loops and pseudo-loops (i.e., independent path equations), respectively. The total number of the above hydraulic constraints, all equality constraints in this case, is J + K + F - 1. The total number of unknowns, the M pipe flows, is the same as the number of equations.

The pump operation problem is, inherently, an extended period simulation problem. For this type of analysis, water levels, E_{st}, in storage tanks for the current time period are functions of water levels from the previous time period, which can be expressed as

$$E_{s\,t} = f(E_{s\,t-1}) \qquad s = 1,\ldots,S \text{ and } t = 1,\ldots,T \qquad (6.2.5)$$

This relationship involves the flowrate in the pipe connected to each tank evaluated at the previous time period.

The lower and upper bounds on the length of time pump P operates, D_{pt}, within each time period are given as

$$\Delta_{t\,min} \leq D_{pt} \leq \Delta_{t\,max} \qquad p = 1,\ldots,P \text{ and } t = 1,\ldots,T \qquad (6.2.6)$$

where $\Delta_{t\,min}$ can be zero in order to simulate pump line closing and $\Delta_{t\,max}$ is the length of one time period. This constraint limits the operating time of a pump within a given time period t. D_{pt} is a nonnegative number which cannot exceed the total length of a time period. The smaller the time period used, the more closely continuous pump operation is approximated. D_{pt} appears implicitly in (5).

The pressure head bounds on nodal heads are

$$\underline{H}_{jt} \leq H_{jt} \leq \overline{H}_{jt} \qquad j = 1,\ldots,J \text{ and } t = 1,\ldots,T \qquad (6.2.7)$$

where $\underline{H}_{jt}$ and $\overline{H}_{jt}$ are the lower and upper bounds, respectively, on the pressure head, H_{jt}, at each node j at time t. No universally accepted values for either bound exists. Normally, the minimum desired pressure at the demand nodes fall in the range of 20 to 40 psi. This may be true during average loading conditions but may be significantly lowered during emergency situations such as when a fire breaks out. The upper bounds, on the other hand, are fixed by structural limits on the pipes. They depend on the type of material used as well as the age of the pipe. This constraint is extended to

handle bounds on storage capacities in tanks expressed in terms of water surface elevations.

The bounds on the tank water surface elevations are

$$\underline{E}_{st} \leq E_{st} \leq \overline{E}_{st} \qquad s = 1,\ldots,S \text{ and } t = 1,\ldots,T \qquad (6.2.8)$$

where $\underline{E}_{st}$ and $\overline{E}_{st}$ are the lower and upper bounds, respectively, on the water surface elevation, E_{st}, for each storage tank s during time t. These storage bounds can be imposed for all time periods. Normally, these bounds correspond to physical limits of the tank. During the last time period, a tighter bound is usually placed on all tanks whereby all tank levels are preferred to revert back to the level at the beginning of the first time period. This is evident from a practical point of view since at the end of the night rate period, which usually has the cheapest rate and is the start of the simulation, storage tanks are preferred to be full. Cohen (1982) stated that optimizing the operation of a network over a limited horizon, say 24 hours has no meaning without the requirement of some periodicity in operation. A simple way to do this is to constrain all final states or tank levels to be the same as the initial states within some tolerance. In the methodology, it was decided that the final water surface elevation in each tank be approximately the same as its initial level and that its lower and upper bounds be expressed as functions of the initial water surface elevation.

The above formulation results in a large scale nonlinear programming problem where the $(q_{i,j})_t$, H_{jt}, E_{st} and D_{pt} are the decision variables. Chase and Ormsbee (1989) also used similar decision variables in their nonlinear formulation. Additional bound constraints on system characteristics, such as pump/pipe flowrate at each time period and total energy consumption, can be

imposed. Prespecified operating rules, such as limits on the number of times a pump can be turned on and off during the entire planning horizon, can also be considered. However, a mixed integer-nonlinear programming formulation would result, one that requires an optimization technique different from the one presented in this study. Thus, these constraints were not implemented.

6.3 Problem Solution

6.3.1 Overview

The problem is formulated in an optimal control framework where an optimal solution to the problem is arrived at by interfacing a hydraulic simulation code with a nonlinear optimization code. The hydraulic simulation model is used to implicitly solve the hydraulic constraints that define the flow phenomena each time the optimizer needs to evaluate these constraints. A general formulation of the problem is stated as follows:

$$\text{Minimize energy costs} = f\,(\mathbf{H},\mathbf{Q},\mathbf{D}) \tag{6.3.1}$$

subject to

a. Conservation of flow and energy constraints and pump operation (6.2.2) - (6.2.5)

$$\mathbf{G}\,(\mathbf{H},\mathbf{Q},\mathbf{D},\mathbf{E}) = \mathbf{0} \tag{6.3.2}$$

b. Bands on pump operation time (6.2.6)

$$\Delta_t \leq \mathbf{D} \leq \Delta_t \tag{6.3.3}$$

c. Nodal pressure head bands (6.2.7)

$$\mathbf{H} \leq \mathbf{H} \leq \overline{\mathbf{H}} \tag{6.3.4}$$

d. Storage bands and final tank levels (6.2.8)

$$\underline{\mathbf{E}} \leq \mathbf{E} \leq \overline{\mathbf{E}} \tag{6.3.5}$$

6.3.2 The Reduced Problem

First, the decision variables are partitioned into two sets such that one set can be expressed in terms of the other. Let D_{pt} be the set of "control" or independent variables while H_{jt} and E_{st} form the set of dependent or "state" variables. The justification is based on the implicit function theorem (Luenberger,1984), which states (dropping subscripts for brevity): if H(D*) and E(D*) solve the hydraulic constraint equations for D = D* and the basis matrix of the equations is nonsingular, then H(D) and E(D) exists in the neighborhood of D*. Thus for the given set of D, there is always a solution of H and E which satisfies the hydraulic equations implying that H and E can be written in terms of D, or H(D) and E(D). Similarly, the objective function can be written in terms of the control variables and is referred to as the reduced objective function F.

$$F(D_{pt}) = f\,(H_{jt}(D_{pt}),\ E_{st}(D_{pt})) \tag{6.3.6}$$

By implicitly expressing the state variables in terms of the control variables, a smaller nonlinear optimization problem can be solved explicitly by an NLP code while delegating the burden of satisfying the hydraulic constraint equations (thus establishing the implicit functions H(D) and E(D)) through a hydraulic simulation code. The hydraulic simulation code KYPIPE by Wood (1980) for water distribution networks not only solves the hydraulic constraint equations but likewise satisfies the storage bound constraints from period t = 1 to t = T-1. This added incentive is reflected in the discussion that follows. The reduced problem takes the form

$$\text{Min } Z_{RP} = \text{Min } F(H_{jt}\,(D_{pt}),\ E_{st}\,(D_{pt}),\ D_{pt}) \tag{6.3.7}$$

subject to upper and lower bounds on D_{pt}, E_{sT}, and H_{jt} where E_{sT} and H_{jt} are written in terms of D_{pt} as $E_{sT}(D_{pt})$ and $H_{jt}(D_{pt})$; and the reduced objective function F is expressed as a function of D_{pt}. The hydraulic simulator (KYPIPE) satisfies the set of hydraulic constraint equations, including the final time period S storage bound constraints. It also calculates the implicit functions $E_{sT}(D_{pt})$ and $H_{jt}(D_{pt})$. Figure 6.3.1 shows the linkage between the optimization and simulation codes. p6-41

control = NB
State = basic

In mathematical programming, the control and state variables may also be referred to as nonbasic and basic variables, respectively. Improvements in the objective function of the nonlinear programming problem is attained by a systematic variation of the nonbasic variables. NLP codes restrict the step size by which the nonbasic variables change so as not to violate their bounds. In an optimal control formulation, the determination of step size of the control (non-basic) variables does not take into consideration the values of the (basic) state variables. If the bounds on these state variables are violated, more iterations would be required to obtain a feasible solution.

As mentioned earlier, the procedure initially reduces the problem size by expressing the pressure heads, H_{jt}, and tank levels, E_{st}, as functions of pump durations, D_{pt}. A penalty function method offers a further reduction in problem size. In general, the method incorporates the upper and lower state bounds into the objective function in the form of penalty terms. Specifically the simple penalty function method approximates the optimal solution to the problem from exterior points. However, it has ill-conditioning effects when the penalty weights become excessively large, i.e., the problem terminates before finding the real local optimizer (Bazaraa, 1979).

A variant of the general penalty method, the augmented Lagrangian method (Hsin, 1980), is used to formulate the optimal control problem. The mathematical derivation of the equations using the augmented Lagrangian method is given in Brion (1990).

Each state bound constraint is converted to the form of a penalty term and is added to the original objective function. The head bound penalty terms and final storage bound penalty terms are added to the original objective function f to develop the augmented Lagrangian function, AL,

$$\text{Min} \quad AL\,(H_{jt}, E_{st}, D_{pt}, \mu_{jt}, \eta_{st}, \sigma_{jt}, \beta_{st}) = f\,(H_{jt}, E_{st}, D_{pt})$$

$$+ \frac{1}{2}\sum_i \sigma_i \left\{\min\left[0, b_i - \frac{\mu_i}{\sigma_i}\right]\right\}^2 - \frac{1}{2}\sum_i \frac{\mu_i^2}{\sigma_i} \tag{6.3.8}$$

The index i is a one-dimensional index representation of the double index (j,t) for head bound penalty terms and double index (s,t) for storage bound penalty terms; and σ_i and μ_i are penalty weights and Lagrange multipliers for the i-th penalty term, respectively. Furthermore, b_i is the bound constraint violation term which is negative if a bound constraint violation indeed occurs. Bound constraints, which are sets of upper and lower limit constraints, are incorporated into the objective function as a single penalty term. At any given time, only one of the two bounds may be violated. State bound constraints are now considered in the determination of the step size used in the search for the optimal solution.

Combining the two approaches presented so far, the original problem is recast into the reduced problem with the augmented Lagrangian formulation, given by

$$\text{Min} \quad L(D_{pt}, \mu_{jt}, \eta_s, \sigma_{jt}, \beta_s) = f(H_{jt}, E_{st}, D_{pt})$$

$$+ \frac{1}{2}\sum_{tj} \sigma_{jt}\left\{\min\left[0, b_{jt} - \frac{\mu_{jt}}{\sigma_{jt}}\right]\right\}^2 - \frac{1}{2}\sum_{tj} \frac{\mu_{jt}^2}{\sigma_{jt}}$$

$$+ \frac{1}{2}\sum_{s} \beta_s\left\{\min\left[0, c_s - \frac{\eta_s}{\beta_s}\right]\right\}^2 - \frac{1}{2}\sum_{s} \frac{\eta_s^2}{\beta_s} \qquad (6.3.9)$$

subject to

$$0 \le D_{pt} \le \Delta_t \qquad p = 1,\ldots,P \text{ and } t = 1,\ldots,T \qquad (6.3.10)$$

Lagrange multipliers, μ_{jt}, and penalty weights, σ_{jt}, are associated with the head bound penalty terms while η_s and β_s are the Lagrange multipliers and penalty weights, respectively, associated with the final storage bound penalty terms. Head bound violations, b_{jt}, and final storage bound violations, c_s, are defined as

$$b_{jt} = \min(\underline{b}_{jt}, \bar{b}_{jt}) \text{ with } \underline{b}_{jt} = H_{jt} - \underline{H}_{jt},\ \bar{b}_{jt} = \overline{H}_{jt} - H_{jt} \qquad (6.3.11)$$

$$c_s = \min(\underline{c}_s, \bar{c}_s) \text{ with } \underline{c}_s = E_{sT} - \underline{E}_{sT},\ \bar{c}_s = \overline{E}_{sT} - E_{sT} \qquad (6.3.12)$$

Again, the penalty terms associated with the storage bound constraints for t =1 to T were not considered because they are implicitly satisfied by the hydraulic simulation code (KYPIPE).

6.3.3 Solution of the Reduced Problem

The reduced problem, equations (6.3.9) - (6.3.10), is the final formulation to be solved by the NLP optimizer, such as the generalized reduced gradient code GRG2 by Lasdon and Waren (1986). The set of decision

variables is narrowed down to include only the control variables. The upper and lower bounds imposed on the control variables are simply handled by the NLP algorithm in order to approach or maintain feasibility during its search of an optimal solution to the problem. The reduction steps of the solution process are summarized in Figure 6.3.2. It should be noted that in this figure, the variables are evaluated only within an inner level optimization as explained next.

p.6-42 p 6-43

Based on the above formulation, the proposed solution methodology can be summarized as follows (also see Figure 6.3.3). The method is a two-level optimization where the final set of variables are partitioned into the control variables, D_{pt}, and the augmented Lagrangian variables μ, η, σ, and β, and has the objective function

$$\text{Min } Z_{RPAL} = \text{Min } \{ \text{Min } L (D_{pt}, \mu_{jt}, \eta_{st}, \sigma_{jt}, \beta_{st}) \} \tag{6.3.13}$$

Initially, the penalty weights and Lagrange multipliers, are fixed and the optimizer is used to solve the inner level minimization for D_{pt}. Given these optimal values of D_{pt}, the outer level minimization involving μ, η, σ and β is then carried out. If a convergence criteria is not met or if an iteration limit is yet to be reached the outer level variables are updated and passed inside the inner level minimization and the procedure is repeated. The inner level or loop is a nonlinear optimization subproblem which takes different forms based on the current values of the penalty weights and Lagrange multipliers. This loop is solved repeatedly by the generalized reduced gradient code GRG2.

The outer level or loop is the master problem and is solved by a heuristic based on Fletcher's (1975) algorithm. Using updating formulas

(Brion, 1990), the values of the Lagrange multipliers and penalty weights are revised. The types of updating formulas vary and the one that follows the steepest descent criteria has the form:

$$\begin{aligned} \mu_i^{m+1} &= \mu_i^m - \sigma_i c_i, && \text{if } c_i \le \mu_i/\sigma_i \\ &= 0, && \text{if } c_i > \mu_i/\sigma_i \end{aligned} \tag{6.3.14}$$

where m is the outer loop iteration index. Convergence is checked at each iteration by evaluating a convergence factor r and testing its value against a preset convergence limit. At the very beginning, r is set to a large absolute number and is then updated by the formula

$$r = \max \mid \min\{\sigma_i, \mu_i/\sigma_i\} \mid. \tag{6.3.15}$$

If r is less than or equal to the preset convergence limit, then the augmented Lagrangian method is said to have converged, otherwise the outer loop undergoes another iteration.

6.3.4 Computation of Reduced Gradients

The state of a water distribution system can be defined by a specification of all pipe flow rates or all nodal heads at any given time of the day. This is evident by the use of the Hazen-Williams equation which ensures that the pipe flow rates determine nodal pressures and vice versa. Furthermore, the solution of the system hydraulic equations would represent system response to a steady-state simulation for independent loading conditions or demand patterns. For an extended period simulation where a series of demand patterns make up the daily cycle loads, tank continuity equations link the submatrices representing the different sets of system equations for each demand pattern, as shown in Figure 6.3.4. Thus, for a time sequence of

demands, the state space would comprise the entire set of nodal pressure heads plus tank levels. The state is said to be defined at the junction nodes and at the storage tanks. The gradient computations are performed using D as the vector of control variables and H as the contiguous vector of state variables. Water level at a fixed grade node is not part of the state space because its value is constant whatever the demand pattern. No system equation is required at this location in order to perform a pipe network simulation.

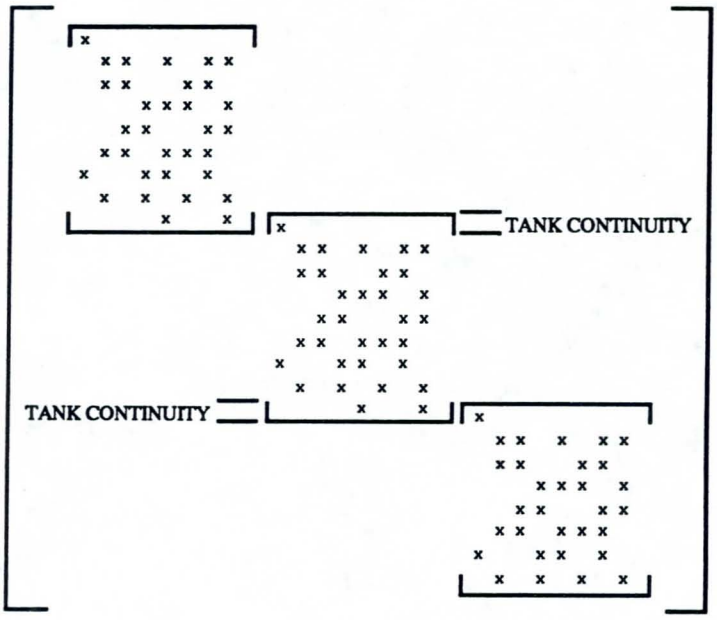


Figure 6.3.4 Matrix Structure for Time Sequence of Demands

The derivatives of the reduced objective function, Equation 4.9, with respect to the control variables are called the reduced gradients. Functions F(D) and H(D) are implicit functions and all the gradients cannot be directly calculated. All are differentials. However, these functions cannot be evaluated in closed form. The two-step procedure of Lasdon and Mantell

(1978) can be used to compute the reduced gradients. First, by applying the chain rule:

$$\partial F/\partial D = \partial f/\partial D + (\partial f/\partial H)^T \cdot \partial H/\partial D \tag{6.3.16}$$

Only the matrix $\partial H/\partial D$ cannot be directly determined given the objective function. In order to evaluate this, the hydraulic constraints, or the G-equations, are used. Taking the derivative of the flow constraint equations with respect to D:

$$\partial G/\partial D = \partial g/\partial D + (\partial g/\partial H)^T \cdot \partial H/\partial D = 0 \tag{6.3.17}$$

which can be written as:

$$\partial H/\partial D = -(\partial g/\partial H)^{-1} \cdot \partial g/\partial D \tag{6.3.18}$$

Substituting in (5.1),

$$\partial F/\partial D = \partial f/\partial D - [(\partial f/\partial H)^T \cdot (\partial g/\partial H)^{-1}\, \partial g/\partial D] \tag{6.3.19}$$

and defining

$$\pi^T = (\partial f/\partial H)^T \cdot (\partial g/\partial H)^{-1} \tag{6.3.20}$$

from which π, the Lagrange multipliers, can be calculated by solving the system of linear equations:

$$(\partial g/\partial H)^T \pi = \partial f/\partial H \tag{6.3.21}$$

In this procedure the derivatives of the network equations and objective function are needed with respect to each variable, D and H. Using these gradients, the Lagrange multipliers can be computed by (6.3.21). With these multipliers and the remaining known gradients, the derivative of the

reduced objective can be calculated from (6.3.19). The Lagrange multiplier π has a useful physical meaning. At the optimum, it defines the change in the objective function due to a small change in the duration of pumping at the corresponding pump locations .

Although the flow equations are solved by a hydraulic simulator in order to obtain q and H, the head equations and tank continuity equations are equally complete and accurate representations of the state of the system. The loop equations are easier to solve at the expense of requiring analysis of the geometry of the network. The node equations, on the other hand, are more manageable for gradient computation since they basically represent the system node connectivity matrix at each time step, with the tank continuity equations as links between adjacent time steps. This is shown in Figure . To calculate the reduced gradients of the objective function, four terms; $\partial f/\partial H$, $\partial f/\partial D$, $\partial g/\partial H$, and $\partial g/\partial D$ are to be computed. The first two terms are actually the derivatives of the objective function, i.e., the cost function with respect to the state and control variables, respectively. The last two terms, taken together, form the Jacobian matrix of the G system equations, i.e., the first partial differential of the matrix g(H,D) with respect to the vector (H,D). In compact form,

$$J(H,D) = [\,\partial g/\partial H, \partial g/\partial D\,] = [\,B, C\,] \qquad (6.3.22)$$

where the basis matrix B (= $\partial g/\partial H$) is assumed to be nonsingular at all points and is comprised of diagonal block matrices; and the right partition C (= $\partial g/\partial D$) is defined only at the lower portions of each diagonal block corresponding to the tank continuity equations. The elements of B are nonlinear functions of H while the elements of C are linear in terms of D.

For a time sequence of demands, the submatrices of B, which represent separate demand patterns, are symmetrical and are very sparse. This special structure is lost when tank continuity equations link these submatrices to form the whole time sequence of demands (Figure 6.3.4). The Lagrange multipliers are computed for the entire sequence of demands which involves the solution to a sparse matrix $\partial g/\partial H$. Sparse matrix solvers, such as the one used in KYPIPE, are available but a more efficient way, based on optimal control theory, considers each demand pattern individually rather than as a whole.

The calculation of π starts at the final time step T by solving the system of equations

$$(\partial g_T/\partial H_T)^T \cdot \pi_T = (\partial f/\partial H)_T \tag{6.3.23}$$

Multipliers for time steps T-1 to 1 are solved backward in succession by

$$(\partial g_t/\partial H_t)^T \cdot \pi_t = (\partial f/\partial H)_t - \pi_{t+1} \cdot \partial g_{t+1}/\partial H_t \qquad \text{for } t = T\text{-}1,\ldots,1 \tag{6.3.24}$$

Finally the computation of the reduced gradients is done by appropriate substitution of the Lagrange multipliers in

$$(\partial F/\partial D)_t = (\partial f/\partial D)_t - \pi_t \cdot \partial g_t/\partial D_t \qquad \text{for } t = 1,\ldots, T \tag{6.3.25}$$

Appendix 7.B shows how to compute each entry in the Jacobian matrix $[\partial g/\partial H, \partial g/\partial D]$, and the partial derivatives of the objective function with respect to the state and control variables, $\partial AL/\partial H$ and $\partial AL/\partial D$, respectively. The derivatives of the augmented Lagrangian (penalty) terms in (6.3.9), $\partial AL/\partial H$, are with reference to (6.3.16) ~~(3.6.16)~~, the augmented Lagrangian terms, Λ_A in (4.10) can be represented by

$$\Lambda_A(H,\mu,\sigma) = \sum_i \begin{cases} -\mu_i c_i + \frac{1}{2}\sigma_i c_i^2, & \text{if } c_i \le \frac{\mu_i}{\sigma_i} \\ -\frac{1}{2}\frac{\mu_i^2}{\sigma_i}, & \text{otherwise} \end{cases} \tag{6.3.26}$$

where H represents the whole vector of states, nodal heads plus tank levels, and c_i is defined by (6.3.11) or (6.3.12). The derivatives of these terms with respect to the state variables become

$$\frac{\partial \Lambda_A}{\partial H} = \begin{cases} 0, & \text{if } c_i > \frac{\mu_i}{\sigma_i} \\ -\mu_i + \sigma_i c_i, & \text{if } c_i \le \frac{\mu_i}{\sigma_i} \text{ and } c_i = h_i - \underline{h}_i \\ \mu_i - \sigma_i c_i, & \text{if } c_i \le \frac{\mu_i}{\sigma_i} \text{ and } c_i = \overline{h}_i - h_i. \end{cases} \tag{6.3.27}$$

6.4. Application

Brion (1990) developed a computer code, PMPOPR that interfaces GRG2 and KYPIPE for determining the optimal operation of pumping stations in water distribution systems. A very extensive system of software was developed in order to combine the augmented Lagrangian algorithm into a single cohesive computer code. The program structure is presented in Appendix 6.C. Brion (1990) and Brion and Mays (1991) presented application of the model to a pressure zone in Austin, Texas for a typical 24-hour day. (See Figure 1.5.) This pressure zone consisted of 126 pipes, 98 nodes, 5 pressure watchpoints, 3 pumps, 1 storage tank, and 12 - 2 hour time periods.

Various computer runs of the model showed savings in pumping costs ranging from 5.2 to 17.3 percent over the actual operating costs for the day.

In the future water distribution systems may be operated using optimal control systems consisting of the components shown in Figure 6.4.1. Such systems will be able to provide operators with an optimal operating policy for each pump station in a water distribution system. The optimal control system can be directly integrated with a SCADA (Supervisory Control And Data Acquisition) system to provide the link between the optimal control software and the system operator.

APPENDIX 6 A

SIMULATION MODEL

6.A.1 Simulator Equations

One approach in steady state analysis of pipe networks would be the generation and solution of mass continuity and energy conservation equations in terms of discharge in each pipe section. The resulting equations are referred to as loop equations, as opposed to node equations wherein the same set of equations are expressed in terms of total grades at junction nodes. At any rate, the solution to both formulations should yield flowrate in each pipe and pressure head at each node. It has been shown that the loop equations have superior convergence characteristics. These equations are used in the simulation model KYPIPE (Wood, 1980).

The mathematical relationship between number of pipes, primary loops, junction nodes and fixed grade nodes for all pipe systems is

$$M = N + LP + F_{max} - 1 \qquad (6.A.1)$$

where M = number of pipe sections, N = number of junction nodes, LP = number of primary loops, and F_{max} = number of fixed grade nodes.

For each junction node a continuity relationship can be written wherein flow into the junction equals flow out of the junction:

$$\left(\sum q_{in}\right)_i - \left(\sum q_{out}\right)_i = \left(q_{ext}\right)_i, \qquad i = 1, 2, \ldots, N \qquad (6.A.2)$$

where $q_{ext,\,i}$ represents the external inflow or demand at the junction node. There are N of these junction equations. For each primary loop, the energy equation can be written for the pipe sections in the loop as follows

$$\left(\sum h_L\right)_l = \left(\sum E_p\right)_l, \qquad l = 1, 2, \ldots, LP \qquad (6.A.3)$$

where hL = energy loss in a pipe in the loop (minor losses included), and E_p = energy put into the liquid by a pump in the loop. For the case wherein no pumps exists within the loop, the sum of the energy losses within the loop becomes zero. F_{max} - 1 independent energy equations can be written for paths between any two fixed grade nodes as follows

$$(\Delta E)_f = \left(\sum h_L\right)_f - \left(\sum E_p\right)_f, \qquad f = 1, 2, \ldots, F_{max} - 1 \qquad (6.A.4)$$

where ΔE is the difference in total grade between the two fixed grade nodes. Note that (6.A.3) can be considered as a special case of (6.A.4) where the difference in total grade, ΔE, is zero for a path which forms a closed loop. Jointly, there are LP + F_{max} - 1 of these path equations. (6.A.1)-(6.A.3), otherwise known as loop equations, constitute a set of p simultaneous nonlinear algebraic equations which describe steady state flow analysis for the solution of the flowrate in each pipe. Path equations can be further modified so as to express them in terms of the flowrates.

The energy loss in a pipe, h_L, is the sum of the line loss h_{LP} and the minor loss h_{LM}. The line loss expressed in terms of the flowrate is given by

$$h_{LP} = K_p q^n \qquad (6.A.5)$$

where K_p is a constant which is a function of line length (L), diameter (D), and roughness (C), or friction factor (f), and n is an exponent. The values of K_p and n depend on the energy loss expression used in the analysis. Using the Hazen-Williams equation

$$K_p = \frac{X L}{C^{1.852} D^{4.87}} \tag{6.A.6}$$

and n = 1.852. In this equation, X = 4.73 for English units or X = 10.69 for SI units. Using the Darcy-Weisbach equation

$$K_p = \frac{8 f L}{g D^5 \pi^2} \tag{6.A.7}$$

and n = 2. The minor loss in a pipe section expressed in terms of flowrate is given by

$$h_{LM} = K_M q^2 \tag{6.A.8}$$

where K_M is a constant which is the sum of the minor loss coefficients which, in turn, are functions of the number and type of fittings used.

The energy put into the liquid by a pump can be described by operating data. The within-range operation can be mathematically represented by a polynomial as follows

$$E_p = A + B q + C q^2 \tag{6.A.9}$$

where A, B, and C are coefficients describing the characteristics of the pumps. Combining (6.A.5), (6.A.8), and (6.A.9), and making the appropriate substitutions in (6.A.4), with (6.A.3) taken as a special case of (6.A.4), we have

$$\left(\Delta E\right)_f = \left(\sum (K_p q^n + K_M q^2)\right)_f - \left(\sum (A + Bq + Cq^2)\right)_f, \; f=1,\ldots,LP+F_{max}-1 \qquad (6.A.10)$$

A set of M simultaneous equations in terms of the unknown flowrates is formed by the N continuity equations (6.A.2) and the $LP+F_{max}-1$ energy equations (6.A.10). The solution of the above equations, jointly called loop equations, involves the use of numerical or iterative methods since the unknown flowrates could not be explicitly expressed in terms of the other variables in the system of equations. Wood and Charles (1972) suggested that the linearization scheme is the most reliable and efficient algorithm in solving the loop equations. A discussion of the algorithm follows.

6.A.2 Algorithm for the Solution of the Loop Equations: The Linear Method

A simple gradient method that handles the nonlinear flowrate in (6.A.10) is used in the proposed method. Consider a single pipe section within a given path. A single term in (6.A.10) would then represent the grade difference across a pipe section carrying a flowrate q such that

$$f(q) = K_p q^n + K_M q^2 - (A + Bq + Cq^2) \qquad (6.A.11)$$

Equation (6.A.11) is linearized using a first-order Taylor series approximation about the point $\mathbf{q} = \mathbf{q}_i$, where subscript i refers to the previous iteration or an initial guess. If we let $\Delta E = f(\mathbf{q})$, where $\mathbf{q} = \{q_1, q_2, \ldots, q_j, \ldots\}$ = set of flowrates in the pipes contained in path f then

$$f(\mathbf{q}) = f(\mathbf{q}_i) + \sum_j \left.\frac{\partial f}{\partial q}\right|_{\mathbf{q}_j = \mathbf{q}_{j,i}} \left(\mathbf{q}_j - \mathbf{q}_{j,i}\right) \qquad (6.A.12)$$

where

$$f(q_i) = \sum_j K_p q_{j,i}^n + \sum_j K_M q_{j,i}^2 - \sum_j \left(A + Bq_{j,i} + Cq_{j,i}^2 \right) \tag{6.A.13}$$

and the gradient of the general form of (6.A.11) is

$$\frac{\partial f}{\partial q} = \sum_j nK_p q_j^{n-1} + \sum_j 2K_M q_j - \sum_j \left(B + Cq_{j,i} \right) \tag{6.A.14}$$

(6.A.12) is employed to formulate $LP+F_{max}-1$ energy equations which when combined with the already linear N continuity equations (6.A.2) form a set of M simultaneous linear equations in terms of flowrate in each pipe.

A unit flow in all pipes is assumed initially and the system of equations is solved using a modified sparse matrix solver developed by the Harwell Industrial Research Group. The computed flowrates become the new "assumed" flowrates and are used to evaluate the linearized equations and obtain a second solution. This procedure or trial is repeated until no significant change in the "assumed" and computed flowrates is observed or a maximum number of trials is reached. The first criterion is met if the relative accuracy, defined as the sum of the changes in flowrate between the last two trials divided by the sum of the flowrates, becomes less than a specified value (default value = .005). Given the final values of pipe flows, starting from a fixed grade node, (6.A.5) and/or (6.A.8) can be written for each pipe to compute pressure heads.

During the development of the code it was observed that the solution tend to oscillate about the actual solution such that the average of the

previous two iterations tend to be very close to the true solution. Thus, q_i was redefined as

$$q_i = \left(\frac{q_{i-1} - q_{i-2}}{2} \right) \tag{6.A.15}$$

and this new value was used in (6.A.12). Because all flows are computed simultaneously convergence occurs faster than other procedures (Wood, 1980). A high degree of accuracy is achieved using only 4-8 trials even for a very large system.

APPENDIX 6.B

COMPUTATION OF BASIS ELEMENTS

The following discussion deals with gradient computations. First, the Jacobian [$\partial g/\partial H$, $\partial g/\partial D$] will be derived taking into consideration the pipe, tank and pump components of the system. Second, the derivative of the implicit derivatives of the reduced objective with respect to H and D will be shown.

The structure of the left partition of the Jacobian is called the basis matrix B. The node system of equations and tank continuity equations are solved for the unknown nodal total heads and tank water levels at all time steps during an extended period simulation.

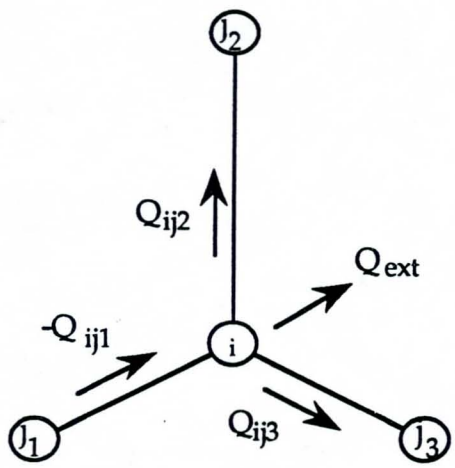


Figure 6.B.1 Node Connected with Three Pipes

Water reaches a node by way of conveyance through pipes converging at the node (Figure 6.B.1). The node equation for a node i with only pipe connections for any time step is

$$G_i: \sum_j q_{ij} + Q_{ext} = \sum_j \frac{K_{ij}\, C_{ij}\, D_{ij}^{2.63}}{L_{ij}^{0.54}} [\mathrm{sign}(H_i - H_j)] [\,|H_i - H_j|\,]^{0.54} = 0 \qquad (6.B.1)$$

where: K_{ij} is a pipe parameter equal to $(4.737 \cdot L)/(C^{1.852} \cdot D^{4.87})$ for English units and $(10.675 \cdot L)/(C^{1.852} \cdot D^{4.87})$ for S.I.; C_{ij}, the Hazen-Williams roughness coefficient; D_{ij}, the length of pipe ij; L_{ij} length of pipe ij; H_i, total head at node i; and H_j, total head at node j (or tank level if pipe ij connects node i to a storage tank j). Double subscript ij denotes a pipe connecting nodes i and j. The order by which these subscripts are written determine flow direction. Flow is positive if there is outflow from node i and negative otherwise. External source Qext is positive for demand and negative for supply.

The derivative of (6.B.1) with respect to any H_j (including tank levels) is

$$\begin{aligned} \partial g_i/\partial h_j &= 0.54\, (K_{ij}\, C_{ij}\, D_{ij}^{2.63})/(L_{ij}^{0.54})\, [\mathrm{sign}(H_i - H_j)]\, [\,|H_i - H_j|\,]^{-0.46}(-1) \\ &= -0.54\, (K_{ij}\, C_{ij}\, D_{ij}^{2.63})/(L_{ij}^{0.54})\, [\mathrm{sign}(H_i - H_j)]\, [\,|H_i - H_j|\,]^{0.54} \\ &= -0.54\, \mathrm{abs}(q_{ij})/(H_i - H_j). \qquad (6.B.2) \end{aligned}$$

Similarly, the relation exists for $\partial G_i/\partial H_i$ and can be shown to be

$$\partial g_i/\partial h_i = \sum_i 0.54\, \mathrm{abs}(q_{ij})/(H_i - H_j). \qquad (6.B.3)$$

A variation of (6.B.3) exists for a node i connected to a fixed grade node f. Two cases will be discussed: a node connected to a fixed grade node without a pump in between, and a node connected to a fixed grade node with a pump in between. The no pump case merely requires the addition of the single

term 0.54 abs(q_{if})/(H_i-H_f) in (6.B.3), where H_f is the water surface elevation at the fixed grade node.

If pumps are online and if the KYPIPE pump representation is used, the derivation is as follows. Pumps operating within its three-point operating data has an exponential Q-H curve of the form

$$Q_p = -[(H1 - H_p)/C]^{(1/B)} \tag{6.B.4}$$

where: H1 is the pump cut-off head, or the head above which the pump can no longer sustain the system pressure requirement; and B and C are pump curve parameters expressed as functions of the three-point (H1-Q1, H2-Q2, H3-Q3) pump data. They are computed by

$$B = \log[(H1-H3)/(H1-H2)]/[\log(Q3)] \tag{6.B.5}$$

$$C = (H1 - H2)/Q_2^B. \tag{6.B.6}$$

The pump head H_p is the energy imparted by the pump on the fluid and is a linear function of the head at the downstream node i,

$$H_p = H_i + h_{fiL} - H_f \tag{6.B.7}$$

where: h_{Lfi} is the head loss due to friction along pipe fi. Taking the derivative of (6.B.4) with respect to H_i yields

$$\begin{aligned} \partial Qp/\partial Hi &= -[1/(BC^{(1/B)})]\,(H1-H_p)^{((1/B)-1)}\,\partial(H1-H_p)/\partial H_i \\ &= [1/(BC^{(1/B)})]\,(H1-H_p)^{((1/B)-1)}\,\partial(H_p)/\partial H_i. \end{aligned} \tag{6.B.8}$$

From (D.5),

$$\partial H_p/\partial H_i = \partial H_i/\partial H_i + \partial h_{Lfi}/\partial H_i - \partial H_f/\partial H_i. \tag{6.B.9}$$

The first term in (6.B.9) is identity while the last term is zero. For suction pipes, the second term is negligible but whose contribution can be derived as follows. In English units, if we let $K_p = (4.737 \cdot L)(C^{1.852} \cdot D^{4.87})$, then

$$\begin{aligned} \partial h_{Lfi}/\partial H_i &= \partial (K_p Q_p^{1.852})/\partial H_i \\ &= 1.852\, K_p Q_p^{0.852}\, \partial Q_p/\partial H_i. \end{aligned} \tag{6.B.10}$$

Substituting (6.B.10) into (6.B.11) and then into (6.B.8), we obtain

$$\partial Q_p/\partial H_i = (1/(BC^{(1/B)}))\,(H1 - H_p)^{((1/B)-1)}\,[\,1+1.852\, K_p Q_p^{0.852} \partial Q_p/\partial H_i\,]. \tag{6.B.11}$$

If we let $X1 = 1.852\, K_p Q_p^{0.852}$ and $X2 = [\,(H1 - H_p)^{((1/B)-1)}\,]\,/\,[\,BC^{(1/B)}\,]$, we get

$$\begin{aligned} \partial Q_p/\partial H_i &= X2\,[\,1 + X1\, \partial Q_p/\partial H_i\,] \\ &= X2 + X1 \cdot X2\, \partial Q_p/\partial H_i. \end{aligned} \tag{6.B.12}$$

Finally, we have

$$\partial Q_p/\partial H_i = X2/(1-X1 \cdot X2). \tag{6.B.13}$$

If the pump operates above its third operating point (see section 1.3.3), KYPIPE represents the pump head discharge relationship by a straight-line formula,

$$Q_p = -\,[\,(H_p - A)/S\,] \tag{6.B.14}$$

where:

$$S = -B \cdot C \cdot (Q3)^{(B-1)} \tag{6.B.15}$$

$$A = H3 - S \cdot Q3. \tag{6.B.16}$$

Proceeding as before,

$$\begin{aligned} \partial Q_p/\partial H_i &= (-1/S)\, \partial(H_p - A)/\partial H_i \\ &= (-1/S)\, \partial H_p/\partial H_i. \end{aligned} \tag{6.B.17}$$

By substituting (6.B.10) into (6.B.9) and then into (6.B.17), we obtain

$$\partial Q_p/\partial H_i = (-1/S)\, [\, 1 + 1.852\, K_p\, Q_p^{0.852}\, \partial Q_p/\partial H_i \,] \tag{6.B.18}$$

$$\begin{aligned} \partial Q_p/\partial H_i &= (-1/S)\, [\, 1 + X1\, \partial Q_p/\partial H_i \,] \\ &= -[\, 1/S + (X1/S)\, \partial Q_p/\partial H_i \,]. \end{aligned} \tag{6.B.19}$$

Finally,

$$\begin{aligned} \partial Q_p/\partial H_i &= -1/[\, S(1 + X1/S) \,] \\ &= -1/(S + X1). \end{aligned} \tag{6.B.20}$$

Equations (6.B.13) or (6.B.20) are added to (6.B.3) in cases wherein a source pump delivers flow into node i.

The analysis of tanks considers a sequence of demands with which tank levels vary with time. During each time step, the tank level is considered fixed and flow distribution is computed. A storage tank may be considered as a node with its water level as a state variable (Figure (6.B.2). Continuity at the

storage tank is a mass balance equation from one time step to the next and can be defined as

$$E_{s\,t} = E_{s\,t-1} - q_{sj\,t-1} \cdot \frac{D_t}{A_s} \qquad (6.B.21)$$

where:

E_{st} = water level in storage tank s at time step t,

E_{st-1} = water level in storage tank s at time step t-1,

q_{sj-1} = flow rate in pipe connecting storage tank s to the rest of the system via node j and is considered positive if tank is emptying,

D_t = duration of time between time step t and time step t-1, and

A_s = horizontal cross-sectional area of tank s.

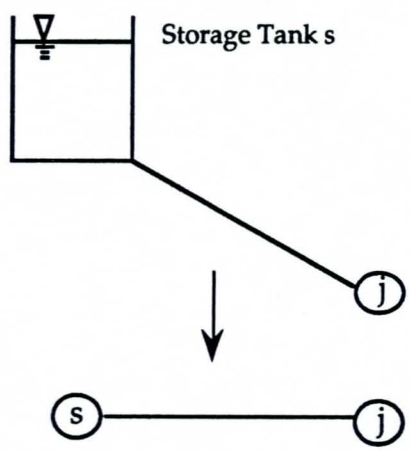


Figure 6.B.2 Tank Representation
(adapted from Lansey, 1987)

Using Hazen-Williams equation and transposing all terms on one side of the equality sign, we have

$$G_{st}: -E_{s\,t} + E_{s\,t-1} - \frac{K_{sj}\,C_{sj}\,D_{sj}^{2.63}}{L_{sj}^{0.54}}\left[\mathrm{sign}(E_{s\,t-1}-H_{j\,t-1})\right]\left[\,|E_{s\,t-1}-H_{j\,t-1}|\,\right]^{0.54} = 0. \quad (6.B.22)$$

This equation links the time sequence of system equations corresponding to the different demand patterns. The required derivatives used in computing the Lagrange multipliers are

$$\partial g_{st}/\partial E_{st} = -1 \quad (6.B.23)$$

$$\begin{aligned}\partial g_{st}/\partial E_{st-1} &= 1 - 0.54\,[\mathrm{sign}(E_{st-1}-H_{jt-1})]\,K_p\,(D_t/A_s)\,[\,|E_{st-1}-H_{jt-1}|\,]^{-0.46} \\ &= 1 - [\,(0.54\,|q_{sjt-1}|)/(E_{st-1}-H_{jt-1})\,]\cdot[\,D_t/A_s\,] \quad (6.B.24)\end{aligned}$$

$$\begin{aligned}\partial g_{st}/\partial H_{jt-1} &= -0.54\,[\mathrm{sign}(E_{st-1}-H_{jt-1})]\,K_p\,(D_t/A_s)\,[\,|E_{st-1}-H_{jt-1}|\,]^{-0.46}\,(-1) \\ &= [\,(0.54\,|q_{sjt-1}|)/(E_{st-1}-H_{jt-1})]\cdot[\,D_t/A_s\,] \quad (6.B..25)\end{aligned}$$

$$\begin{aligned}\partial g_{st}/\partial D_t &= -[\mathrm{sign}(E_{st-1}-H_{jt-1})]\,(K_p/A_s)\,[\,|E_{st-1}-H_{jt-1}|\,]^{-0.54} \\ &= -q_{sjt-1}\,/\,A_s \quad (6.B.26)\end{aligned}$$

where K_p is previously defined. The procedure applied to compute the Lagrange multipliers for multiple time steps is discussed in section 6.4.4. Essentially, computation of the element of the state space is done forward in time while the multipliers are solved backward in time. Equations (6.B.24) and (6.B.25) may be converted to apply from time steps T-1 to 1. The more useful forms are

$$\partial g_{st+1}/\partial E_{st} = 1 - [\,(0.54\,|\,q_{sjt}\,|\,)\,/\,(E_{st} - H_{jt})]\cdot[\,D_{t+1}/A_s\,] \tag{6.B.27}$$

$$\partial g_{st+1}/\partial H_{jt} = [\,(0.54\,|\,q_{sjt}\,|)\,/\,(E_{st} - H_{jt})\,]\cdot[\,D_{t+1}/A_s\,]. \tag{6.B.28}$$

The partial derivative of the original objective function with respect to the control variable is the gradient of the pumping cost. It is always positive since pumping cost increases as pumps are used at longer durations. Mathematically, it is given by

$$\partial f/\partial D_{pt} = (1/EFF_{pt})\cdot UC_t\cdot(0.746\,\gamma\,Q_{pt}\,H_{pt})\,/\,550. \tag{6.B.29}$$

where all terms are defined previously.

The partial derivative of the original objective function with respect to state variable H can be derived by examining the effect of pumps on the rest of the network. First, the objective function is expressed in terms of the state variable H. The pump may be considered as a special type of pipe with a head-flow relationship mathematically defined by the pump characteristic curve. If placed online, additional energy is brought into the flow. A pipe with a pump will still experience energy loss by friction in the direction of flow. Thus, the total head at the downstream end of a pipe with a pump is equal to the total head at the upstream end minus the frictional head loss plus the pump head or

$$H_{ds} = H_{us} - h_L + H_p. \tag{6.B.30}$$

The downstream end would be a node in the system while the upstream end would be a fixed grade node. By using the previous notation, (6.B.30) is transformed into

$$H_p = H_i - H_f + h_{L_{fi}}. \tag{6.B.31}$$

H_i is a state variable and $h_{L_{fi}}$ is a function of the pipe flow which is also the pump discharge. Since pump discharge is a function of pump head, we can deduce that it is also a function of the state of the system by way of (6.B.31). By the same reasoning, pump efficiency is also a function of the state of the system. For illustration purposes, the partial derivative of the cost function with respect to the state variables will be derived for the type of pump head-discharge-efficiency relationship used in the first and second example applications.

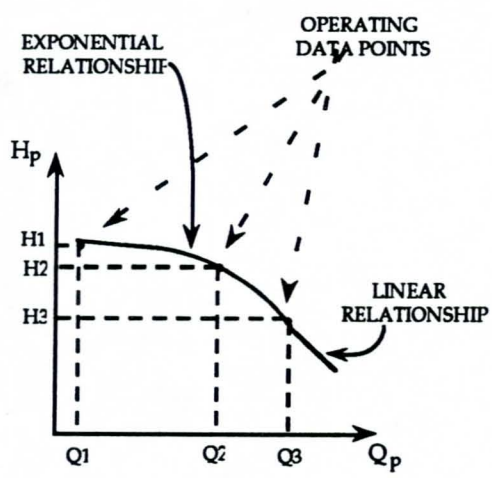


Figure 6.B.3 Pump H - Q Curve Representation in KYPIPE

KYPIPE fits an exponential curve to define the pump characteristics (Figure 6.B.3). Based on a user specified set of three operating points, KYPIPE determines the coefficients of the exponential function defined in (6.B.4). Above the third operating point, KYPIPE uses an alternate curve by extending the characteristic beyond the third point using the linear form (6.B.14). Assuming a maximum efficiency, EFF_{max}, at the second operating point that linearly decreases to a minimum, EFF_{min}, at the other operating points; a triangular efficiency "curve" is defined (Figure 6.B.4) and has the form

$$
\begin{aligned}
EFF &= EFF_{max} && \text{if } Q_p = Q2 \\
&= a \cdot Q_p + b && \text{if } Q1 < Q_p < Q3 \\
&= EFF_{min} && \text{otherwise}
\end{aligned}
\qquad (6.B.32)
$$

where a and b are the slope and intercept, respectively. The three regions marked in Figures 6.B.3 and 6.B.4 require three different types of gradients for the cost function. They are:

Case I:

$$\partial f/\partial H_i = K4 \cdot C4 \{ [k6 \cdot H_{ip}^2) + k4(H_{1p}-H_{1pd})] / [k6 \cdot H_{1p} + k4]^2 \} \quad (6.B.33)$$

Case II:

$$\partial f/\partial H_i = K4 \cdot C4 \{ [k5 \cdot H_{ip}^2) + k2(H_{1p}-H_{1pd})] / [k5 \cdot H_{1p} + k2]^2 \} \quad (6.B.34)$$

Case III:

$$\partial f/\partial H_i = K2 \cdot C4 \{ 2 H_p - A \} \quad (6.B.35)$$

where:

$$H_{1p} = (H1 - H_p)^{(1/B)}$$

$$H_{1pd} = (-1/B)\,(H1 - H_p)^{((1/B)-1)}$$

$$k1 = (EFF_{max}-EFF_{min})/(Q2-Q1)$$

$$k2 = EFF_{max} - k1 \cdot Q2$$

$$k3 = (EFF_{max}-EFF_{min})/(Q2-Q3)$$

$$k4 = EFF_{max} - k3 \cdot Q2$$

$$k5 = k1 / C^{(1/B)}$$

$$k6 = k3 / C^{(1/B)}$$

$$K1 = UC_t \cdot (0.746\, \gamma\, D_{pt})/550$$

$$K2 = K1/(EFF_{min} \cdot S)$$

$$K3 = K1/(EFF_{min} \cdot C^{(1/B)})$$

$$K4 = K1/C^{(1/B)}$$

$$C1 = 1.852\, K_p\, Q_p^{0.852}$$

$$C2 = H_{1pd} / C^{(1/B)}$$

$$C3 = C1 \cdot C2$$

$$C4 = 1 + C3 / (1 - C3).$$

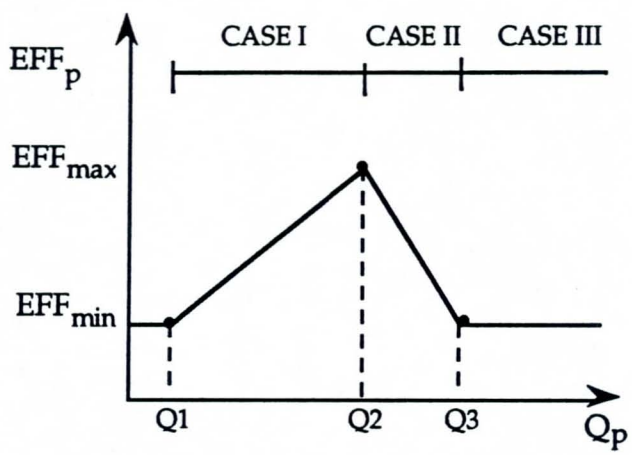


Figure 6.B.4 Triangular Pump Efficiency Curve

Other forms of functions relating H_p, Q_p, EFF_p and even the unit pumping cost, UC_p (if a per usage rate structure is used) require different forms of the gradient of the cost function with respect to the state variable.

REFERENCES

Bazaraa, M.S. and C.M. Shetty, Nonlinear Programming: Theory and Algorithms, John Wiley & Sons, Inc., New York, 1979.

Brion, L.M., "Methodology for Optimal Operation of Pumping Stations in Water Distribution Systems," Ph.D. Dissertation, Department of Civil Engineering, The University of Texas at Austin, Austin, Texas, 1990.

Brion, L.M. and L.W. Mays, "Methodology for Optimal Operation of Pumping Station in Water Distribution System," Journal of Hydraulic Engineering, ASCE, Vol. 117, No. 11, pp. 1551-1571, Nov. 1991.

Carpentier, P., and G. Cohen, "Decomposition, Coordination and Aggregation in the Optimal Control of a Large Water Supply Network," Proceedings of the Ninth Triennial World Congress of IFAC, Budapest, Hungary, 2-6 July 1984, Vol. 6, J. Gertler and L. Keviczky, eds., Pergamon Press, pp. 3207-3212, 1985.

Chase, D.V. and L.E. Ormsbee, Optimal Pump Operation of Water Distribution System with Multiple Storage Tanks, Proceedings, ASCE Conference on Water Resources Planning and Management, Sacramento, California, pp. 733-736, 1989.

Cohen, G., "Optimal Control of Water Supply Networks," Chapter 8 in Optimization and Control of Dynamic Operational Research Models, Vol. 4, S.G. Tzafestas, ed., North-Holland Publishing Company, Amsterdam, pp. 251-276, 1982.

Coulbeck, B., and C.H. Orr, "Optimized Pumping in Water Supply Systems," Proceedings of the Ninth Triennial World Congress of IFAC, Budapest, Hungary, 2-6 July 1984, Vol. 6, J. Gertler and L. Keviczky, eds., Pergamon Press, pp. 3175-3180, 1985.

Coulbeck, B., and M.J.H. Sterling, "Optimised Control of Water Distribution Systems," IEE Proceedings, Vol. 125, No. 9, pp. 1039-1044, October 1978.

Duan, N., L.W. Mays, and K.E. Lansey, "Optimal Reliability-Based Design and Analysis of Pumping Systems for Water Distribution Systems," Journal of Hydraulic Engineering, ASCE, Vol. 116, No. 2, pp. 249-268. February 1990.

Fallside, F. and P.F. Perry, "Hierarchical Optimization of Water-Supply Network," IEE Proceedings, Vol. 122, No. 2, pp. 202-208, February 1975A.

Fletcher, R., "An Ideal Penalty Function for Constrained Optimization," Journal of the Institute of Mathematics and its Applications, No. 15, pp. 319-342, 1975.

Hsin, J.K., "The Optimal Control of Deterministic Econometric Planning Models," Ph.D. Dissertation, Department of General Business, The University of Texas at Austin, Austin, Texas, 1980.

Joalland, G., and G. Cohen, "Optimal Control of a Water Distribution Network by Two Multilevel Methods," Automatica, Vol. 16, pp. 83-88, 1980.

Lansey, K.E. and L.W. Mays, Optimization Model for Water Distribution System Design, Journal of Hydraulic Engineering, ASCE, Vol. 115, No. 10, pp. 1401-1418, Oct. 1989.

Lasdon, L.S. and A.D. Waren, GRG2 User's Guide, Department of General Business, The University of Texas at Austin, Austin, Texas, 1986.

Lasdon, L.S., A.D. Waren, A. Jain and M.S. Ratner, "Design and Testing of A Generalized Reduced Gradient Code for Nonlinear Programming," ACM Trans. on Mathematical Software, Vol.4, No.1, pp. 34-50, 1978.

Luenberger, D.G., Linear and Nonlinear Programming, second edition, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1984.

Ormsbee, L.E. (ed.) Energy Efficient Operation of Water Distribution Systems, Report by the ASCE Task Committee on the Optimal Operation of Water Distribution System, Research Report No. UKCE 9104, Dept. of Civil Engineering, University of Kentucky, Lexington, Kentucky, 1991.

Ormsbee, L.E., T.M. Walski, D.V. Chase and W.W. Sharp, "Techniques for Improving Energy Efficiency at Water Supply Pumping Stations," Technical Report EL-87-16, Environmental Laboratory, US Army Engineer Waterways Experiment Station, Vicksburg, Mississippi, 242p., November, 1987.

Powell, M.J.D, "Algorithms for Nonlinear Constraints That Use Lagrangian Functions," Mathematical Programming, M.L. Balinski, ed., Vol. 14, No. 2, pp. 224-248, March 1978.

Sabet, M.H., and O.J. Helweg, "Cost Effective Operation of Urban Water Supply System Using Dynamic Programming," Water Resources Bulletin, Vol. 21, No. 1, pp. 75-81, 1985.

Solanas, J.L., and J.M. Montolio, "The Optimum Operation of Water Systems," International Conference, Computer Applications for Water Supply and Distribution, Leicester Polytechnic, Leicester, England, 1987.

Solanas, J.L., and M. Vergés, "Approximations Procedure and its Application to Automatic Operational Control of Water Distribution Systems," IFAC-IFORS Symposium, Varan, Bulgaria, October 1974.

Wanakule, N., L.W. Mays, and L.S. Lasdon, Optimal Management of Large-Scale Aquifers: Methodology and Application, Water Resources Research, AGU, Vol. 22, No. 4, pp. 447-466, April 1986.

Unver, O.I. and L.W. Mays, Model for Real-Time Optimal Flood Control Operation of a Reservoir System, Water Resources Management, Kluwer Academic Publishers, Vol. 4, pp. 21-46, 1990.

Wood, D.J., "Computer Analysis of Flow in Pipe Networks Including Extended Period Simulation-User's Manual," Office of Engineering Continuing Education and Extension, University of Kentucky, Lexington, Kentucky, 1980.

Zessler, U. and U. Shamir, "Optimal Operation of Water Distribution Systems," unpublished report, Technion - Israel Institute of Technology, Haifa, Israel, 1985.

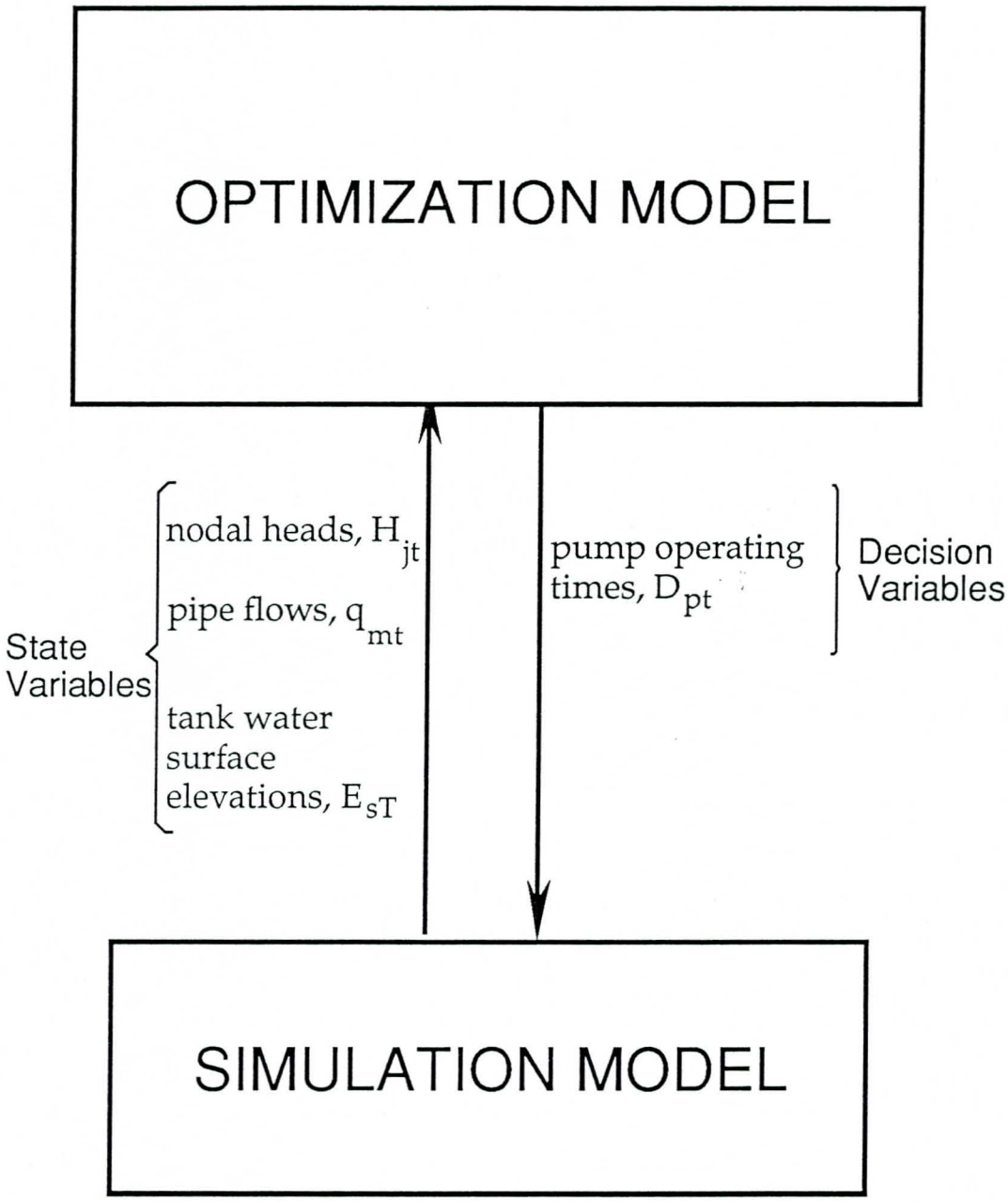


FIG. ~~7~~ 6.3.1 Optimization-Simulation Model Linkage

Formulation	Problem Size
Minimize Cost = $f(q_{mt}, H_{jt}, E_{st}, D_{pt})$ subject to: 1. Conservation Laws 2. Tank Continuity 3. Pumping Duration Bounds 4. Storage Bounds 5. Head Bounds	variables $(M + J + S + P) \cdot T$ constraints $(M + 2S + P + J) \cdot T$

↓ use simulation model to solve implicit function

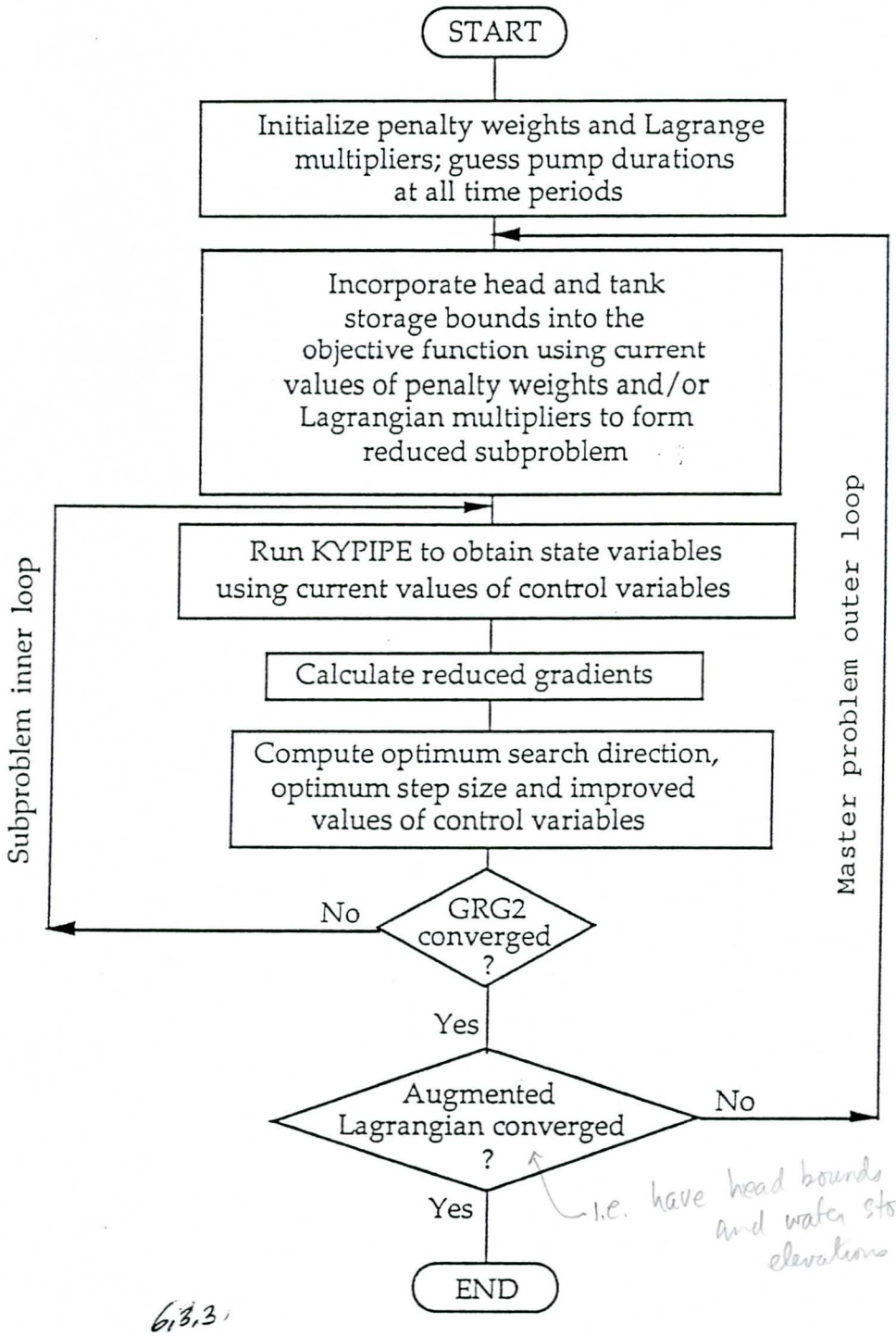
Minimize Cost = $F(H_{jt}(D_{pt}), E_{sT}(D_{pt}), D_{pt})$ subject to: 3. Pumping Duration Bounds 4. Final Storage Bounds 5. Head Bounds	variables $(J + P) \cdot T + S$ constraints $(P + J) \cdot T + S$

↓ use augmented Lagrangian method to incorporate head bound and storage bound constraints into the objective function

Minimize Cost = $L(D_{pt})$ subject to: 3. Pumping Duration Bounds	$P \cdot T$ variables $P \cdot T$ constraints

M = number of pipes; J = number of junction nodes;
S = number of storage tanks; P = number of pumps;
T = number of time periods

FIG. ~~2~~ 6.3.2 Transformation of Optimal Pump Operation Problem



i.e. have head bounds and water storage tank elevations been satisfied?

Figure ~~4.6.1~~ 6.3.3 Flow Chart of Overall Optimization Model

Tracy
Can you
do this one
on the
computer.

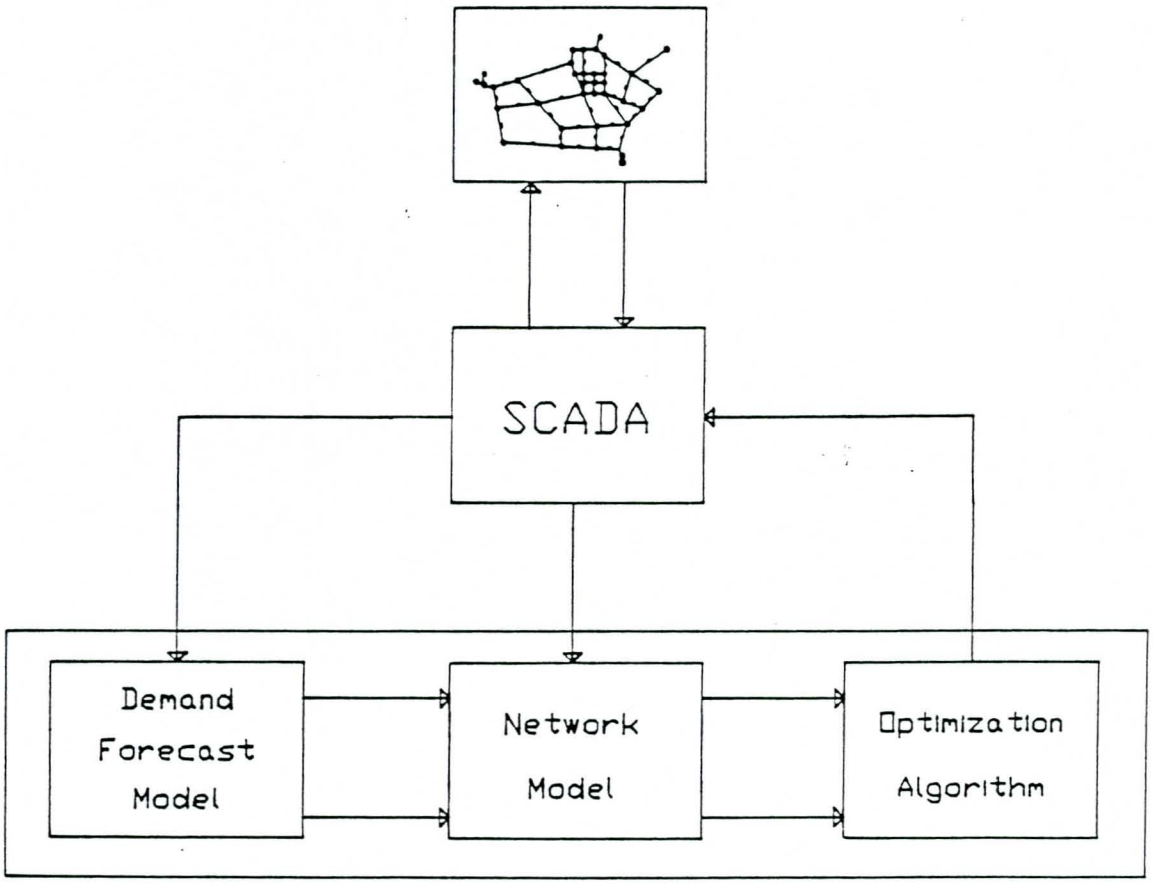


Figure ~~4.1~~ 6.9.1 Optimal Control System (Ormsbee, 1991)

CHAPTER 7

OPTIMIZATION OF FRESHWATER INFLOWS TO ESTUARIES

7.1 Problem Identification

7.2 Problem Formulation

7.2.1 Hydrodynamic Transport Models for Estuaries

7.2.2 Constraints

7.2.3 Alternative Management Model Strategies

7.2.4 Chance Constraints Formulation of Harvest Equations

7.3 Problem Solution

7.3.1 Overview

7.3.2 The Reduced Problem

7.3.3 Solution Procedure

7.3.4 Computation of Reduced Gradient

7.4 Model Development and Application

7.4.1 Overview of Mode

7.4.2 Application

References

Appendix

7.A HYD-SAC Simulation Model

7.B Chance Constraint Formulation

7.C Program Structure of OPTFLOW

CHAPTER 7

OPTIMIZATION OF FRESHWATER INFLOWS TO ESTUARIES

7.1 Problem Identification

In many areas of the country, particularly the Gulf Coast states, California and elsewhere in the world, the freshwater discharge of rivers has become a limited commodity, for which the need for freshwater inflow to maintain the productivity of coastal estuaries must compete with the demands of upstream users, viz. municipal and industrial uses, and agriculture. The desired approach to water-resources management is to optimize flow into the estuary (by minimizing the total volume of flow, or by maximizing the diversions and storage within limits of water rights and capacity, or both) while preserving an acceptable habitat in specific regions of the estuary to accommodate the requirements of key organisms. Salinity has been long established as an index to ecological habitat in an estuary because it measures the relative proportion of fresh water to sea water. Even for those organisms which are euryhaline, i.e. whose physiology can accommodate wide excursions of salt concentration, salinity still provides a useful habitat index because of other "information" contained in the freshwater ratio, such as nutrient supply, sediment and detritus, or stenohaline components of the food web.

A key element in this optimization problem is the mathematical relation between salinity in the estuary and flow, $S = F(Q)$. Usually the relation is based upon statistical association, i.e., a regression form established

from field data. The Texas Water Development Board (TWBD) has made particularly extensive application of this approach in establishing freshwater inflow requirements, as a part of its Bays and Estuaries Program. The work of the TWDB probably represents the most extensive incorporation of water requirement for estuaries within a larger water-resources management context, and the Texas bays are an excellent model for similar problems elsewhere.

The statistical regression S = F(Q) proves to be extremely noisy because of the variability in salinity. In the case of the Texas bays, nearly the entire possible range of salinity values can be found in the historical field data for any given value of concurrent inflow. The reasons for this are twofold. First, the value of salinity in a given region of the bay is dependent upon several other factors in addition to freshwater inflow, notably the various hydrodynamic circulation processes including tides, responses of the bay to meteorological forcing, and the effect of density currents particularly operating in conjunction with deep draft ship channels. Second, the time scale of response of salinity is typically much longer than the variability of freshwater inflow. The value of salinity is the integrated response to perhaps several months of the freshwater inflow "signal."

It should also be noted that the optimization problem as summarized above is in fact time varying, primarily because the salinity requirements of key organisms in the estuary will vary with season through the year, depending upon the life stage of the organism and its presence or absence within the estuary. (Many of the important commercial species are anadromous, migrating into or out of the estuary.) The salinity limits for a specific organism are based upon the statistical association between the

presence of that organism in the estuary (as reflected in catch data or harvest data) and salinity, or upon the physiological dependence upon salinity as revealed in laboratory studies. Thus far, the optimization problem has only been treated on a steady-state basis. Accommodation of the seasonal variation in salinity requirement was made by the TWDB by subdividing the year into several seasons and solving the steady-state problem separately for each season. The most general formulation of the problem, however, should accommodate not only seasonal variation in salinity limits of the organisms, but also seasonal variation in upstream water demands and the specific time response of salinity to freshwater inflow.

The essential weakness in the above formulation is the mathematical expression of salinity dependence upon freshwater inflow to the bay. This chapter reformulates the problem, replacing the statistical regression $S = F(Q)$ with a mathematical model of hydrodynamic transport, relating salinity at a given point in the estuary to a time-varying boundary condition of riverine inflow. Such an approach has the following advantages:

(1) more accurate and self-consistent definition of salinity as a function of flow, enabling greater precision in the optimization results;

(2) explicit incorporation of physical processes other than freshwater inflow affecting salinity in the real system, including tides, meteorology, and internal circulations;

(3) the ability to accommodate time variation in the response of salinity to freshwater inflow, so as to readily generalize to the full time-varying problem (although the optimization problem can also be solved in a steady-state framework with steady inflows);

(4) the ability to accommodate generalization to full time variation in upstream water demands, including seasonality of irrigation and long-term demographic changes;

(5) the ability to consider either averaged inflow, prespecified scenarios of inflow, or long-term simulations using real hydrological data.

In some estuaries, a direct measure of organism abundance is available in the data on commercial fishery landings taken from the estuary. This "harvest" data can be employed as an index of populations of key organisms and analyzed statistically to establish its dependence on freshwater inflow,

$$H_k = f(Q) \qquad (7.1.1)$$

While this might appear superior to the indirect salinity-index approach, the causal connection between flow and harvest may be obscured by unmeasureable parameters of the fishing process such as effort, selectivity and skill, and may be corrupted by poor reporting or the difference between locality of landing (i.e., port) and locality of catch, to say nothing of other environmental variables unrelated to inflow. This regression therefore tends to be noisy and statistically uncertain. On the other hand, it is directly pertinent to the problem, and when the data are available, should be accommodated within the optimization problem, either as an objective function or as a constraint.

7.2 Problem Formulation

7.2.1 Hydrodynamic Transport Simulator for Estuaries

The essence of this chapter is to develop a general methodology for the estuarine freshwater resources management, so that for discussion purposes, the hydrodynamic transport model needed for simulation of temporal and spatial variation of salinity is not restricted to a particular model. The selection of an appropriate model depends on a number of factors such as efficiency, accuracy, complexity, and availability of the model. Even if a desired hydrodynamic transport model has been chosen and applied in the simulation, a better model can always be used to replace it in the future as more efficient models are developed. The formulation of hydrodynamic and transport governing equations varies slightly for each model depending on the various assumption and approximation introduced. The model used for discussion purposes the application in this research is a two-dimensional, finite-difference model. Such a model is used as an example for the formulation of governing equations and their finite-differencing approximations.

The governing equations for the two-dimensional horizontal model are the vertical-averaged equations of momentum, continuity and salinity mass budget: the momentum equation in x-direction,

$$\frac{\partial q_x}{\partial t} - \Omega q_y = -g d \frac{\partial h}{\partial y} - f q q_x + X_w \tag{7.2.1}$$

the momentum equation in y-direction,

$$\frac{\partial q_y}{\partial t} - \Omega q_x = -g d \frac{\partial h}{\partial y} - f q q_y + X_w \tag{7.2.2}$$

the continuity equation,

$$\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y} + \frac{\partial h}{\partial t} = r - e \tag{7.2.3}$$

and the conservation (transport) equation,

$$\frac{\partial s}{\partial t} + \frac{\partial (Us)}{\partial x} + \frac{\partial (Vs)}{\partial y} = \frac{\partial}{\partial x} E_x \frac{\partial s}{\partial x} - \frac{\partial}{\partial y} E_y \frac{\partial s}{\partial y} \tag{7.2.4}$$

where

t is time;

x, and y are horizontal Cartesian coordinates;

q_x and q_y are depth-averaged flow components in x and y directions per unit width;

Ω is the Coriolis parameter equal to 2 w sinf;

w is the angular rotation of the earth;

f is the latitude;

g is the gravitational acceleration;

h is the water surface elevation;

d is the water depth equal to h - z;

z is the bottom elevation;

f is the bottom friction term from the Manning equation;

q is flow per unit width equal to $\sqrt{q_x^2 + q_y^2}$;

X_W is the wind stress per unit density of water in x-direction equal to $K V_w^2 \cos\theta$;

Y_W is the wind stress per unit density of water in y-direction equal to $K V_w^2 \sin\theta$;

K is a wind stress coefficient;

V_W is the wind velocity at 10 meters above the water surface;

q is the wind direction with respect to the x-axis;

r is the rainfall intensity;

e is the evaporation rate;

U and V are the net velocities over a tidal cycle;

s is the vertical-averaged salinity; and

E_X, E_Y are horizontal dispersion coefficients in the x and y directions.

In the momentum equations, the advective terms are neglected and the water density is treated as a constant. The assumption of constant density considerably simplifies the governing equations by decoupling salinity from the momentum equations, but at the expense of neglecting salinity-induced accelerations. The remaining terms in the momentum equations are the

inertia, the Coriolis acceleration, gravity, friction, and wind stress. The precipitation and evaporation terms are also added in the continuity equation for the mass conservation. The transport equation is a linear second order PDE of convective-dispersion equation. The dispersion coefficients are introduced to absorb the density-current fluxes.

Boundary conditions are imposed around the periphery of the estuary including water-land boundaries, partial internal boundaries (e.g., submerged reefs for hydrodynamic equations only), freshwater flows (e.g., river flows, diversions, and return flows), and open saltwater ocean boundaries (tidal excitation). For salinity, $s = s_o$ is imposed at the ocean boundaries, a von Neumann condition (zero flux) at land boundaries, and an open-boundary condition at the inflow points. These boundary conditions can all be functions of time.

The hydrodynamic equations are non-linear first order partial differential equations to solve for three unknowns of flow flux in x and y directions and water surface elevation (q_x, q_y and h). A fully explicit method is used for solving the hydrodynamic equations is a time-centered difference scheme involving time stepping of the "leap frog" type for computations of flows and water surface elevations. Knowing the values at time t, the unknowns q_x, q_y and h can be solved at time (t+1) (derived from equations 7.2.1-7.2.3):

$$q_x^{t+1}(i,j) = \frac{1}{C_x^{t-1}}\left[q_x^{t-1}(i,j) + g\Delta t\left\{\frac{d^t(i,j) + d^t(i+1,j)}{2}\right\}\left\{\frac{h^t(i,j) - h^t(i+1,j)}{\Delta x}\right\}\right]$$

$$+\frac{1}{C_x^{t-1}}\left[X_w^t(i,j)\Delta t + \Omega\bar{q}_y^{t-1}(i,j)\Delta t\right] \tag{7.2.5}$$

$$q_y^{t+1}(i,j) = \frac{1}{C_y^{t-1}}\left[q_y^{t-1}(i,j) + g\Delta t\left\{\frac{d^t(i,j) + d^t(i,j+1)}{2}\right\}\left\{\frac{h^t(i,j) - h^t(i,j+1)}{\Delta y}\right\}\right]$$

$$+\frac{1}{C_y^{t-1}}\left[Y_w^t(i,j)\Delta t - \Omega\bar{q}_x^{t-1}(i,j)\Delta t\right] \tag{7.2.6}$$

$$h^{t+2}(i,j) = h^t(i,j) + \Delta t\left[\frac{q_x^{t+1}(i-1,j) - q_x^{t+1}(i,j)}{\Delta x} + \frac{q_x^{t+1}(i,j-1) - q_x^{t-1}(i,j)}{\Delta y}\right]$$

$$+\Delta t\left[r^{t+1}(i,j) - e^{t+1}(i,j)\right] \tag{7.2.7}$$

where

$$d(i,j) = h(i,j) - z(i,j)$$

$$\bar{q}_x(i,j) = \frac{q_x(i,j) + q_x(i,j+1) + q_x(i-1,j+1) + q_x(i-1,j)}{4}$$

$$\bar{q}_y(i,j) = \frac{q_y(i,j) + q_y(i+1,j) + q_y(i,j-1) + q_y(i+1,j-1)}{4}$$

$$C_x = 1 + \frac{gn^2(i,j)}{2.21\left[\frac{d^t(i,j) + d^t(i+1,j)}{2}\right]^{1/3}}\Delta t\frac{\left[\left\{q_x^{t-1}(i,j)\right\}^2 + \left\{\bar{q}_y^{t-1}(i,j)\right\}^2\right]^{1/2}}{\left[\frac{d^t(i,j) + d^t(i+1,j)}{2}\right]^2}$$

$$s^{t+1}(i,j-1)\left\{ E_y^t(i,j-1)\left[\frac{\Delta t}{\Delta x^2}\right] + V^t(i,j-1)\left[\frac{\Delta t}{2\Delta x}\right]\right\} +$$

$$s^t(i,j)\left\{ 1 - E_y^t(i,j)\left[\frac{\Delta t}{\Delta x^2}\right] - E_y^t(i,j-1)\left[\frac{\Delta t}{\Delta x^2}\right] + V^t(i,j-1)\left[\frac{\Delta t}{2\Delta x}\right] -\right.$$

$$\left. V^t(i,j)\left[\frac{\Delta t}{2\Delta x}\right] + K\Delta t\right\} + s^t(i,j+1)\left\{ -E_y^t(i,j)\left[\frac{\Delta t}{\Delta x^2}\right] - V^t(i,j)\left[\frac{\Delta t}{2\Delta x}\right]\right\}$$

Similarly, the implicit approximation can be written in y-direction at time step (t+2). The resultant linear algebraic equations for the solution of s^{t+1} (or s^{t+2}) can be solved by inversion of a tridiagonal matrix.

7.2.2 Constraints

The mathematical programming model can have the objective of minimizing the sum of freshwater inflows, Q_{tj}, for month t and river j

$$\text{Min} \sum_j \sum_t Q_{tj} \tag{7.2.8}$$

subject to the following constraints:

(1) The nonlinear relationship of estuary salinity and freshwater inflow.

$$\mathbf{G(Q,s) = 0} \tag{7.2.9}$$

(2) Upper ($\bar{s}$) and lower ($\underline{s}$) bounds on the monthly average salinity at a specified location in the estuary, for each river j.

$$\underline{s}_{tj} \le s_{tj} \le \bar{s}_{tj} \tag{7.2.10}$$

$$C_y = 1 + \frac{gn^2(i,j)}{2.21\left[\frac{d^t(i,j) + d^t(i,j+1)}{2}\right]^{1/3}} \Delta t \frac{\left[\left\{q_y^{t-1}(i,j)\right\}^2 + \left\{\bar{q}_x^{t-1}(i,j)\right\}^2\right]^{1/2}}{\left[\frac{d^t(i,j) + d^t(i,j+1)}{2}\right]^2}$$

The alternating direction implicit (ADI) method is used to solve the transport equation. Thus, theoretically, it is unconditionally stable for any size of time or spatial step (because of the implicit). The linear system equations result in a tridiagonal matrix which is efficiently solved using the Thomas algorithm. The ADI method is carried out in two steps. At time step (t+1), the x-derivatives are written in implicit form and y-derivatives in explicit form. At time step (t+2), the direction is switched that the y-derivatives are written implicit form and x-derivatives in explicit form. The resultant two sets of simultaneous equations are solved directly without iteration.

At time step (t+1) the conservation equation (7.2.4) can be approximated in x-direction as,

$$s^{t+1}(i-1,j)\left\{-E_x^{t+1}(i-1,j)\left[\frac{\Delta t}{\Delta x^2}\right] - U^{t+1}(i-1,j)\left[\frac{\Delta t}{2\Delta x}\right]\right\} +$$

$$s^{t+1}(i,j)\left\{1 + E_x^{t+1}(i,j)\left[\frac{\Delta t}{\Delta x^2}\right] + E_x^{t+1}(i-1,j)\left[\frac{\Delta t}{\Delta x^2}\right] - U^{t+1}(i-1,j)\left[\frac{\Delta t}{2\Delta x}\right] +\right.$$

$$\left. U^{t+1}(i,j)\left[\frac{\Delta t}{2\Delta x}\right]\right\} + s^{t+1}(i+1,j)\left\{-E_x^{t+1}(i,j)\left[\frac{\Delta t}{\Delta x^2}\right] + U^{t+1}(i,j)\left[\frac{\Delta t}{2\Delta x}\right]\right\} =$$

(3) Lower limits on the t-th monthly inflows for the j-th river, QI_{tj}, to express seasonal biological requirements, e.g. of the estuarine marsh inundation.

$$Q_{tj} \geq QI_{tj} \tag{7.2.11}$$

(4) The sum of monthly flows must be less than or equal to the upper limit of the total annual inflow, QT_j, from each river j.

$$\sum_t Q_{tj} \leq QT_j \tag{7.2.12}$$

(5) Upper and lower limits on mean monthly flows in seasons for each river j,

$$\underline{QS}_{jm} \leq QS_{jm} \leq \overline{QS}_{jm} \tag{7.2.13}$$

where $QS_{jm} \equiv \frac{1}{N_m} \sum_{t \in M_m} Q_{tj}$; M_m is the set of months in season m and N_m is the number of months in season m.

(6) The nonlinear regression relationship between the harvest of organism k and the seasonal inflow in river j.

$$H_k = \Psi_k \left(QS_{jm} \right) \tag{7.2.14}$$

(7) Lower limits on annual fish harvest, $\underline{H}_k$, by species k.

$$H_k \geq \underline{H}_k \tag{7.2.15}$$

(8) Upper and lower limits on monthly inflows $\left(\overline{Q}_{tj} \text{ and } \underline{Q}_{tj}\right)$ from each river.

$$\underline{Q}_{tj} \leq Q_{tj} \leq \overline{Q}_{tj} \tag{7.2.16}$$

Monthly mean salinity bounds are specified for selected locations. There are two types of upper and lower limits on monthly salinity selected to provided a salinity range. The first type is based on the bounds for viable metabolic and reproductive activity, and the second salinity upper bound selected is the median monthly historical salinity level, or equal to the first type salinity upper bound if it is lower than the median monthly historical salinity level.

7.2.3 Alternative Management Model Strategies

Four alternative formulations of the optimization model can be applied to achieve different management objectives, as summarized below. Other management objectives are possible, and can be similarly formulated within the general framework of (7.2.8) - (7.2.16).

Alternative I The basic formulation of the problem for estuarine management is to minimize the total annual freshwater inflow subject to salinity level control, which will accomplish the requirements of nutrient transport, habitat maintenance, and marsh inundation requirement. The corresponding mathematical model can be formulated as

$$\text{Min} \sum_j \sum_t Q_{tj} \tag{7.2.17}$$

subject to constraints (7.2.9)-(7.2.11) and (7.2.15)-(7.2.16).

Alternative II Maintenance of the fishery harvest. The objective is to minimize the total annual freshwater inflow while satisfying minimum seasonal flow needs to maintain the annual commercial harvest of key species at desired levels, and meeting viability limits for salinity. The constraints for Alternative II are equations (7.2.9)-(7.2.11) and (7.2.13)-(7.2.16).

Alternative III Enhancement of the fishery harvests. It is to maximize the total annual commercial harvest of a selected organism k while meeting viability limits for salinity, satisfying minimum seasonal flow needs, and limiting an annual combined inflow no greater than its historical mean value. The objective is to

$$\text{Max } \mathbf{QS}^T \hat{\beta}_{HK} \tag{7.2.18}$$

subject to Equations (7.2.9)-(7.2.11) and (7.2.15)-(7.2.16), where $\mathbf{QS}^T$ is the transpose of vector of the seasonal freshwater inflow, and $\hat{\beta}_{HK}$ is the vector of estimated coefficients of the harvest regression equation for species k.

The periodic inundation of deltaic marshes serves to maintain shallow protected habitats for postlarval and juvenile stages of several important estuarine species, provides a suitable fluid medium for nutrient exchange processed, and acts as a transport mechanism to move detrital materials from the deltaic marsh into the open estuary (TDWR, 1980; Valiela and Teal, 1974).

In the problem formulation, these stochastic constraints are transformed into probabilistic statements so that each chance-constraint states the probability that the constraint will be satisfied with a specified reliability level. The harvest constraint (7.2.4) can be rewritten in chance-constraint form as

$$P_r\left\{H_k \geq \underline{H}_k\right\} \geq p_k \tag{7.2.19}$$

where the harvest H_k is a random variable due to the uncertainty induced by the regression equation (7.2.14); p_k is the desired or required reliability. The chance-constraint (7.2.19) must be transformed into an equivalent deterministic form in order to implement the optimization algorithm.

The harvest regression equations are either multiple linear models or transformed linear models after logarithmic transformation of H_k and QS_{jm} depending upon the species of fish. The commercial fish harvest can be written in a linear or nonlinear form depending upon the species (again, using the regressions of the Texas Department of Water Resources, 1980), see Table 7.2.1

$$H_k = (QS)_j^T \bullet \beta_{H_{kj}} \tag{7.2.20}$$

or

$$\ln\left(H_k\right) = \left[\ln(QS)_j\right]^T \bullet \beta_{H_{k\varphi}} \tag{7.2.21}$$

The harvest chance-constraint (7.2.19) is determined using (7.2.20) and (7.2.21), respectively,

7.2.1

Table ~~3.4~~ Regression Equations of Fish Harvest and Freshwater Inflow Relations
(Texas Department of Water Resources, 1980)

Index k for Fish Species	Equations	$\hat{\sigma}_k$	Inflow used in regression equations
1 : All shellfish	$H_1 = 3107.9 - 11.3QS_1 + 7.7QS_2 - 24.2QS_3$	482.8	a *
2 : Spotted seatrout	$\ln(H_2) = 6.8264 - 1.2473 \ln(QS_1) + 1.1526 \ln(QS_2) - 0.40371 \ln(QS_4)$	0.2901	b**
3 : Red drum	$\ln(H_3) = 4.3204 + 0.6937 \ln(QS_2) - 0.8718 \ln(QS_3)$	0.2900	b
4 : All penaeid shrimp	$\ln(H_4) = 1735.8 - 3.7\, QS_1 + 2.7QS_2 - 1.0QS_5$	412.0	c***
5 : Blue crab	$\ln(H_5) = 208.3 + 2.7QS_3 + 0.4QS_4 + 0.5QS_5$	259.5	c

where H_k is the commerical harvest of species k in thousands of pounds,

QS is the mean monthly freshwater inflow during the season (acre/ft):

QS_1 = January - March
QS_2 = April - June
QS_3 = July - August
QS_4 = September - October
QS_5 = November - December

and $\hat{\sigma}_k$ is the standard error.

* using freshwater inflow at the Lavaca Delta
** using freshwater inflow at the Colorado Delta
***using combined freshwater inflows from all contributing rivers and coastal drainage basins

$$P_r\left\{(QS)_j^T \bullet \beta_{H_{kj}} \geq \underline{H}_k\right\} \geq p_k \tag{7.2.22}$$

or

$$P_r\left\{\left[\ln(QS)_j\right]^T \bullet \beta_{H_{kj}} \geq \ln\left(\underline{H}_k\right)\right\} \geq p_k \tag{7.2.23}$$

The deterministic form of equations (7.2.22) and (7.2.23) are, respectively,

$$t_{n-v,\,1-p_k} \bullet \hat{\sigma}_s \sqrt{\left(QS_j\right)^T\left[\left(QSD_j\right)^T \bullet \left(QSD_j\right)\right]^{-1}\left(QS_j\right) + 1} + \left(QS_j\right)^T \hat{\beta}_{H_{kj}} \geq \underline{H}_k \tag{7.2.24}$$

and

$$t_{n-v,\,1-p_k} \bullet \hat{\sigma}_{s_{tj}} \sqrt{\left[\ln\left(QS_j\right)\right]^T\left\{\left[\ln\left(QSD_j\right)\right]^T\left[\ln\left(QSD_j\right)\right]\right\}^{-1}\left[\ln\left(QS_j\right)\right] + 1} + \ln\left(QS_j\right)^T \bullet \hat{\beta}_{H_{kj}} \geq \ln\left(\underline{H}_k\right) \tag{7.2.25}$$

where $t_{n-v,\,1-p_k}$ is the quantile of t - random variable with n-v degrees of freedom and the probability of $1-p_k$, $\hat{\sigma}_{H_k}$ is the estimated standard error associated with the harvest regression equations, QSD_j is a matrix of the observed data of seasonal freshwater inflow used for the harvest regression equations, and

Alternative IV Minimum the total annual freshwater inflow subject to the salinity restriction. This is similar to Alternative I except the minimum seasonal flow (marsh inundation) requirement, (constraint (7.2.11)) being removed.

7.2.4 Chance-Constraint Formulation for Harvest Equation

The regression equations in the optimization model for salinity and harvest are subject to uncertainty due to the variance in the basic data. This uncertainty arises because for the population of observations associated with the sampling process, there is a probability distribution of salinity of commercial harvest for each level of freshwater inflow. ~~Figure 3.1 (b) shows an example of this sampling distribution for the salinity inflow regression equations~~. The basic application of chance-constraints in stochastic programming is to account for the uncertainty of the regression due to random variation in the regression variables by formulating the corresponding constraints into probabilistic form and then transforming them into their deterministic equivalents. (Charnes and Cooper, 1959, 1962, 1963; Charnes and Sterdy, 1966; Jagannathan, 1974; Miller and Wagner, 1965; Sengupta, 1972). In environmental and water resources area, there are a number of papers on water quality models and reservoir design and operation models using chance constraints (Fujiwara et al., 1986; Houck, 1979; Ellis, 1987, Ellis et al., 1985, 1986; Lohani and Thanh, 1978, 1979; Burn and McBean, 1985; Loucks and Dorfman, 1975).

$\ln(QSD_j)$ is a matrix in which each element is logarithmic transformed of the corresponding one in QSD_j.

The chance-constrained model for various alternatives is obtained by using the associated objective along with constraints (7.2.24) and (7.2.25), replacing the respective regression relationships. Derivation of the deterministic equivalent of chance-constraints based on regression equations is shown in Appendix 7.B.

7.3 Problem Solution

7.3.1 Overview

The overall optimization model can be stated in the following general nonlinear programming format using an objective to minimize freshwater inflows or to maximize fishery harvest.

$$\text{Optimize } f(\mathbf{Q}, \mathbf{s}, \mathbf{H}) \tag{7.3.1}$$

subject to the following constraints:

(a) hydrodynamic transport equations that relate salinity, **s**, (vector in spatial and temporal domains) to the freshwater inflow, **Q**,

$$G(\mathbf{Q}, \mathbf{s}) = 0 \tag{7.3.2}$$

where **Q** is a vector of the independent variable (control variable) as a function of time and **s** is a vector of the dependent variable (state variable) as a function of time and location;

(b) regression equations that relate inflow to fishery harvest

$$h(Q, H) = 0 \tag{7.3.3}$$

where **H** is a vector of the fishery harvest for different species;

(c) constraints that define limitations on freshwater inflows due to upstream demands and water uses, and historical ranges

$$\underline{Q} \leq Q \leq \overline{Q} \tag{7.3.4}$$

where **Q** and the limitations are defined as the general terms that they can be interpreted as monthly, seasonal, and annual flows. The marsh inundation requirements are also included in this expression, which are basically lower bound of flows during certain time periods.

(d) constraints that define limitations on salinity.

$$\underline{s} \leq s \leq \overline{s} \tag{7.3.5}$$

The problem posed is a discrete time optimal control problem in which the constraints that relate the state variables (salinities) to the control variables (freshwater inflows) are grouped as a simulator, this is separated from the original constraint set and are solved implicitly. For each iteration in the process of optimization, the optimizer computes the new values of control variables and passes that information to the simulator to update the corresponding state variables. A reduced optimization problem is then formed with a smaller number of decision variables and constraints. The control variables are the freshwater inflows as a function of time. The state variables are the salinities as a function of time and location in the bay and

estuary. During each iteration of the optimizer, a set of control or decision variables, the freshwater inflows for each time period, are sent to the simulator, as shown in Figure 7.3.1. The purpose of the estuarine hydrodynamic transport model is to simulate the flow circulation in the bay system and to be able compute the salinity spatial distribution in the bay for the time period of interest for given freshwater inflow and other boundary conditions . The hydrodynamic transport model then solves for the salinities for each location in the bay and estuary at each time period. Solution of the simulator is performed to evaluate the embedded hydrodynamic transport in the optimization problem. Basically, the state variables (salinities) and the control variables (freshwater inflows) are related through the hydrodynamic transport model. In essence, the simulator equations are used to express the states in terms of the controls yielding a much smaller nonlinear optimization problem.

One of the key elements of the above problem formulation is the relation G(**Q**,**s**) whereby salinity levels in the estuary are defined in terms of a particular sequence of inflows. The hydrodynamic model embedded within this procedure should satisfy the following desired criteria:

(1) The model should be capable of representing an estuarine system with complex circulation, to offer a fair level of complexity in the salinity-inflow relation and therefore in the optimization methodology;

(2) The model should be capable of exhibiting a significantly filtered response to time variations in freshwater inflow, including time

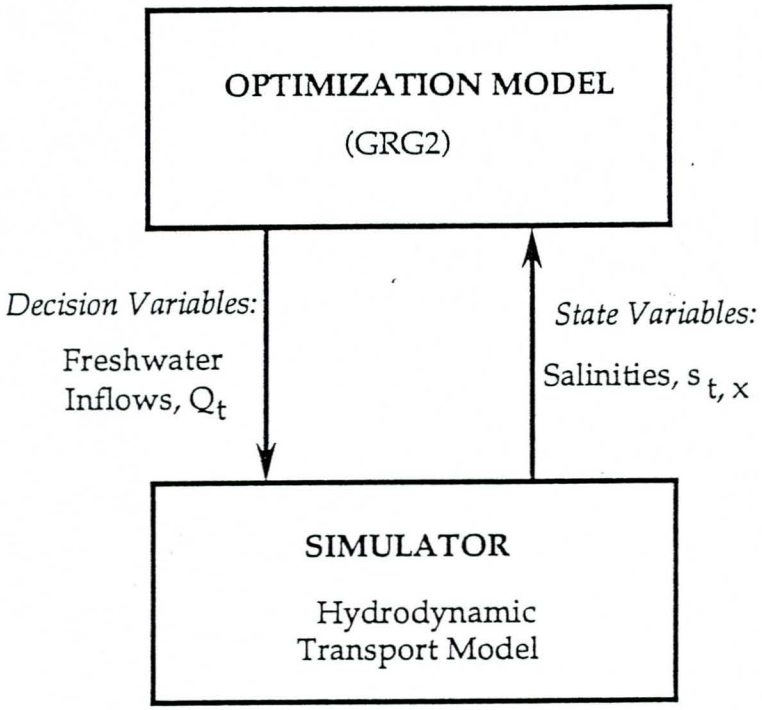


Figure ~~5.1~~ 7.3.1 Optimizer-Simulator Interface

lags and inertia, in order to differentiate the salinity-inflow association from the simple regression forms used in past studies;

(3) The model should be representative of a real estuarine system, so as to allow demonstration of the methodology in a case-study format;

(4) The model should facilitate generalization to a more sophisticated high-resolution estuarine model for detailed applications of the optimization methodology.

In addition to the general requirements for an estuarine hydrodynamic model, the following criteria are considered, in an order of priority, when selecting a simulation model:

(1) The hydrodynamic transport model needs to be called by the optimizer so frequently that the most restrictive requirement for a suitable simulation model is the speed of execution of the code.

(2) The model should be capable of representing an estuarine system with complex circulation, temporal, and spatial salinity variability.

(3) The model should be capable of simulating long term salinity values such as monthly averaged salinity in the bay system.

The above requirements can be met for most applications using a two-dimensional horizontal depth-averaged tidal hydrodynamic transport model, implemented for one of the Texas bays. The computational model to be employed is one of several models currently available. These include the

finite-difference models developed in the Galveston Bay Project (Ward and Espey, 1971), the finite-difference models developed by the Texas Water Development Board for the Texas bays (Texas Department of Water Resources, 1980), the finite-difference model developed by RAND (Leendertse, 1967; Ward and Espey, 1971), and the quasi-2D finite-difference Dynamic Estuary Model (DEM) developed for Sabine Lake estuarine system (Brandes et al., 1975). The available two-dimensional finite-element models tested for selection are FESWMS-2DH (Froehich, 1989), GEVIS (developed by the Notre Dame University in 1990), TXBLEND (Matsumoto, 1992a), and the simplified finite-element model, FETEX, (Matsumoto, 1991).

7.3.2 Reduced Problem

For illustration purpose, the Alternative II, for minimizing the total annual freshwater inflow, is selected to demonstrate the formulation of the optimization problem and solution procedure. The independent (decision) variables are the monthly averaged freshwater inflows from each river connected to the bay system. Thus, even in the original general format (Equation 7.3.1) the objective function is a function of flow vector, **Q**, only. The problem formulated below, however, is still defined as the "reduced" problem for the reasons that (1) it can be viewed as the coefficients associated with **s** (salinity vector) terms in the objective function are set to zero; (2) the size of the optimization problem is dramatically reduced because the G constraints in Equation 7.3.2 are solved implicitly by a separate hydrodynamic transport simulator; and (3) this notation make it more convenient for the description of model formulation and structure hereafter. The reduced problem consists of the "reduced" objective function,

$$\text{Minimize } f(Q, s(Q)) = \text{Min } F(Q) \tag{7.3.6}$$

subject to constraints of harvest (7.3.3), bounds of inflows (7.3.4) and salinity limits (3.3.5).

7.3.3 Solution Procedure

In order to force satisfaction of the salinity bound constraints in the optimizer, these bounds on the state variables (salinities) are incorporated into the objective function using the augmented Lagrangian algorithm. Such an approach not only forces the state bounds to be satisfied, but also reduces the number of constraints. Since only inequality bound-type salinity constraints need to be incorporated, the objective function with the augmented Lagrangian function is derived from Equation ( )

$$\text{Min } L(s(Q), Q, \mu, \sigma) = F(Q) + \frac{1}{2}\sum \sigma_i \left\{ \min\left[o, c_i - \frac{\mu_i}{\sigma_i} \right] \right\}^2 - \frac{1}{2}\sum_i \frac{\mu_i^2}{\sigma_i} \tag{7.3.7}$$

where i is the index for each bound constraint; s_i and m_i are the penalty weights and Lagrangian multipliers for the i-th bound; and c_i is the violation of the bounds either above or below the minimum defined as,

$$c_i = \min\left[s_i - \underline{s}_i, \bar{s}_i - s_i \right] \tag{7.3.8}$$

The reduced optimization problem with augmented Lagrangian terms for minimizing freshwater inflows solved by GRG2 is the objective Equation (7.3.7)

$$\text{Minimize } L(s(Q), Q, \mu, \sigma) \tag{7.3.9}$$

subject to

$$h(Q, s(Q), H) = 0 \tag{7.3.10}$$

$$\underline{Q} \leq Q \leq \overline{Q} \tag{7.3.11}$$

which are, respectively, the constraints on harvest and the bounds on the freshwater inflows. The solution to this reduced problem is a two-step procedure. The overall problem is

$$\min_{\sigma, \mu} \left[\min_{Q \varepsilon S} L(s(Q), Q, \mu, \sigma \right] \tag{7.3.12}$$

where **S** is the set of feasible fresh water inflows as given by Equation (7.3.11). For given values of vectors σ and μ, the reduced problem, Equations (7.3.9), (7.3.10) and (7.3.11), is then solved using a nonlinear optimizer, which is based upon the reduced gradient method. The outer problem is iterated by updating the values of σ and μ for the next solution run of the inner problem. The overall optimization is attained when σ and μ both converge.

The updating formula used for μ is:

$$\mu_i^{(k+1)} = \begin{cases} \mu_i^{(k)} - \sigma_i c_i & \text{if } c_i < \dfrac{\mu_i}{\sigma_i} \\ 0 & \text{otherwise} \end{cases} \tag{7.3.13}$$

where k is the number of the current iteration. The value of **s** is normally adjusted once during early iterations and then kept constant (Powell, 1978).

The overall solution procedure is further illustrated through the flowchart in Figure 7.3.2. There are two loops in this procedure, with the outer loop determining the Lagrangian multipliers (dual variables) and

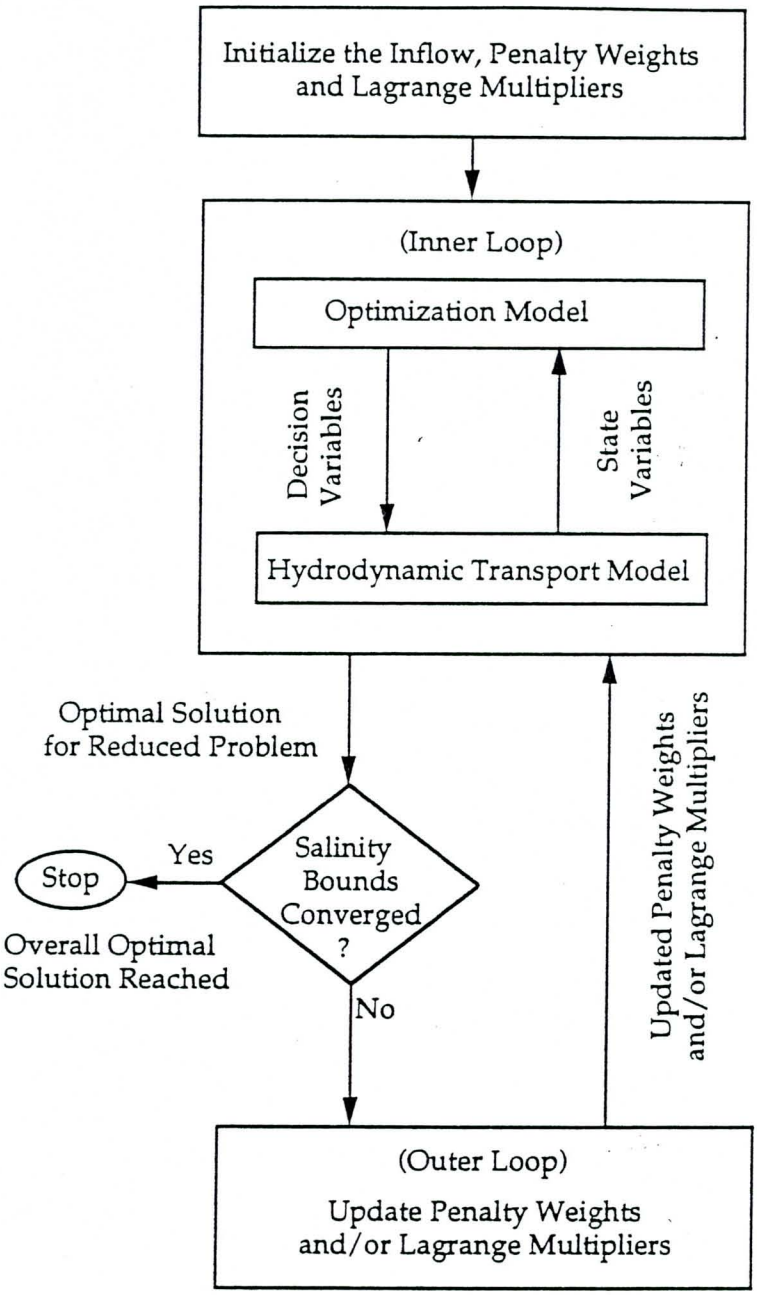


Figure ~~5.3~~ 7.3.2 Overall Solution Procedure

penalty weights. The inner loop solves the reduced augmented Lagrangian problem using an NLP optimizer such as GRG2, whose dual variables and penalty weights are fixed at the values determined by the outer loop. Once an inner loop is finished, the convergence criterion is checked by looking at the size of the salinity bound infeasibility. If it is small enough, the procedure terminates; otherwise, the procedure returns to the outer loop and updates the dual variables and penalty weights and then goes to the inner loop and solves the new reduced augmented Lagrangian again with the updated σ and μ from the outer loop. This process continues until an optimal solution of the overall problem is found.

7.3.4 Computation of Reduced Gradients of AL Problem

The augmented Lagrangian (AL) function (Equation 7.3.7) is a function of flow (Q), salinity (s(Q)), and Lagrangian parameters μ, and σ, which is also expressed as follows:

$$\text{Min } L(s(Q), Q, \mu, \sigma) = f(Q) + \sum_i l_i \left(c_i \left\{ s_i(Q) \right\}, \mu_i, \sigma_i\right) \tag{7.3.14}$$

where

$$\sum_i l_i(s, \mu, \sigma) = \sum_i \begin{cases} -\mu_i c_i(s) + \frac{1}{2}\sigma_i \left[c_i(s) \right]^2, & \text{if } c_i(s) < \frac{\mu_i}{\sigma_i} \\ -\frac{1}{2}\frac{\mu_i}{\sigma_i}, & \text{if } c_i(s) \geq \frac{\mu_i}{\sigma_i} \end{cases} \tag{7.3.15}$$

and the salinity violation vector **c**, is a function of salinity **s**, and the salinity bounds (7.3.8). From Equation (7.3.8), the salinity violation term $c_i(s) = c_i(s_i)$ or $c_i(s) = c_i(\, s_i(Q)\,)$. The gradients of the augmented Lagrangian function can be derived by applying the chain rule:

$$\partial L/\partial Q = \partial f/\partial Q \; + \; S_i \, (\partial l/\partial c_i) \cdot (\partial c_i/\partial s_i) \cdot \partial s_i/\partial Q \qquad (7.3.16)$$

where $\partial l/\partial c_i$ is a function of m_i, and s_i. Hence, $(\partial l/\partial c_i)$ is constant for the inner optimization problem. From Equation (7.3.8), $\partial c_i/\partial s_i$ is either 1 or -1. Thus, the key component for the computation of the (reduced) gradients of the augmented Lagrangian is the partial derivatives of the salinity with respect to the monthly flow, $\partial s/\partial Q$.

The spatial and temporal salinities in the bay system are computed by solving the simulator. The freshwater inflows, Q, are part of the boundary conditions (water - land boundaries) for the hydrodynamic model and the salinity values in the river inlets are part of the boundary conditions for the transport model (source concentration boundaries). In order to compute the matrix $\partial s/\partial Q$ analytically, a new set of simulator equations need to be derived and the analytical solution may be very difficult, if it is not impossible. In this research, the computation of $\partial s/\partial Q$ is carried out by finite-difference methods either the forward differencing or the central differencing. More specifically, $\partial s/\partial Q$ are computed by perturbation Q and running the hydrodynamic transport simulator repeatedly.

7.3.5 CPU Concern

The computation of the reduced gradient is done by the forward difference or the central difference method through calling of the hydrodynamic transport model to simulate the temporal and spatial salinity variability in the nonlinear optimizer. In order to update the objective

function, 12 calls to the hydrodynamic transport simulator are required with each simulating for a period of one month.

Theoretically, if the central difference is used, it requires 24 x 12 x 2 = 576 calls of hydrodynamic transport simulator in order to update the AL reduced gradients, where 24 is the number of decision variables (monthly river flows); 12 is the number of months to be simulated for each variable (Q) to be perturbed to obtain $\partial L/\partial Q$, which is on an annual basis, Equation (7.3.16); and 2 results from the fact that the central difference requires monthly flows to be perturbed at both sides for computation of the AL reduced gradients. Although the number estimated above for the simulation requirement can be reduced by 50% by running the simulation only for the remaining months, 288 calls of the simulations are still extremely expensive in CPU time, for only updating the AL gradients once.

In work done by Bao (1992) the simulation results using HYDSAL (Appendix 7.A) indicate that the impact of a monthly flow perturbation in month t on river j of the salinities in the bay system for the remaining months (t = t+1, t+2, ... 12) is so small that might be mainly affected due to the numerical computation errors (less than 1.0E-8). Therefore the effect of the flow perturbation from previous months is considered as negligible. Hence, the number of hydrodynamic transport simulation calls for updating the AL reduced gradient matrix can be reduced from 576 to 48 for the central difference method for not simulating the salinity in the bay for the remaining months.

Other test run results by Bao (1992) indicate that the difference of the computed AL reduced gradients between the forward and the central

difference methods is insignificant. The forward difference method is sufficient for the purposes of the AL reduced gradient computations. Thus the number of simulation calls to the hydrodynamic transport model can be further reduced to 24.

The test results indicate that over 95% of the CPU time for the model run is required in the hydrodynamic transport model runs for flow and salinity simulations. This is confirmed based on comparison of CPU time requirements of LAV2106 and HYDSAL and estimation of the number of calls of the hydrodynamic transport model. Although this dramatical reduction in the number of hydrodynamic transport simulations (from 576 to 24) will save the CPU time significantly, it is still an extremely intensive computation effort for the whole model. The inner optimization model of GRG2 requires 7 to 60 iterations before the optimal solution is found for the given augmented Lagrangian parameters (initial multiplier, initial penalty, and penalty multiplier). Each iteration may require one or more updates of the reduced gradient and many times for computing the objective functions. The number of the simulation calls is then multiplied by the number of outerloop iterations for updating the augmented Lagrangian parameters and rerun to the inner optimizer.

7.3.6 Gradient Approximation Scheme

The frequent number of simulations require such high CPU time that it is too expensive to run the model. Some innovative modification is needed to reduce the CPU time in simulation. Many considerations and attempts are made to approaches for solving this problem, which are briefly

described below as examples. (1) Increase the grid size from one to two nautical miles to reduce the number of grids. The problem with this approach is that the grid will be too coarse to have reasonable resolution of the simulation results. The water - land boundaries are also very difficult to fit with this grid network not to mention the ship channel. (2) Reduce modeling area from the whole estuary to part of the bay system such as the upper Lavaca bay and part of the Matagorda bay. This is quite reasonable for solving the problem for this application but does not solve the real problem which would limit the model from application in the future. (3) Run the simulations separately and build a data base to establish the relationship between the freshwater inflow and salinities in the bay system. Intuitively, this approach is pragmatic, however, the use of the actual salinity v.s. flow as the entities for the data base might also cause as high an uncertainty as in the cases of salinity regression equations.

The approximation scheme for computing the AL gradients, presented here, is based on the premise that the change of the salinity derivatives with respect to flow is relatively small compared with the flow changes within a certain flow range. In another words, for a set of given flows, the higher order of salinity derivatives (second partial derivatives) are negligible. This assumption is not proven in theory, but the fact that the linearity in the formed transport PDE (second order though) and and the fully explicit time-centered differencing for the nonlinear hydrodynamic PDE's might suggest that the assumption be a close guess.

Figure 7.3.~~2~~3 is a flowchart of the procedure for the approximation scheme for computing the AL gradients and the objective functions. By the finite difference method, the gradients of the AL objective function L(s(Q), Q,

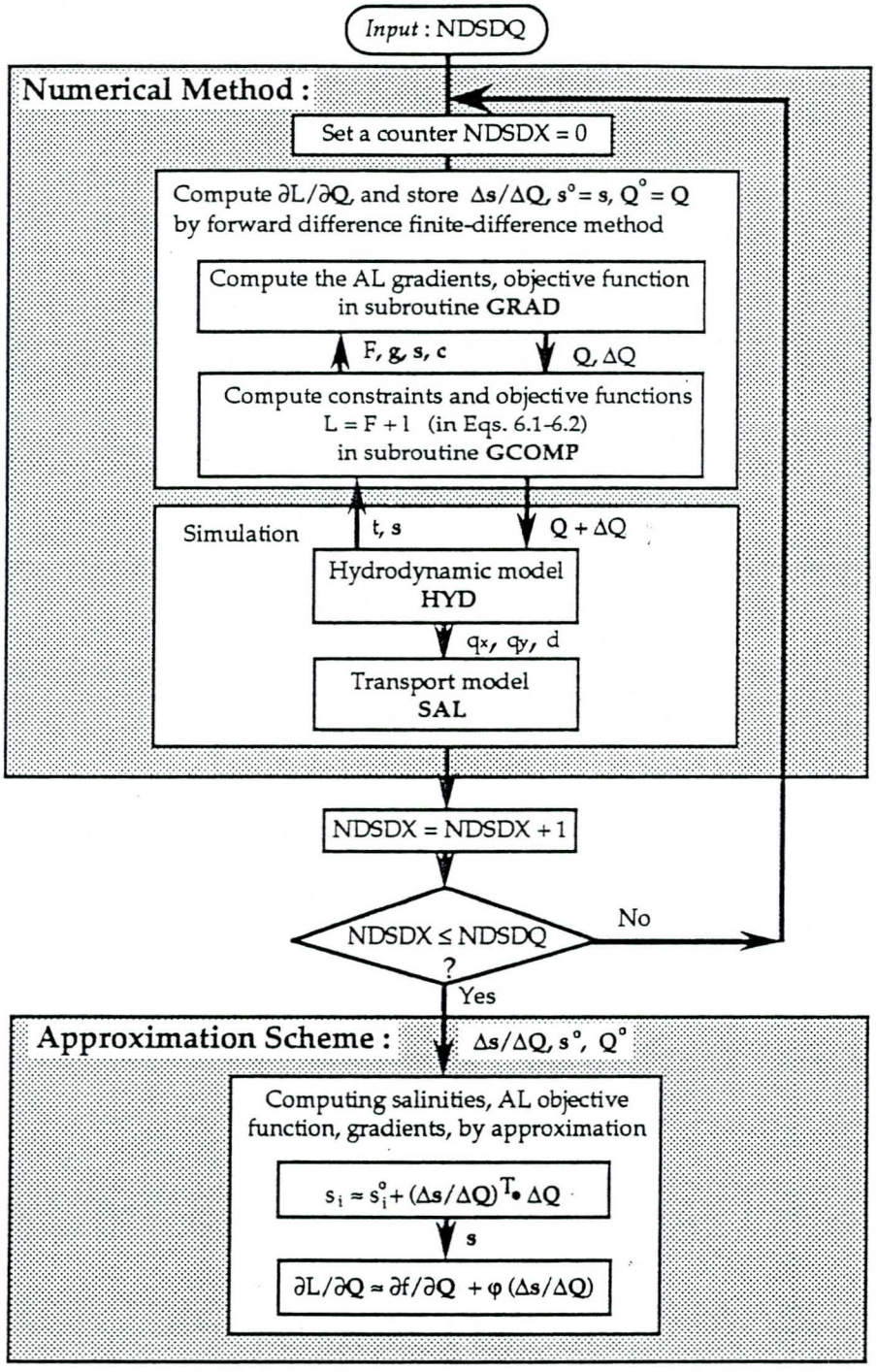


Figure ~~6.2~~ 7.3.3 Flowchart of Approximation Scheme for Computing the Gradients and Objective Function

μ, σ) (Eq. 7.3.14), with respect to monthly inflows, **Q** (**Q** = {$Q_1, Q_2, ..., Q_n$}) is computed by the forward difference method,

$$\frac{\partial L}{\partial Q_m} = \frac{L\Big|_{Q_m + \Delta Q_m} - L\Big|_{Q_m}}{\Delta Q_m} \tag{7.3.17}$$

for variable element Q_m (m = 1, 2, ..., n). The simulator is called to compute the AL terms in the objective, $\sum_i l_i(c_i\{s_i(Q)\}, \mu_i, \sigma$, (Eq. 7.3.14) by simulating the salinities and computing the salinity violation terms. The resultant Jacobian matrix of $\partial s/\partial Q$, and the vectors of **s** and **Q** are stored as Ds/DQ, s^o and Q^o.

The approximation scheme can be described as follows. To compute the new AL gradients for flows of **Q**, the changes of flow from previous evaluation Q^o is simply the difference of the two as

$$DQ = Q^o - Q \tag{7.3.18}$$

The associated salinities, **s**, are computed by

$$s_i \approx s_i^c + (\Delta s/\Delta Q)^T \bullet \Delta Q \tag{7.3.19}$$

and the updated objective function is

$$\begin{aligned} L(s(Q), Q, \mu, \sigma) &= f(Q) + \sum_i l_i\left(c_i\{s_i(Q)\}, \mu_i, \sigma_i\right) \\ &= \sum_m Q_m + \sum_i l_i\{c_i(s_i), \mu_i, \sigma_i\} \end{aligned} \tag{7.3.20}$$

The gradients of the AL objective with respect to the monthly flow are approximated by

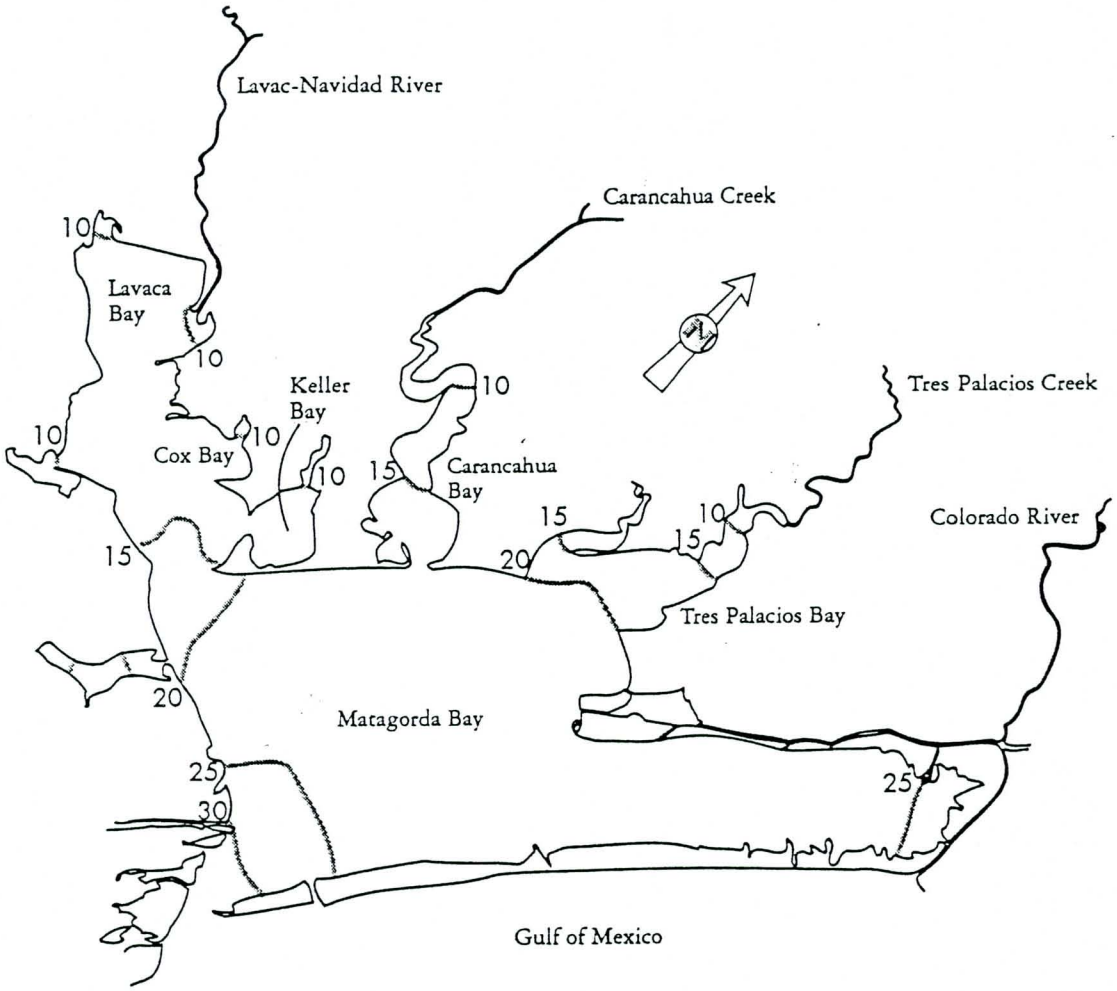
$$\partial L/\partial Q \approx \partial f/\partial Q + j(Ds/DQ) \tag{7.3.21}$$

where j denotes that the derivative of $\partial t/\partial Q$ (Eq. 7.3.16) is a function of the Jacobian matrix of salinities Ds/DQ. Once the computation of $\partial L/\partial Q$ is completed, the s^o and Q^o values are updated by **s** and **Q** in the current iteration.

7.4 Application

Bao (1992) developed a compter code, OPTFLOW, that interfaces GRG2 amd HYD-SAL (Appendix 7.A) for determining the optimal freshwater inflows to bays and estuaries. Appendix 7.C presents a description of the program structure of OPTFLOW. Bao (1992) and Bao and Mays (1992) presented application of the model to the Lavaca-Tres Palacios estuary in Texas, i.e. Matagorda Bay and its secondary (e.g. Lavaca Bay), and tertiary (e.g. Cox Bay) systems, shown in Figure 7.4.1. The major freshwater inflow sources are the Colorado River, which principally affects the eastern segment of Matagorda Bay, and the Lavaca River, which principally influences Lavaca Bay.

The regression equations for fishery harvest (see Table 7.2.1) and the monthly mean salinity bounds are specified for selected locations. For the Matagorda Bay system, these are two types of upper and lower limits on monthly salinity which determine a salinity range. The first type is based on the bounds for viable metabolic and reproductive activity. The second type upper bound selected is the lesser of the historical median monthly salinity level or the first type salinity upper bound, i.e. viability limits (TDWR, 1980).



7A1

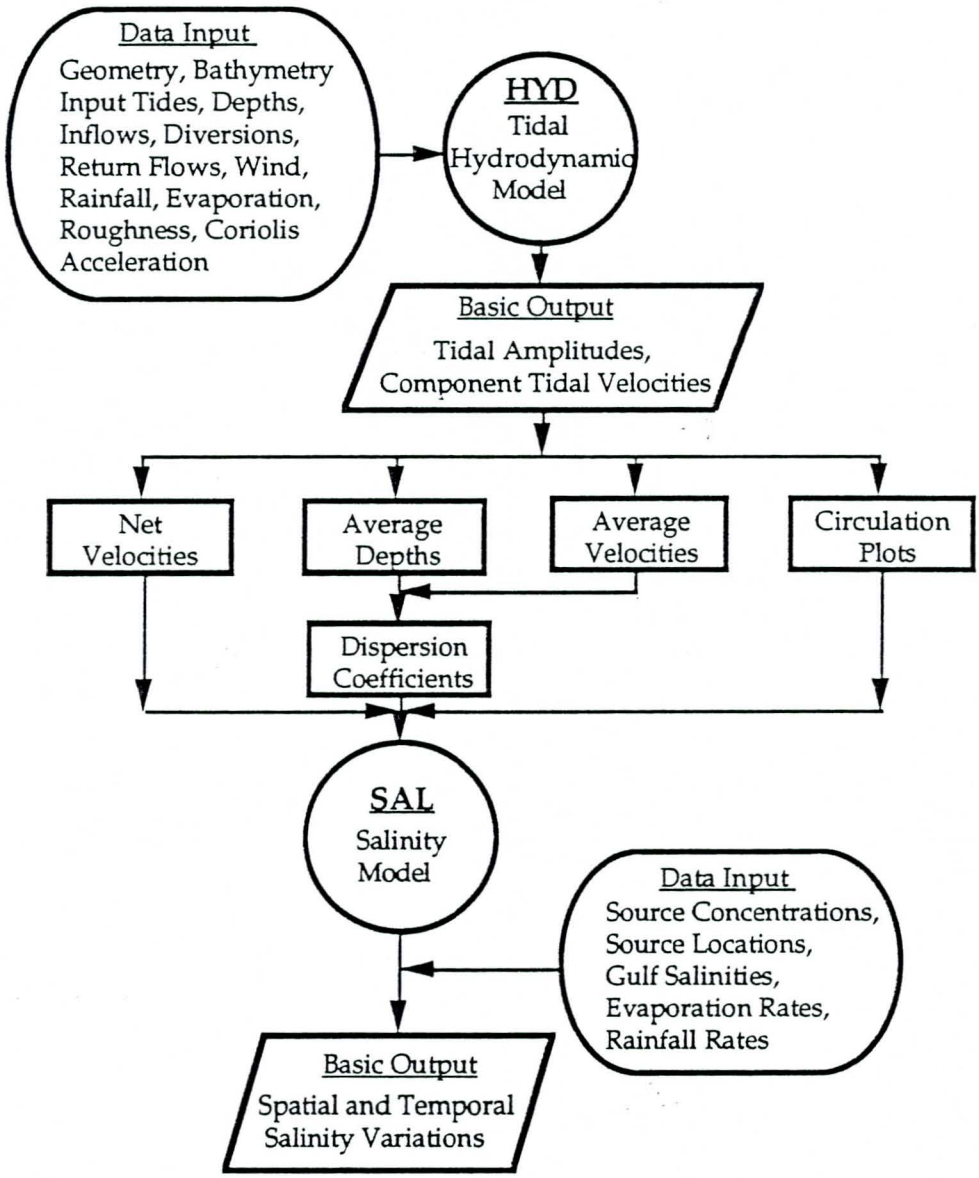
Figure ~~3.4~~ Bay System Map and Example of Simulated Salinity Contour (ppt) in Lavaca-Tres Palacio Estuary

Appendix 7.A

HYD-SAL Simulation Models

The finite difference model HYD-SAL consists of two separate but linked models: a tidal hydrodynamic model (HYD) and a salinity transport model (SAL). The input and output of the models and their linkage are shown in Figure 7.A.1 . Major efforts have been devoted to the development of, and applications of these models (Masch and Associates, 1971; Masch and Brandes, 1971; Texas Department of Water Resources, 1980). HYD and SAL have been applied to four bay systems in Texas, including San Antonio, Matagorda (Lavaca-Tres Palacio), Corpus Christi-Aransas-Copano, and Galveston (Texas Department of Water Resources, 1979, 1980).

The hydrodynamic model (HYD) is developed for vertically well mixed estuaries to solve the two-dimensional dynamic equations of motion and the unsteady continuity equation. These are non-linear partial differential equations to solve for three unknowns of flow flux in x and y directions and depth (or the tidal amplitude). The transport equation for SAL is a linear second order partial differential equation. The fully explicit method is used to solve the hydrodynamic equations. The explicit method used is a time-centered difference scheme involving time stepping of the "leap frog" type for computations of flows and water levels. The alternating direction implicit (ADI) method is used to solve the transport equation, therefore, it is unconditionally stable for any size of time or spatial step. The linear system equations result in a tridiagonal matrix which is efficiently solved using the Thomas algorithm (Masch and Associates, 1971).



7.A.1

Figure ~~5.2~~ Relationship Between Tidal Hydrodynamic and Salinity Models (After: TDWR, 1980)

The hydrodynamic model also incorporates the Coriolis acceleration and wind stress. The four basic types of boundary conditions considered in the hydrodynamic model are as follows:

(1) water - land boundaries,

(2) partial internal boundaries,

(3) artificial ocean boundaries, and

(4) fresh water inflow, diversion, and return flow magnitudes and location.

The salinity model is simplified as the convective-dispersion equation based on the principle of mass conservation. The effect of evaporation and precipitation on salinity is considered in SAL. Similarly, the boundary conditions for the salinity model are listed as;

(1) water - land boundaries,

(2) impermeable internal boundaries, and

(3) source concentration boundaries.

The HYD and SAL models were modified and run as separate models, so that the output of HYD is used as input to SAL, and as a combined model (HYDSAL) to simulate long-term salinity pattern (monthly and annual). For monthly simulation, the CPU time for execution is about 2.5 minutes on a Sun4/390 workstation and 12 seconds on Cray Y-MP8/864.

Appendix 7.B

Derivation of Deterministic Equivalent of

Chance-Constraints

Based on Regression Equations

In order to transform the chance-constraint (7.2.19) into their deterministic equivalent forms, first consider a general multiple linear regression model,

$$Y = \underline{X}^T \underline{\beta} + \varepsilon \tag{7.B.1}$$

where Y is the dependent variable; $\underline{X}$ is a v x 1 column vector of independent variables, $\{1, x_1, x_2, \ldots, x_{v-1}\}^T$; $\underline{\beta}$ is a v x 1 column vector of regression parameters, $\{\beta_0, \beta_1, \beta_2, \ldots, \beta_{v-1}\}^T$; ε is the model error with $E(\varepsilon) = 0$, and $Var(\varepsilon) = \sigma^2$. Because ε is a random variable, the true value of Y and the coefficients of regression equation, $\underline{\beta}$, are never known. Replacing the Y, $\underline{\beta}$ and ε by their estimators, the regression model becomes,

$$\hat{Y} = \underline{X}^T \hat{\underline{\beta}} + \hat{\varepsilon} \tag{7.B.2}$$

For a given set of independent variables, $\underline{x}_0$, the corresponding dependent variable Y_0 can be estimated as,

$$\hat{Y}_0 = \underline{x}_0^T \hat{\underline{\beta}} \tag{7.B.3}$$

with the associated mean

$$E\left(\hat{Y}_0 \,\middle|\, \underline{x}_0\right) = \underline{x}_0^T \hat{\underline{\beta}}$$

and variance

$$\mathrm{Var}\left(\hat{Y}_0 \middle| \underline{x}_0\right) = \sigma^2\left[\underline{x}_0^T\left(X^TX\right)^{-1}\underline{x}_0 + 1\right]$$

where **X** is an n x v matrix of observed data used in developing the regression equations. Replacing the unknown population variance by its estimator, the predicted variance becomes

$$\mathrm{Var}\left(\hat{Y}_0 \middle| \underline{x}_0\right) = \hat{\sigma}^2\left[\underline{x}_0^T\left(X^TX\right)^{-1}\underline{x}_0 + 1\right]$$

Consider a chance-constraint

$$P_r\left\{\underline{Y} \le Y_0\right\} \ge p \tag{7.B.4}$$

by standardizing,

$$P_r\left\{\frac{Y_0 - \hat{E}\left(\hat{Y}_0 \middle| \underline{x}_0\right)}{\sqrt{\mathrm{Var}\left(\hat{Y}_0 \middle| \underline{x}_0\right)}} \ge \frac{\underline{Y} - \underline{x}_0^T\underline{\hat{\beta}}}{\sqrt{\hat{\sigma}^2\left\{\underline{x}_0^T\left(X^TX\right)^{-1}\underline{x}_0 + 1\right\}}}\right\} \ge p$$

which can be rearranged

$$P_r\left\{T_{n-v} \le \frac{\underline{Y} - \underline{x}_0^T\underline{\hat{\beta}}}{\sqrt{\hat{\sigma}^2\left\{\underline{x}_0^T\left(X^TX\right)^{-1}\underline{x}_0 + 1\right\}}}\right\} \le 1 - p \tag{7.B.5}$$

Knowing the reliability p, the standard student distribution deviate can be easily computed. Hence the deterministic equivalent of the chance constraint is

$$\frac{\underline{Y} - \underline{x}_0^T\hat{\underline{\beta}}}{\sqrt{\hat{\sigma}^2\left\{\underline{x}_0^T\left(X^TX\right)^{-1}\underline{x}_0 + 1\right\}}} \le t_{n-v,\,1-p}$$

or

$$t_{n-v,\,1-p}\,\hat{\sigma}\sqrt{\left\{\underline{x}_0^T\left(X^TX\right)^{-1}\underline{x}_0 + 1\right\}} + \underline{x}_0^T\hat{\underline{\beta}} \ge \underline{Y} \qquad (7.B.6)$$

with n-v degree of freedom, and probability of 1-p.

Consider the case that the constraint is bounded on both sides:

$$P_r\left\{\underline{Y} \le Y_0 \le \overline{Y}\right\} \ge p$$

then

$$F_{T,\,n-v}\left[\frac{\overline{Y} - \underline{x}_0^T\hat{\underline{\beta}}}{\hat{\sigma}\sqrt{\left\{\underline{x}_0^T\left(X^TX\right)^{-1}\underline{x}_0 + 1\right\}}}\right] -$$

$$F_{T,\,n-v}\left[\frac{\overline{Y} - \underline{x}_0^T\hat{\underline{\beta}}}{\hat{\sigma}\sqrt{\left\{\underline{x}_0^T\left(X^TX\right)^{-1}\underline{x}_0 + 1\right\}}}\right] \ge p \qquad (7.B.7)$$

However, the explicit expression of the deterministic equivalent of this type of chance-constraint can not be derived. The deterministic equivalents of the commercial harvest constraints can be obtained by substitution of the corresponding variables and parameters into equation (7.B.6). The salinity constraints can be written in the form of equation (7.B.7). The fact that this salinity constraint has only an implicit form must be considered when selecting programming algorithm.

References

Bao, Y.X., Tung, Y.K., Mays, L.W., and Ward, W.H. Jr., "Analysis of the Effect of Freshwater Inflows on Estuary Fishery Resources", Technical Memorandum 89-2, Report to Texas Water Development Board, by the Center for Research in Water Resources, The University of Texas at Austin, Austin, TX, 1989, 49 pp.

Brandes, R.J., Johnson, A.E., Icemena, K.R., and F.D. Masch, "Computer Program Documentation for the Dynamic Estuary Model with Application to Sabine Lake Estuarine System," Final Report to Texas Water Development Board, by Water Resources Engineers, Inc., Austin, Texas, April 1975.

Charnes, A. and Cooper, W.W., "Chance-Constrained Programming," *Management Science*, Vol. 6, p.73-79, 1959.

Charnes, A. and Cooper, W.W., "Chance Constraints and Normal Deviates," *Journal of American Statistics Association*, Vol. 57, p.143-148, 1962.

Charnes, A. and Cooper, W.W., "Deterministic Equivalents for Optimizing and Satisficing Under Chance Constraints," *Operations Research*, Vol. 11, No. 1, p.18-39, 1963.

Charnes, A. and Sterdy, A.C., "A Chance-Constrained Model for Real-Time Control in Research and Development," *Management Science*, Vol. 12, No. 8, p.B-353 to B-362, April 1966.

Ellis, J.H., "Stochastic Water Quality Optimization Using Imbedded Chance Constraints," *Water Resources Research*, Vol. 23, No. 12, p.2227-2238, December 1987.

Ellis, J.H., McBean, E.A., and Farquhar, G.J., "Chance-Constrained/ Stochastic Linear Programming Model for Acid Rain Abatement, 1, Complete Conlinearity and Nonconlinearity," *Atmos. Environ.*, Vol. 19, p.925-937, 1985.

Ellis, J.H., McBean, E.A., and Farquhar, G.J., "Chance-Constrained/ Stochastic Linear Programming Model for Acid Rain Abatement, 2, Limited Conlinearity," *Atmos. Environ.*, Vol. 20, p.501-511, 1986.

Froehich, D.C. 1989. "Finite Element Surface-Water Modeling System: Two-Dimensional Flow in a Horizontal Plane Users Manual," Technical Report, FHWA-RD-88-177, Federal Highway Administration Office of Research, Development, and Technology, 6300 Georgetown Pike, McLean, Virginia 22101, 285 pp.

Fujiwara, O., Gnanendran, S.K., and Ohgaki, S., "River Quality Management Under Stochastic Streamflow," *Journal of Environmental Engineering*, Division of American Society of Civil Engineering, Vol. 112, p.185-198, 1986.

Houck, M.H., "A Chance Constrained Optimization Model for Reservoir Design and Operation," *Water Resources Research*, Vol. 15, No. 5, p.1011-1016, October 1979.

Jagannathan, R., "Chance-Constrained Programming with Joint Constraints," *Operation Research*, Vol. 22, p.358-372, 1974.

Leendertse, J.J., "Aspects of a Computational Model for Long-Period Water-Wave Propagation," The RAND Corp., Santa Monica, CA, 1967a.

Leendertse, J.J., Aspects of a Computational Model for Well-Mixed Estuaries and Coastal Seas, RM 5294 - PR, The Rand Corporation, Santa Monica, CA, 1967b.

Liu, S-K and Leendertse, J.J., "A Three Dimensional Model for Estuaries and Coastal Seas; Vol. VI,". Bristol Bay Simulations, R-2405-NOAA, Rand Corporation, Senta Monica, CA, 1979.

Lohani, B.N. and Thanh, N.C., "Stochastic Programming Model for Water Quality Management in a River," *J. of Water Pollut. Control Fed.*, Vol. 50, p.2175-2182, 1978.

Lohani, B.N. and Thanh, N.C., "Probabilistic Water Quality Control Policies," *Journal of Environmental Engineering,* Division of American Society of Civil Engineering, 5, p.713-725, 1979.

Masch, F.D. and Associates, "Tidal Hydrodynamic and Salinity Models for San Antonio and Matagorda Bays, Texas," A Report to Texas Water Development Board, Austin, Texas, June 1971.

Masch, F. D. and Brandes, R.J., Tidal Hydrodynamic Simulation In Shallow Estuaries, Report to Office of Water Resources Research, U.S. Department of the Interior, Technical Report, HYD 12-7102, Hydraulic Engineering Laboratory, Univ. of Texas, Austin, August 1971.

Matsumoto, J., "Mathematical Description of the FETEX Model Based on a New Computational Method: a Simplified Finite Element Method or a

Generalized Finite Difference Method," Technical Report of the Texas Water Development Board, Austin, Texas, 1991.

Matsumoto, J., "User's Manual for the Texas Water Development Board's Circulation and Salinity Model: TXBLEND," Texas Water Development Board, Austin, Texas, 1992a.

Matsumoto, J., "Guadalupe Estuary Example Analysis," Chapter 8 in Freshwater Inflows to Texas Bays and Estuaries: Ecological Relationships and Methods for Determination of Needs, edited by Longley, W.L., Texas Water Development Board and Texas Parks and Wildlife Department, Austin, TX. 1992b.

Miller, B.L. and Wagner, H.M., "Chance-Constrained Programming with Joint Constraints," *Operation Research*, Vol. 13, p.930-945, 1965.

Sengupta, J.K., "Chance-Constrained Linear Programming with Chi-Square Type Deviates," *Management Science*, Vol. 19, p.337-349, 1972.

Texas Department of Water Resources, "Mathematical Simulation Capabilities in Water Resource Systems Analysis," Report LP-16, Austin, Texas. 1979.

Texas Department of Water Resources, "Lavaca-Tres Palacios Estuary: A Study of the Influence of Freshwater Flows," Report LP-106, Austin, Texas. 1980.

Tung, Y.K, Bao, Y.X., Mays, L.W., and Ward, W.H. "Optimization of Freshwater Inflow to Estuaries", *Journal of Water Resources Planning and Management*, Vol. 116, No. 4, July/August 1990.

Valiela, I. and Teal, J.M., "Nutrient Limitation in Salt Marsh Vegetation," in Ecology of Halophytes, edited by Reimold, R.J. and Queen, W.H., p 547-563, Academic Press, N. Y., 1974.

Ward, G.H. and Espey, W.H., Estuarine Modeling: An Assessment, EPA 16070 DZV, Government Printing Office, Washington, D.C., 1971.

CHAPTER 8

OPTIMAL CONTROL BY FEEDBACK CONTROL METHODS

8.1 Dynamic Programming

Dynamic programming (DP) transforms a sequential or multistage decision problem that may contain many interrelated decision variables into a series of single-stage problems, each containing only one or a few variables. In other words, the dynamic programming technique decomposes an N--decision problem into a sequence of N separate, but interrelated, single-decision subproblems. Decomposition is very useful in solving large, complex problems by decomposing a problem into a series of smaller subproblems and then combining the solutions of the smaller problems to obtain the solution of the entire model composition. The reason for using decomposition is to solve a problem more efficiently which can lead to significant computational savings. As a rule of thumb, computations increase exponentially with the number of variables, but only linearly with the number of subproblems.

Dynamic programming can overcome the shortcomings of an exhaustive enumeration procedure using the following concepts.

1. The problem is decomposed into subproblems and the optimal alternative is selected for each subproblem so that it is never necessary to enumerate all combinations of the problem in advance.

2. Because optimization is applied to each subproblem, nonoptimal combinations are automatically eliminated.

3. The subproblems should be linked together in a special way so that it is never possible to optimize over infeasible combinations.

Referring to Fig. 8.1.1, the basic elements and terminologies in a dynamic programming formulation are introduced as follows:

1. **Stages** *(n)* are the points of the problem where decisions are to be made. In the funds allocation example, each projector represents a stage in the dynamic programming model. If a decision making problem can be decomposed into N subproblems, there will be N stages in the dynamic programming formulation.

2. **Decision Variables** (d_n) are courses of action to be taken for each stage. The decision in the project funding example is the alternative within the project to be selected. The number of decision variables, d_n, in each stage is not necessarily equal to one.

3. **State Variables** (S_n) are variables describing the state of a system at any stage n. A state variable can be discrete or continuous, finite or infinite. Referring to Fig. 8.1.1, at any stage n, there are input states, S_n, and output states, S_{n+1}. The state variables of the system in a dynamic programming model have the function of linking succeeding stages so that, when each stage is optimized separately, the resulting decision is automatically feasible for the entire problem. Furthermore, it allows one to make optimal decisions for the remaining stages without having to check the effect of future decisions for decisions previously made.

4. **State Return** (r_n) is a scalar measure of the effectiveness of decision making in each stage. It is a function of the input state, the output

state, and the decision variables of a particular stage, that is $r_n = r\,(S_n, S_{n+1}, d_n)$.

5. **Stage Transformation or State Transition** (t_n) is a single-valued transformation which expresses the relationships between the input state, the output state, and the decision. In general, through the stage transformation, the output state at any stage n can be expressed as the function of the input state and the decision as

$$S_{n+1} = t_n\,(S_n, d_n) \tag{8.1.1}$$

The basic features that characterize all dynamic programming problems are as follows:

1. The problem is divided into stages, with decision variables at each stage.

2. Each stage has a number of states associated with it.

3. The effect of the decision at each stage is to produce return, based on the stage return function, and to transform the current state variable into the state variable for the next stage, through the state transform function.

4. Given the current state, an optimal policy for the remaining stages is independent of the policy adopted in previous stages. This is called **Bellman's principle of optimality**, which serves as the backbone of dynamic programming.

5 The solution begins by finding the optimal decision for each possible state in the last stage (called the **backward recursive**) or

in the first stage (called the **forward recursive**). A forward algorithm computationally advances from the first to the last stage whereas a backward algorithm advances from the last stage to the first.

6. A recursive relationship that identifies the optimal policy for each state at any stage n can be developed, given the optimal policy for each state at the next stage, $n + 1$. This backward recursive equation, referring to Fig. 8.1.1, can be written as

$$f^*_n(S_n) = \underset{d_n}{\text{opt.}} \{r_n(S_n, d_n) \circ f^*_{n+1}(S_{n+1})\}$$

$$= \underset{d_n}{\text{opt.}} \{r_n(S_n, d_n) \circ f^*_{n+1}[t_n(S_n, d_n)]\} \qquad (8.1.2)$$

where o represents an algebraic operator which can be +, -, x, or whichever is appropriate to the problem. The recursive equation for a forward algorithm is stated as

$$f^*_n(S_n) = \underset{d_n}{\text{opt.}} \{r_n(S_n, d_n) \circ f^*_{n-1}(S_{n-1})\} \qquad (8.1.3)$$

The recursive equation for the backward dynamic programming technique can be written as

$$f^*_n(S_n) = \begin{cases} \underset{d_n}{\text{opt.}} \left[r_n(S_n, d_n)\right], & \text{for } n = N \\ \underset{d_n}{\text{opt.}} \left[r_n(S_n, d_n) \, o \, f^*_{n+1}(S_{n+1})\right] & \text{for } n = 1 \text{ to } N-1 \end{cases} \qquad (8.1.4a), (8.4.1b)$$

Although dynamic programming possessess several advantages in solving water resources problems, especially for those involving the analysis of multistage processes, it has two disadvantages, that is, the computer memory

and time requirements. These disadvantages could become especially severe under two situations: (1) when the number of state variables is large; and (2) when the dynamic programming is applied in a discrete fashion to a continuous state space. The problem associated with the latter case is that there exist difficulties in obtaining the true optimal solution without a considerable increase in discretization of state space. With the advancement in computer technology those disadvantages are becoming less and less severe.

An increase in the number of discretizations and/or state variables would geometrically increase the number of evaluations of the recursive formula and core-memory requirement per stage. This problem of rapid growth of computer time and core-memory requirement associated with multiple-state variable dynamic programming problems is referred to as curse of dimensionality. From the problem-solving viewpoint, the problem of increased computer time is of much less concern than that of the increased computer storage requirement. Therefore, the rapid growth in memory requirements associated with multiple-state variable problems can make the difference between solvable and unsolvable problems.

8.2 Feedback Method of Optimal Control for Linear Systems

The general optimal control problem for hydrosystems is stated as follows

$$\text{Optimize } f(\mathbf{x}_t, \mathbf{u}_t, t) \tag{8.2.1}$$

subject to the state equation

$$\mathbf{x}_t = g_t(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{u}_t) \qquad t=1,\ldots T \tag{8.2.2}$$

Many hydrosystems optimal control problems can be formulated to minimize the sum of the squared deviations of a state variable from a specified target of the state variable, subject to the state equation. These problems constitute a linear deterministic control problem which consist of minimizing a quadratic loss function measuring the preference subject to the state equation that defines the dynamics of the system given as

$$\underset{\mathbf{u}_t}{\text{Minimize}}\ Z = \text{Min}\sum_{t=1}^{T}(x_t-r_t)^T P_t(x_t-r_t) \tag{8.2.3}$$

subject to the state equation

$$\mathbf{x}_t = A_t\mathbf{x}_{t-1} + C_t\mathbf{u}_t + \mathbf{b}_t \tag{8.2.4}$$

in which $\mathbf{x}_t$ is the n-dimensional state variable vector, where $\mathbf{x}_0$ is the vector of initial (known) state variables; $\mathbf{r}_t$ is an n-dimensional vector of target values for the state variable at time t; P_t is an n x n positive semi-definite penalty matrix for deviating from target $\mathbf{r}_t$ at time t; A_t is an n x n matrix of known elements; C_t is an n x n matrix of known elements; m is the number of control variables; $\mathbf{b}_t$ is an n x 1 vector of known constants; $\mathbf{u}_t$ is an m-dimensional vector of control variables. The above problem defined by equations (8.2.3) and (8.2.4) is a linear-quadratic optimal control problem.

The optimal solution to the above optimal control problem is the time sequence of the control variables, $\mathbf{u}_t$, t=1,....T, which is the decision variable. The feedback method to solve these optimal control problems is a dynamic programming approach consisting of a stage-by-stage optimization of the objective function subject to the system state equation. The control solves the above optimal control problem by deriving a set of feedback rules from a set of recursive equations (Chow 1981). Dynamic programming requires that the

objective function be separable in order to perform the stage-by-stage optimization. The quadratic objective function satisfies the requirement of separability.

The loss for period T, conditioned upon information up to T-1, which is a function of $\mathbf{u}_T$,

$$\psi_T = \left(\mathbf{x}_T - \mathbf{r}_T\right)^T P_T \left(\mathbf{x}_T - \mathbf{r}_T\right) \tag{8.2.5a}$$

$$= \left(\mathbf{x}_T^T H_T \mathbf{x}_T - 2\mathbf{x}_T^T \mathbf{h}_T + \mathbf{c}_T\right) \tag{8.2.5b}$$

where

$$P_T = H_T \tag{8.2.6}$$

$$P_T \mathbf{r}_T = \mathbf{h}_T \tag{8.2.7}$$

$$\mathbf{c}_T = \mathbf{r}_T^T P_T \mathbf{r}_T \tag{8.2.8}$$

in which H_T is an nxn matrix at time T and $\mathbf{h}_T$ is an n-dimensional vector at time T. Substituting $A_T \mathbf{r}_{T-1} + C_T \mathbf{u}_T + \mathbf{b}_T$ for $\mathbf{x}_T$ in (8.2.5) and minimizing with respect to $\mathbf{u}_T$ by differentiation results in

$$\hat{\mathbf{u}}_T = V_T \mathbf{x}_{T-1} + \mathbf{w}_T \tag{8.2.9}$$

where

$$V_T = -\left(C_T^T H_T C_T\right)^{-1} \left(C_T^T H_T A_T\right) \tag{8.2.10}$$

$$\mathbf{w}_T = -\left(C_T^T H_T C_T\right)^{-1} C_T^T \left(H_T \mathbf{b}_T - \mathbf{h}_T\right) \tag{8.2.11}$$

The minimum expected loss for the last period is obtained by substiuting for $\mathbf{u}_T$ in ψ_T

$$\hat{\psi}_T = \mathbf{x}_{T-1}^T \left(A_T + C_T V_T\right)^T H_T \left(A_T + C_T V_T\right) \mathbf{x}_{T-1}$$

$$+ 2\,\mathbf{x}_{T-1}^T \left(A_T + C_T V_T\right)^T \left(H_T \mathbf{b}_T - \mathbf{h}_T\right)$$

$$+ \left(\mathbf{b}_T + C_T \mathbf{w}_T\right)^T H_T \left(\mathbf{b}_T + C_T \mathbf{w}_T\right)$$

$$- 2 \left(\mathbf{b}_T + C_T \mathbf{w}_T\right)^T \mathbf{h}_T + \mathbf{c}_T \tag{8.2.12}$$

To obtain the optimal for the last two periods, consider that $\hat{\mathbf{u}}_T$ has been computed that would yield the minimum $\hat{\psi}_T$ and that by the principle of optimality of dynamic programing $\mathbf{u}_{T-1}$ is needed to minimize

$$\psi_{T-1} = \left[\left(\mathbf{x}_{T-1} - \mathbf{r}_{T-1}\right)^T P_{T-1} \left(\mathbf{x}_{T-1} - \mathbf{r}_{T-1}\right) + \hat{\psi}_T\right] \tag{8.2.13a}$$

$$= \left[\mathbf{x}_{T-1} H_{T-1} \mathbf{x}_{T-1}^T - 2\,\mathbf{x}_{T-1}^T \mathbf{h}_{T-1} + \mathbf{c}_{T-1}\right] \tag{8.2.13b}$$

where the expression (8.2.12) for $\hat{\psi}_T$ has been defined and

$$H_{T-1} = P_{T-1} + \left(A_T + C_T V_T\right)^T H_T \left(A_T + C_T V_T\right) \tag{8.2.14}$$

$$\mathbf{h}_{T-1} = P_{T-1} \mathbf{r}_{T-1} - \left(A_T + C_T V_T\right)^T \left(H_T \mathbf{b}_T - \mathbf{h}_T\right) \tag{8.2.15}$$

$$\mathbf{c}_{T-1} = \mathbf{r}_{T-1}^T P_{T-1} \mathbf{r}_{T-1} + \left(\mathbf{b}_T + C_T \mathbf{w}_T\right)^T H_T \left(\mathbf{b}_T + C_T \mathbf{w}_T\right)$$

$$- 2 \left(\mathbf{b}_T + C_T \mathbf{w}_T\right)^T \mathbf{h}_T + \mathbf{c}_T \tag{8.2.16}$$

Because equation (8.2.13b) is identical to (8.2.5b) with T replaced by T-1, the solution for $\hat{\mathbf{u}}_{T-1}$ is identical with (8.2.9) with T replaced by T-1 where V_{T-1} and $\mathbf{w}_{T-1}$ are defined by (8.2.10) and (8.2.11) respectively with a similar change in time subscripts. Accordingly, $\hat{\psi}_{T-1}$ is given by (8.2.12) with the subscripts T replaced by T-1. When solving the problem for the last three periods, $\hat{\mathbf{u}}_T$ and $\hat{\mathbf{u}}_{T-1}$ have been found that would yield the minimum expected loss $\hat{\psi}_{T-1}$ for the last two periods.

By the principle of optimality we only need to minimize

$$\psi_{T-2} = \left[\left(\mathbf{x}_{T-2} - \mathbf{r}_{T-2} \right)^T P_{T-2} \left(\mathbf{x}_{T-2} - \mathbf{r}_{T-2} \right) + \hat{\psi}_{T-1} \right] \tag{8.2.17}$$

with respect to $\mathbf{u}_{T-2}$ and so forth. At the end of this process $\hat{\mathbf{u}}$, is determined from $\hat{\mathbf{u}}_1 = V_1 \mathbf{x}_0 + \mathbf{w}_1$, as the optimal policy for the first period and the associated minimum loss $\hat{\psi}_1$ for all periods (or from period 1 forward). Computationally solve (8.2.10) and (8.2.14) with t replacing T for V_t and H_t backward in time, for t = T, T-1, . . . , 1. Then solve (8.2.11) and (8.2.15) with t replacing T and $\mathbf{w}_t$ and $\mathbf{h}_t$ backward in time, for t = T, T-1, . . . , 1. Finally solution of (8.2.16) with t replacing T backward in time yields **c**, which is used to evaluate $\hat{\psi}_1$ given by (8.2.12) with 1 replacing T.

The expression $\hat{\psi}_T$ given by (8.2.12) can be used to obtain the values (shadow prices or dual variables) of the initial resources $\mathbf{x}_{t-1}$. The vector of dual variables (shadow prices) is the derivation of $-\hat{\psi}_t$ (negative loss or benefits) with respect to x_{t-1}, namely,

$$\frac{\partial \hat{\psi}_t}{\partial \mathbf{x}_{t-1}} = -2\left(A_t C_t V_t\right)^T \left[H_t \left(A_t + C_t V_t \right) \mathbf{x}_{t-1} + H_t \mathbf{b}_t - \mathbf{h}_t \right] \tag{8.2.18}$$

The algorithm is summarized below.

1. Initialization of the recursive equations are

$$H_t = P_t \tag{8.2.19}$$

$$\mathbf{x}_t = P_t \mathbf{r}_t \tag{8.2.20}$$

where H_t is an n x n matrix at time t and $\mathbf{x}_t$ is an n-dimensional vector at time t.

2 The following recursive equations are solved backwards in time from the terminal period t=T,...,1 to derive the feedback coefficients V_t and

$$V_t = -(C_t^T H_t C_t)^{-1}(C_t H_t A_t) \tag{8.2.21}$$

$$\mathbf{w}_t = -(C_t^T H_t C_t)^{-1} C^T (H_t \mathbf{b}_t - \mathbf{x}_t) \tag{8.2.22}$$

$$H_{t-1} = P_{t-1} + (A_t + C_t V_t)^T H_t (A_t + C_t V_t) \tag{8.2.23}$$

$$\mathbf{x}_t = P_{t-1} \mathbf{r}_{t-1} - (A_t + C_t V_t)^T (H_t \mathbf{b}_t - \mathbf{x}_t) \tag{8.2.24}$$

where V_t is an m x n feedback coefficient matrix at time t and $\mathbf{w}_t$ is an m-dimensional feedback vector at time t.

3. After the feedback coefficients, V_t and $\mathbf{w}_t$ have been computed, the optimal control variable, $\mathbf{u}_t$ can be computed using the following feedback rule

$$\mathbf{u}_t = V_t \mathbf{x}_{t-1} + \mathbf{w}_t \tag{8.2.25}$$

and the state equation

$$x_t = A_t x_t + C_t w_t + b_t \tag{8.2.26}$$

8.3 Groundwater Management Problems

Makinde-Odusola and Marino(1989) employed the feedback method of optimal control (discussed in section 8.2) to model groundwater hydraulic management problems. The generalized mathematical formulation of the two-dimensional groundwater hydraulic management problem defines the state variable vector $\mathbf{x}_t$ as the piezometric head vector $\mathbf{h}_t$, the control vector, $\mathbf{u}_t$, as the vector of pumping rates $\mathbf{q}_t$ so that the problem is

$$\text{Minimize(or maximize)} \quad f(\mathbf{h}_t, \mathbf{q}_t, t) \tag{8.3.1}$$

subject to

$$\mathbf{h}_t = \mathbf{g}_t(\mathbf{h}_{t-1}, \mathbf{q}_t) \qquad t = 1,\ldots,T \tag{8.3.2}$$

The feedback method of control described in section (8.2) applied to the groundwater management problem would constitute a linear deterministic control problem of the form

$$\underset{\mathbf{q}_t}{\text{Min}}\, Z = \sum_{t=1}^{T} (\mathbf{h}_t - \mathbf{r}_t)^T P_t (\mathbf{h}_t - \mathbf{r}_t) \tag{8.3.3}$$

subject to

$$\mathbf{h}_t = A_t \mathbf{h}_{t-1} + C_t \mathbf{q}_t + \mathbf{b}_t \tag{8.3.4}$$

where $\mathbf{r}_t$ is an n-dimensional vector of targets for the state variable; P_t is an nxn positive semi-definite penalty matrix for deviating from target $\mathbf{r}_t$ at time t; A_t and C_t matrices with known elements; m is the number of control variables; and n is the number of state variables.

The feedback rule is initialized as follows

$$H_t = P_t \tag{8.3.5}$$

$$\mathbf{h}_t = P_t \mathbf{r}_t \tag{8.3.6}$$

Feedback coefficients V_t and $\mathbf{w}_t$ are derived by solving the recursive equations (?) backwards in time for $\mathbf{h}_t = \mathbf{x}_t$. Once the feedback rule coefficients have been computed for all time periods, the optimal pumping strategy, $\mathbf{q}_t$, is determined using the following feedback rule and the initial piezometric head vector, $\mathbf{h}_0$

$$\mathbf{q}_t = G_t \mathbf{h}_{t-1} + \mathbf{g}_t \tag{8.3.7}$$

Many groundwater management problems can be formulated as equations (?) and (?), so that the feedback method of control can be used for the solution procedure. The state variable is the vector of piezometric heads $\mathbf{h}_t$ at all simulation nodes; the control variable is the vector of pumping (or recharge rates) $\mathbf{q}_t$; and the target state vector, $\mathbf{r}_t$, is the optimized piezometric head. If the hydraulic management problem is to estimate the rate of recharge to the aquifer, the target state vector, $\mathbf{r}_t$, should be set equal to the historical (or estimated) piezometric head (Makinde-Odusola and Marino, 1989). Yazdanian and Peralta(1986) discussed other methods for obtaining the vector of target piezometric heads, $\mathbf{r}_t$.

Makinde-Odusola and Marino (1989) solved the groundwater flow equation (state equation) numerically using SUTRA (Voss, 1984), which is a model for simulating two-dimensional saturated-unsaturated fluid density-dependent groundwater flow. This model also solves the energy or reactive adsorptive single-species solute transport. A two dimensional finite element

scheme for the spatial discretization of the governing groundwater flow equation and a finite difference scheme is used for the temporal discretization of the groundwater flow equation.

Some advantages of the feedback control method include (Makinde-Odusola and Marino, 1989): (1) the ease in specification and interpretation of the objective function parameters; (2) stochasticity in either the parameters or in the state variable can be handled; (3) the objective function form exploits the duality between control and parameter estimation; (4) the objective function can be used to impose physical hydraulic constraints; and (5) the incorporation of operational experience in specification of the vector of target piezometric heads.

8.4 Feedback Method of Optimal Control for Nonlinear Systems

Consider an optimal control problem with a quadratic objective function of the form

$$\text{Minimize } Z = \text{Min} \sum_{t=1}^{T} \left(x_t - r_t\right)^T P_t \left(x_t - r_t\right) \tag{8.4.1}$$

subject to the u_t nonlinear state equation of the form

$$x_t = g\left(x_t, x_{t-1}, u_t, \eta_t\right) + \varepsilon_t \tag{8.4.2}$$

where x_t is the vector of state variables; r_t is a vector of specified target values; η_t is the vector of parameters that are not subject to control; and ε_t is a vector of random disturbances with mean zero, variance σ and is distributed independently through time. For purposes of the following discussion the elements of η_t are given leaving ε_t as the only random variable.

The quadratic objective function can be expressed as

$$Z = \sum_{t=1}^{T} (x_t^T P_t x_t - 2x_t^T P_t r_t + r_t^T P_t r_t) \tag{8.4.3}$$

The problem is to minimize the expectation of Z so that using DP the optimal control problem for the last period T is to minimize

$$\psi_T = E_{T-1} (x_T^T P_T x_T - 2x_T^T P_T r_T + r_T^T P_T r_T)$$

$$= E_{T-1} (x_T^T H_T x_T - 2x_T^T h_T + C_T) \tag{8.4.4}$$

with respect to x_T, where

$$H_T = P_T \tag{8.4.5}$$

$$h_T = P_T r_T \tag{8.4.6}$$

$$C_T = r^T{}_T P_T r_T \tag{8.4.7}$$

The following steps are required to solve the optimal control problem for period T:

<u>Step 1</u>

Start with a nominal policy (control) $\tilde{u}_T$ and set ε_T equal to zero; then linearize (8.4.2) about $x_{T-1} = x_{T-1}^o$ (given); $x_T = x_T^*$ and $u_T = \tilde{u}_T$, so that the solution of the system is

$$x_T^* = g(x_T^*, x_{T-1}^o, \tilde{u}_T, \eta_T) \tag{8.4.8}$$

where u_T^* is solved by an iterative method such as Gauss - Siedel. The linearized version of (8.4.2) is

$$x_T = x_T^* + B_{1T}(x_T - x_T^*) + B_{2T}(x_{T-1} - x_{T-1}^*) + B_{3T}(u_T - \tilde{u}_T) + \varepsilon_T \tag{8.4.9}$$

The j-th column of B_{1T} consists of the partial derivatives of the vector function **g** with respect to the j-th element of x_T^* evaluated at the given values x_T^*, x_{T-1}^o, $\tilde{u}_T$ and η_T. Similar is true for the j-th column of B_{2T}, and B_{3T}.

Step 2

Equation (8.4.9) can be rearranged (solved) to obtain the linearized approximation

$$x_T = A_T x_{T-1} + C_T u_T + b_T + y_T \tag{8.4.10}$$

where

$$\left(A_T C_T y_T\right) = \left(I - B_{1T}\right)^{-1}\left(B_{2T} B_{3T} \varepsilon_T\right) \tag{8.4.11a}$$

$$b_T = x_T^* - A_T x_{T-1}^o - C_T \tilde{u}_T \tag{8.4.11b}$$

and y_T is a random vector that is serially independent and identically distributed.

The matrix $I\text{-}B_{IT}$ is

$$I\text{-}B_{1T} = \begin{bmatrix} I\text{-}B_{1T}^* & 0 \\ 0 & I \end{bmatrix} \tag{8.4.12}$$

in which the order of B^*_{1T} is the number of simultaneous state equations excluding the identities.

Step 3

Equation (8.4.10) is minimized with respect to $\mathbf{u}_T$ subject to equation (8.4.10) by differentiating (8.4.10) with respect to $\mathbf{u}_T$ and interchanging the order of taking expectation and differentiation

$$\frac{\partial \mathbf{y}_T}{\partial \mathbf{u}_T} = 2\left[\left(\frac{\partial \mathbf{x}_T^T}{\partial \mathbf{u}_T}\right) H_T \mathbf{x}_T - \left(\frac{\partial \mathbf{x}_T^T}{\partial \mathbf{u}_T}\right) \mathbf{h}_T\right]$$

$$= 2\left[C_T^T H_T\left(A_T \mathbf{x}_{T-1} + C_T \mathbf{u}_T + \mathbf{b}_T + \mathbf{y}_T\right) - C_T^T \mathbf{h}_T\right] = 0 \tag{8.4.13}$$

where (8.4.10) has been substituted for $\mathbf{x}_T$ and used to compute $\frac{\partial \mathbf{x}_T^T}{\partial \mathbf{u}_T}$.

Equation (8.4.13) is solved for $\mathbf{u}_T$

$$\hat{\mathbf{u}}_T = V_T \mathbf{x}_{T-1} + \mathbf{w}_T \tag{8.4.14}$$

where

$$V_T = -\left(E_{T-1} C_T^T H_T C_T\right)^{-1}\left(E_{T-1} C_T^T H_T A_T\right) \tag{8.4.15}$$

$$\mathbf{w}_T = -\left(E_{T-1} C_T^T H_T C_T\right)^{-1}\left(E_{T-1} C_T^T H_T \mathbf{b}_T - E_{T-1} C_T^T \mathbf{h}_T\right) \tag{8.4.16}$$

In the linear approximation (8.4.10) A_T, C_T, and $\mathbf{b}_T$ are not functions of ε_T and as a result are not random. The expectations signs in (8.4.15) and (8.4.16) can be dropped.

Step 4

The solution from (8.4.14), $\hat{\mathbf{u}}_T$ replaces the initial guess $\tilde{\mathbf{u}}_T$ in step 1; then repeat steps 1 through 4 until there is convergence in $\tilde{\mathbf{u}}_T$. Even when convergence occurs the solution is not truly optimal because the approximate form (8.4.10) is used with constant coefficients A_T, C_T, and $\mathbf{b}_T$.

Step 5

Using (8.4.10) for $\mathbf{x}_T$ and (8.4.14) for $\mathbf{u}_T$, the minimum objective for period T is from (8.4.4),

$$\begin{aligned}\hat{\psi}_T = {} & \mathbf{x}_{T-1}^T E_{T-1}(A_T + C_T V_T)^T H_T (A_T + C_T V_T)\mathbf{x}_{T-1} \\ & + 2\,\mathbf{x}_{T-1}^T E_{T-1}(A_T + C_T V_T)^T (H_T \mathbf{b}_T - \mathbf{h}_T) \\ & + E_{T-1}(\mathbf{b}_T + C_T \mathbf{w}_T)^T H_T (\mathbf{b}_T + C_T \mathbf{w}_T) \\ & + E_{T-1}\mathbf{y}^T H_T \mathbf{y} - 2E_{T-1}(\mathbf{b}_T + C_T \mathbf{w}_T)^T \mathbf{h}_T \\ & + E_{T-1} C_T \end{aligned} \tag{8.4.17}$$

Applying the principle of optimality in dynamic programming, minimizing with respect to $\mathbf{u}_{T-1}$.

$$\begin{aligned}\psi_{T-1} &= E_{T-2}(\mathbf{x}_{T-1}^T P_{T-1}\mathbf{x}_{T-1} - 2\mathbf{x}_{T-1}^T P_{T-1}\mathbf{r}_{T-1} + \mathbf{r}_{T-1}^T P_{T-1}\mathbf{r}_{T-1}^T + \hat{\psi}_T) \\ &= E_{T-2}(\mathbf{x}_{T-1}^T H_{T-1}\mathbf{x}_{T-1} - 2\mathbf{x}_{T-1}^T \mathbf{h}_{T-1} + c_{T-1})\end{aligned} \tag{8.4.18}$$

Substituting (8.4.17) for $\hat{\psi}_T$

$$\begin{aligned}H_{T-1} &= P_{T-1} + E_{T-1}(A_T + C_T V_T)^T H_T (A_T + C_T V_T) \\ &= P_{T-1} + E_{T-1}(A_T^T H_T A_T) + V_T^T (E_{T-1} C_T^T H_T A_T)\end{aligned} \tag{8.4.19}$$

$$\begin{aligned}h_{T-1} &= P_{T-1} + \mathbf{r}_{T-1} + E_{T-1}\left(A_T + C_T V_T\right)^T\left(\mathbf{h}_T - H_T \mathbf{b}_T\right) \\ &= P_{T-1} + \mathbf{r}_{T-1} + E_{T-1}\left(A_T + C_T V_T\right)^T \mathbf{h}_T - E_{T-1}\left(A_T^T H_T \mathbf{b}_T\right) \\ &\quad - V_T^T\left(E_{T-1} C_T^T H_T \mathbf{b}_T\right)\end{aligned}$$

$$c_{T-1} = E_{T-1}\left(\mathbf{b}_T + C_T\mathbf{w}_T\right)^T H_T\left(\mathbf{b}_T + C_T\mathbf{w}_T\right) - 2E_{T-1}\left(\mathbf{b}_T + C_T\mathbf{w}_T\right)^T \mathbf{h}_T$$

$$+ \mathbf{r}_{T-1}^T P_{T-1}\mathbf{r}_{T-1} + E_{T-1}\mathbf{y}_T^T H_T \mathbf{y}_T + E_{T-1}c_T$$

The second line of (8.4.18) has the same forms as (8.4.4); therefore the steps in the solution for $\mathbf{u}_T$ with T-1 replacing T, yield an optimal solution $\hat{\mathbf{u}}_{T-1}$ in the form (8.4.14) and the corresponding minimum two-period loss, $\hat{\psi}_{T-1}$ from (8.4.14) the process continues backward in time until , $\hat{\mathbf{u}}_1$ and $\hat{\psi}_1$ are obtained.

REFERENCES

R. Bellman and S. Dreyfus, "Applied Dynamic Programming," Princeton University Press, Princeton, New Jersey, 1962.

P. Dyer and S. McReynolds, "The Computational Theory of Optimal Control," Academic, New York, 1970.

S. H. Gershwin and D. H. Jacobson, "A Discrete-Time Differential Dynamic Programming Algorithm with Application to Optimal Orbit Transfer," AIAA J. 8, 1616-1626 (1970).

D. Jacobson and D. Mayne, "Differential Dynamic Programming," Elsevier, New York, 1970.

D. Mayne, "A Second Order Gradient Method for Determining Optimal Trajectories for Nonlinear Discrete-Time Systems," Int. J. Control 3, 85-95 (1966).

D. M. Murray, and S. Yakowitz, "The Application of Optimal Control Methodology to Nonlinear Programming Problems," Math. Programming, 21, 331-347 (1981).

K. Ohno, "Differential Dynamic Programming and Separable Programs," J Optim. Theory Appl. 24, 617-637 (1978)

S. J. Yakowitz and B. Rutherford, "Computational Aspects of Discrete-Time Optimal Control," Appl. Math. Comput. 15, 29-45 (1984).

S. Yakowitz, "A Statistical Foundation for Machine Learning, with Applications to Go-Moku," Comput. & Math. with Appl. 17, 1989.

CHAPTER 9 DIFFERENTIAL DYNAMIC PROGRAMMING

9.1 Differential Dynamic Programming Algorithm

9.1.1

This section is concerned with the differential dynamic programming method for discrete - time optimal control problems. The term differential dynamic programming (DDP) used by Jacobson and Mayne (1970) broadly refer to stagewise nonlinear programming procedures. Earlier works that basically developed DDP procedures for unconstrained discrete-time control problems include Bellman and Dreyfus (1962), Mayne (1966), Gershwin and Jacobson (1970), Dyer and Mc Reynolds (1970), Jacobson and Mayne (1970), Yakowitz and Rutherford (1984), and Yakowitz (1989), Ohno (1978) and Murray and Yakowitz (1981) are contributed to DDP techniques for constrained optimal control problem.

Yakowitz and Rutherford (1984) summarized the following ;

> "Our opinion is that a little - known technique called 'differential dynamic programming' offers the potential of enormously expanding the scale of discrete - time optimal control problems which are subject to numerical solution. Among the attractive features of this method are that no discretization of control or state space is used; the memory requirements grow as m^2 and the computational requirements as m^3, with m being the dimension of the control variable; the successive approximation converges globally under lenient smoothness assumption and the convergence is quadratic if certain convexity assumption hold"

The objective of DDP is to minimize a quadratic approximation instead of solving the actual control problem. Yakowitz and Rutherford (1984) pointed out the following properties of DDP for unconstrained problems :

(a) DDP overcomes the curse of dimensionality (computational burden and memory requirements grow exponentially with state and control dimensions) in that the computational requirements grow as m^3N and memory requirements as mnN, where n and m are, respectively, the state and control variable dimensions, and N is the number of decision times;

(b) Under lenient condition, DDP is globally convergant;

(c) no discretization of state or control spaces is required;

(d) the convergant rule of the DDP algorithm is quadratic for control problems in which the Hessian matrix of the objective function is convex in a neighborhood of the solution.

The basic optimal control problem considered here is stated as follows

state control

$$\underset{\mathbf{u}}{\text{Min}}\ Z = \sum_{t=1}^{T} f_t(\mathbf{x}_t, \mathbf{u}_t, t) \tag{9.1.1}$$

subject to

$$\mathbf{x}_{t+1} = \mathbf{g}_t(\mathbf{x}_t, \mathbf{u}_t, t) \qquad t = 1, \ldots, T \tag{9.1.2}$$

9.1.2 Algorithm Definition

Define the current or known control policy as $\bar{\mathbf{u}}_t$ for t = 1, . . . , T and the current or known state trajectory or $\bar{\mathbf{x}}_t$ for t = 1, . . . , t+1. The initial state is x_1. For

end time

any function W(x, u) defined by the control and state variables, let QP(W(x, u)) denote the linear and quadratic path of the Taylor's series expansion of W( ) about $(\bar{u}, \bar{x})$. The quadratic for time T where the DDP backward recursion begins is

$$L(x, u, T) = QP(f(x, u, T))$$

$$= \frac{1}{2}\delta x^T\left(\frac{\partial^2 f}{\partial x^2}\right)\delta x + \delta x\left(\frac{\partial^2 f}{\partial x \partial u}\right)\partial u$$

$$= \frac{1}{2}\delta u^T\left(\frac{\partial^2 f}{\partial u^2}\right)\delta u + \left(\frac{\partial f}{\partial u}\right)\delta u + \frac{\partial f}{\partial x}\delta x \tag{9.1.3}$$

where $\delta x = (x - \bar{x}_T)$ and $\delta u = (u - \bar{u}_T)$ are state and input perturbations and the gradients and Hessian of f(x, u, T) are evaluated at $\bar{x}_T$ and $\bar{u}_T$. Equation (8.3.3) can be presented in a more compact form as

what are A, B, C, D + E ?
See p-9-5

$$L(x, u, T) = \delta x^T A_T \delta x + \delta x^T B_T \delta u + \delta u^T C_T \delta u + D_T^T \delta u + E_T^T \delta x \tag{9.1.4}$$

The idea of DDP is to minimize the quadratic approximation instead of the actual control problem value function, thereby obtaining a computer amenable function which is at the expense of involving truncation error. A necessary condition that a control u* minimize L(x, u, T) is

$$\nabla_u L(x, u, T)^T = 2C_T \delta u + B_T \delta x + D_T \tag{9.1.5}$$

Using the quadratic approximation then the optimal control u* can be found from Eq. (8.3.5). Under the assumption the C_T is non singular

$$\delta u(x, T) = (u^* - \bar{u}_T) \tag{9.1.6a}$$

$$= -\left(\frac{1}{2}\right) C_T^{-1}\left(D_T + B_T \delta x\right) \tag{9.1.6b}$$

$$= \alpha_T + \beta_T \delta x \tag{9.1.6c}$$

where $\alpha_T = \left(-\frac{1}{2}\right) C_T^{-1} D_T$ and $\beta_T = \left(-\frac{1}{2}\right) C_T^{-1} B_T$

The optimal value function is

$$F(x, T) = \min_{u} Z = \min f\,(x_T, u_T, T) \tag{9.1.7}$$

which is approximated by the quadratic as

$$V(x; T) = L(x, u\,(x, T), T)$$

$$= L(x_1 \bar{u}_T + (\alpha_T + \beta_T\, \delta x), T) \tag{9.1.8}$$

V(x; T) is quadratic because

$$V(x; T) = \delta x^T P_T \delta x + Q_T \delta x \tag{9.1.9}$$

where

$$P_T = A_T - \left(\frac{1}{4}\right) B_T^T C_T^{-1} B^T \tag{9.1.10}$$

$$Q_T^T = -\left(\frac{1}{2}\right) D_T^T C_T^{-1} B_T + E_T \tag{9.1.11}$$

as long as C_T is nonsingular

DDP backward recursion procedure is performed for t = T, T - 1, . . . , 1 using the quadratic

$$L(\mathbf{x}, \mathbf{u}, t) = QP\,[\,f(\mathbf{x}, \mathbf{u}, t) + V(\mathbf{g}(\mathbf{x}, \mathbf{u}, t)); t + 1] \tag{9.1.12}$$

where $V(\mathbf{x}, t + 1)$ is the quadratic approximate optimal return function defined as

$$V(\mathbf{x}, t + 1) = \delta\mathbf{x}^T P_{t+1} \delta\mathbf{x} + Q_{t-1} \delta\mathbf{x} \tag{9.1.13}$$

Similar to Equation ( ) the quadratic can be expressed as

$$L(\mathbf{x}, \mathbf{u}, T) = \delta\mathbf{x}^T A_t \delta\mathbf{x} + \delta\mathbf{u}^T B_t \delta\mathbf{x} + \delta\mathbf{u}^T C_t \delta\mathbf{u} + D_t^T \delta\mathbf{u} + E_t^T \delta\mathbf{x} \tag{9.1.14}$$

Using calculus it can be shown that, the coefficients A_t, B_t^T, C_t, D_t^T and E_t^T can be written as

L. Mayne could not come up with the derivation of these, Carlos has it.

$$A_t = \tfrac{1}{2}\left(\frac{\partial^2 f}{\partial \mathbf{x}^2}\right)_t + \left(\frac{\partial g}{\partial \mathbf{x}}\right)_t^T P_{t+1} \left(\frac{\partial g}{\partial \mathbf{x}}\right)_t + \tfrac{1}{2} \sum_{i=1}^{n} (Q_{t+1})_i \left(\frac{\partial^2 g}{\partial \mathbf{x}_i^2}\right) \tag{9.1.15}$$

$$B_t^T = \left(\frac{\partial^2 f}{\partial \mathbf{x} \partial \mathbf{u}}\right)_t + 2\left(\frac{\partial g}{\partial \mathbf{x}}\right)_t^T P_{t+1} \left(\frac{\partial g}{\partial \mathbf{x}}\right)_t + \frac{1}{2} \sum_{i=1}^{n} \left(Q_{t+1}\right)_i \left(\frac{\partial^2 g}{\partial \mathbf{x}_i \partial \mathbf{u}_i}\right) \tag{9.1.16}$$

$$C_t = \tfrac{1}{2}\left(\frac{\partial^2 f}{\partial \mathbf{u}^2}\right)_t + \left(\frac{\partial g}{\partial \mathbf{x}}\right)_t^T P_{t+1} \left(\frac{\partial g}{\partial \mathbf{x}}\right)_t + \tfrac{1}{2} \sum_{i=1}^{n} (Q_{t+1})_i \left(\frac{\partial^2 g}{\partial \mathbf{u}_i^2}\right) \tag{9.1.17}$$

$$D_t^T = \left(\frac{\partial f}{\partial \mathbf{u}}\right)_t + Q_{t+1} \left(\frac{\partial g}{\partial \mathbf{x}}\right)_t \tag{9.1.18}$$

$$E_t^T = \left(\frac{\partial f}{\partial \mathbf{x}}\right)_t + Q_{t+1} \left(\frac{\partial g}{\partial \mathbf{x}}\right)_t \tag{9.1.19}$$

The first order derivatives of $f(\mathbf{x}, \mathbf{u}, t)$ in the above equations are components of the gradient of $f(\mathbf{x}, \mathbf{u}, t)$; the second order derivatives are the components of the Hessian of $f(\mathbf{x}, \mathbf{u}, t)$; the first order derivatives of $g(\mathbf{x}, \mathbf{u}, t)$ are components of the Jacobian of $g(\mathbf{x}, \mathbf{u}, t)$; and the second order derivatives of

$g(x, u, t)_i$ for $1 \le i \le n$ are the blocks of the Hessian matrices of the coordinates of $g(x, u, i)$. All derivatives are evaluated about the current states and controls.

The first - order necessary condition for optimality is

$$\nabla_u L(x, u, t) = 0 \tag{9.1.20}$$

so that the minimizing strategy for the quadratic $L(x, u, t)$ is

$$\partial u(x; t) = \alpha_t + \beta_t(x - \bar{x}_t) \tag{9.1.21}$$

where $$\alpha_t = -\tfrac{1}{2}C_t^{-1}D_t^1 \tag{9.1.22}$$

and $$\beta_t = -\tfrac{1}{2}C_t^{-1}B_t \tag{9.1.23}$$

The approximating polynomial for the optimal return function is

$$\begin{aligned} V(x; t) &= L(x, u(x, t), t) \\ &= (\partial x)^T P_t(x - \bar{x}_t) + Q_t' \partial x \end{aligned} \tag{9.1.24}$$

where

$$P_t = A_t - \frac{1}{4}B_t^T C_t^{-1} B_t \tag{9.1.25}$$

$$Q_t = -\frac{1}{2}D_t^T C_t^{-1} E_t^T \tag{9.1.26}$$

These are equations necessary for the DDP backward recursion. α_t and β_t for $(1 \le t \le N)$ must be stored for use in the forward sweep. The forward sweep determines the successor DDP policy by successively selecting controls according to the rule $u(x^*, t)$ and then calculating the successor state at each time so that $u_1^* = u(x_1; 1)$ and $x_2^* = g(x_1^*, u_1^*, 1)$. Then the following is for $t = 2, \ldots, T$

$$\mathbf{u}_t^* = \mathbf{u}(\mathbf{x}_1^*; 1) + \overline{\mathbf{u}}_1 \tag{9.1.27}$$

and

$$\mathbf{x}_{t+1}^* = g(\mathbf{x}_t^*, \mathbf{u}_t^*, t) \tag{9.1.28}$$

For the next DDP iteration the DDP sucessor control is the current control sequence $\overline{\mathbf{u}}$.

9.1.3 Algorithm Description

Input for the DDP procedure consist of T, the number of decision times; m, the dimension of the control variable; n, the dimension of the state variable; $\overline{\mathbf{u}}$, a nominal policy; components of the gradients, $\frac{\partial f}{\partial \mathbf{x}}$ and $\frac{\partial f}{\partial \mathbf{u}}$; components of the Hessian, $\frac{\partial^2 f}{\partial \mathbf{x}^2}$, $\frac{\partial^2 f}{\partial \mathbf{u}^2}$, and $\frac{\partial^2 f}{\partial \mathbf{x} \partial \mathbf{u}}$; the Jacobian of g, $\frac{\partial g}{\partial \mathbf{x}}$ and $\frac{\partial g}{\partial \mathbf{u}}$; and the blocks of the Hessian matrices of the coordinates of g, $\frac{\partial^2 g}{\partial \mathbf{x}_i^2}$, $\frac{\partial^2 g}{\partial \mathbf{u}_i^2}$ and $\frac{\partial^2 g}{\partial \mathbf{x}_i \partial \mathbf{u}_i}$.

The program parameters are A_t, B_t, C_t, D_t, E_t for $1 \le t \le T$, which are the coeffecients of $L(\mathbf{x}, \mathbf{u}, t)$; P_t, Q_t for $2 \le t \le T+1$; α_t, β_t for $1 \le t \le N$ which are the coeffecients of the linear strategy function $\mathbf{u}(\mathbf{x}, t)$; and θ_t for $1 \le t \le N + 1$, which are the parameters for the acceptance tests. The steps of the algorithm are outlined below.

Step 0: Select an initial (nominal) policy.

$\overline{\mathbf{u}}_t$, $t = 1,\ldots,N$

Step 1: Initialize parameters and compute loss and trajectory for the given policy.

From Class

Set $P_{n+1} = 0_{nxn}$ $\quad P_{T+1} = 0_{nxn}$

$Q_{n+1} = 0_{1xn}$ $\quad Q_{T+1} = 0_{1xN}$

$q_{n+1} = 0$ $\quad \theta_{T+1} = 0$

$\bar{x}_1$, is given and fixed

given initial state; given initial control policy

$\bar{x}_{t+1} = g(\bar{x}_t, \bar{u}_t, t)$ state eqns (simulator)

$$Z(\bar{u}) = \sum_{t=1}^{N} f_t\left(\bar{x}_t, \bar{u}_t, t\right)$$ objective function (cost function)

Step 2: Backward sweep (Perform the following (a) - (d) for t = ~~1, . . . ,~~ N) $T \rightarrow 1$

(a.) Compute A_t, B_t, and C_t

$$A_t = \frac{1}{2}\left(\frac{\partial^2 f}{\partial x^2}\right)_t + \left(\frac{\partial g}{\partial x}\right)_t^T P_{t+1}\left(\frac{\partial g}{\partial x}\right)_t + \frac{1}{2}\sum_{i=1}^{n}\left(Q_{t+1}\right)_i\left(\frac{\partial^2 g}{\partial x_i^2}\right) \qquad (9.1.29)$$

$$B_t^T = \left(\frac{\partial^2 f}{\partial x \partial u}\right)_t + 2\left(\frac{\partial g}{\partial x}\right)_t^T P_{t+1}\left(\frac{\partial g}{\partial x}\right)_t + \sum_{i=1}^{n}\left(Q_{t+1}\right)_i\left(\frac{\partial^2 g}{\partial x_i \partial u_i}\right) \qquad (9.1.30)$$

$$C_t = \frac{1}{2}\left(\frac{\partial^2 f}{\partial u^2}\right)_t + \left(\frac{\partial g}{\partial x}\right)_t^T P_{t+1}\left(\frac{\partial g}{\partial x}\right)_t + \frac{1}{2}\sum_{i=1}^{n}\left(Q_{t+1}\right)_i\left(\frac{\partial^2 g}{\partial u_i^2}\right) \qquad (9.1.31)$$

(b.) Compute D_t^T and E_t^T

$$D_t^T = \left(\frac{\partial f}{\partial u}\right)_t + Q_{t+1}\left(\frac{\partial g}{\partial u}\right)_t \tag{9.1.32}$$

$$E_t^T = \left(\frac{\partial f}{\partial x}\right)_t + Q_{t+1}\left(\frac{\partial g}{\partial x}\right)_t \tag{9.1.33}$$

(c.) Compute P_t and Q_t

$$P_t = A_t - \frac{1}{4} B_t^T C_t^{-1} B_t \tag{9.1.34}$$

$$Q_t = -\frac{1}{2} D_t^T C_t^{-1} B_t + E_t \tag{9.1.35}$$

(d.) Compute α_t and β_t

$$\alpha_t = -\frac{1}{2} C_t^{-1} D_t \tag{9.1.36}$$

$$\beta_t = -\frac{1}{2} C_t^{-1} B_t \tag{9.1.37}$$

(e.) Compute θ_t

$$\theta_t = -\frac{1}{2} D_t^T C_t^{-1} D_t + \theta_{t+1} \tag{9.1.38}$$

In the above steps (a.) - (e.) store P_t, Q_t, θ_t and replacing P_{t+1}, Q_{t+1}, θ_{t+1}, respectively. Also store α_t and β_t.

Step 3: Forward sweep.

If $\theta_1 \leq q_{min}$ then is the optimal policy. Otherwise compute the updated control policy.

(a.) set $\varepsilon = 1.0$

(b.) compute $u_t(\varepsilon)$ recursively for $t = 1, \ldots, T$ using

$$u_t(\varepsilon) = \varepsilon\alpha_t + \beta_t\left(x_t - \bar{x}_t\right) + \bar{u}_t \tag{9.1.39}$$

$$x_{t+1} = g\left(x_t, u_t(\varepsilon), t\right) \tag{9.1.40}$$

(c.) compute $(u(\varepsilon))$

$Z(u(\varepsilon))$

$$Z(\bar{u}) = \sum_{t=1}^{n} f_t\left(x_t, u_t(\varepsilon), t\right) \tag{9.1.41}$$

$Z(u(\varepsilon))$

(d.) If $Z(u) - Z(\bar{u}) \leq \varepsilon\left(\frac{\theta_1}{2}\right)$, set $\bar{u}_t = u_t(\varepsilon)$ for t=1,...,T and go to step 1

Otherwise perform a line search; set $\varepsilon = \frac{\varepsilon}{2}$ and go to step 3 (b.)

9.3 Multi-Reservoir Operation

Murray and Yakowitz (1979) applied constrained DDP to muti-reservoir control problems. Consider the four reservoir systems shown in Figure 9.3.1. Let $r_{i,t}$ denot the release of the t-th reservoir during decision time t, $s_{i,t}$ denotes the begining storage, and $q_{i,t}$ denotes the inflow to the i-th reservoir. The basic reservoir mass balance equation is

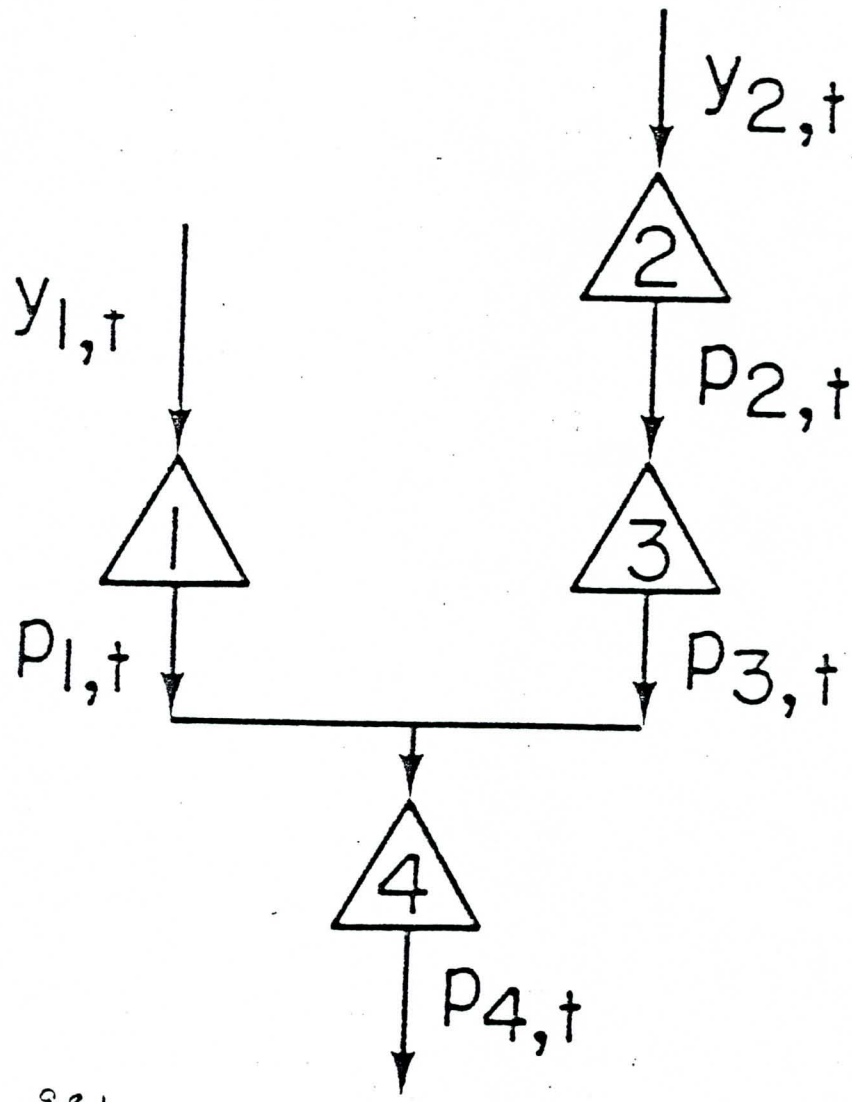


Fig. ~~1~~ 9.3.1. Four-reservoir configuration.

$$\mathbf{s}_{t+1} = \mathbf{s}_t + \mathbf{q}_t + M\,\mathbf{r}_t \tag{9.3.1}$$

where $\mathbf{x}_t = (x_{1,t}, \ldots, x_{4,t})^T$, $\mathbf{s}_t = (s_{1,t}, \ldots, s_{4,t})^T$, $\mathbf{r}_t = (r_{1,t}, \ldots, r_{4,t})^T$, and M is a fourth-order matrix with -1's on the diagonal and +1's in the position (k, j) is reservoir k releases into reservoir j and zero elsewhere. The mass balance equation for the four-reservoir problem is

$$\begin{bmatrix} s_{1,t+1} \\ s_{2,t+1} \\ s_{3,t+1} \\ s_{4,t+1} \end{bmatrix} = \begin{bmatrix} s_{1,t} \\ s_{2,t} \\ s_{3,t} \\ s_{4,t} \end{bmatrix} + \begin{bmatrix} q_{1,t} \\ q_{2,t} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} r_{1,t} \\ r_{2,t} \\ r_{3,t} \\ r_{4,t} \end{bmatrix} \tag{9.3.2}$$

Reservoir storage bound constraints are defined as

$$\begin{aligned} 0 \le s_{1,t} \le 10 \\ 0 \le s_{2,t} \le 10 \\ 0 \le s_{3,t} \le 10 \\ 0 \le s_{4,t} \le 10 \end{aligned} \tag{9.3.3}$$

The initial state is $\mathbf{s}_1 = (5, 5, 5, 5)^T$ and the terminal state is $\mathbf{s}_{13} = (5, 5, 5, 7)^T$ for 12 time periods. Inflows are $s_{1,t} = 2$ and $s_{2,t} = 3$ for $t = 1, \ldots, 12$. Release constraints are

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \le \begin{bmatrix} r_{1,t} \\ r_{2,t} \\ r_{3,t} \\ r_{4,t} \end{bmatrix} \le \begin{bmatrix} 3 \\ 3 \\ 4 \\ 7 \end{bmatrix} \qquad 1 \le t \le 12 \tag{9.3.4}$$

The loss function to be minimized is

$$F(\mathbf{r}) = \sum_{t=1}^{12} f\left(\mathbf{s}_t, \mathbf{r}_t, t\right) \tag{9.3.5}$$

where

$$f\left(\mathbf{s}_t, \mathbf{r}_t, t\right) = \sum_{j=1}^{4} c_j, t, r_{j,t} \tag{9.3.6}$$

The loss coefficients are given in Table 9.3.1.

The terminal constraint

$$\mathbf{s}_{13} = (5, 5, 5, 7)^T$$

can be enforced by translating this condition into constraints on the preceding state and control, e.g. for $s_{1,13} = 5$ the mass balance is]

$$r_{1,12} = 5 - s_{1,12} - q_{1,12} = 3 - s_{1,12}$$

The release $r_{1,12}$ is bounded by $0 \leq r_{1,12} \leq 3$ so that for the terminal state to be readable then

$$3 \leq s_{1,12} \leq 6$$

this in turn requires bounds on the control $r_{1,11}$ and this sets allowable ranges for $s_{1,11}$. One could then construct sequences $\left\{\eta_{1,t}\right\}$ and $\left\{\xi_{1,t}\right\}$ so that for any t, $\eta_{1,t} < s_{1,t} < \xi_{1,t}$. In general form this can be written as

$$\eta_t < \mathbf{s}_t < \xi_t \tag{9.3.7}$$

Equation (9.3.7) also is assumed to satisfy the state constraints (9.3.3) which may be rewritten in the form

$$\eta_t < s_{t-1} + q_{t-1} + Mr_{t-1} < \xi_t \tag{9.3.8}$$

These linear constraints can be rewritten as

$$\eta^c_{t-1} \leq r_{t-1} \leq \xi^c_{t-1} \tag{9.3.9}$$

where η^c_{t-1} and ξ^c_{t-1} are constructed from η_t, ξ_t, and $M^{-1}(s_{t-1} + q_{t-1})$. The problem would then be to require that (9.3.4) and (9.3.9) be satisfied simultaneously.

9.4 Approach for Non-LQP Groundwater Management Problem

This section presents a differential dynamic programming algorithm (DDP) for solving large-scale, nonlinear groundwater management models. The groundwater management model for the optimal control of operational costs of an unconfined aquifer can be posed as follows (Jones, et al., 1987):

$$\underset{q_t}{\text{Min}}\, Z = \sum_{t=1}^{T} q_t'(L - \hat{h}_{t+1}) \tag{9.4.1}$$

subject to

$$h_{t+1} = g(h_t, q_t, t) \qquad t = 1,\ldots, T \tag{9.4.2}$$

$$l^T q_t \geq d_t \qquad t = 1,\ldots, T \tag{9.4.3}$$

$$\underline{q}_t \leq q_t \leq \bar{q}_t \qquad t = 1,\ldots, T \tag{9.4.4}$$

where l' is a row vector of 1's, L is an m vector of the distance from the ground surface to the lower datum of the aquifer; $\hat{h}_t$ is an m vector of hydraulic heads; h_t

is an n vector of hydraulic heads; and $m \leq n$. Operational costs are assumed to be the product of the pumpage rate, q_t', and the left $\left(L - \hat{h}_{t+1}\right)$ Equation (9.4.3) requires that the sum of pumping in each planning period satisfy the demand d_t. Equation (9.4.4) is the capacity constraints.

The above problem (9.4.1) - (9.4.4) is a nonlinear discrete-time, optimal control problem, in which the state equation defines the groundwater hydraulics for two-dimensional flow defined by equation ( ) for confined and/or unconfined flow. These partial differential equations can be expressed in finite difference form for unconfined conditions as

$$\alpha_1 h_{1,t+1}^2 + \alpha_2 h_{2,t+1}^2 + \alpha_3 h_{3,t+1}^2 + \alpha_4 h_{4,t+1}^2 + \alpha_0 h_{0,t+1}^2 + \beta_{h_{o,t+1}} = \beta_o h_{o,t} + l_o \gamma_{o,t} \qquad (9.4.5)$$

where the node numbers refer to Figure (9.4.1), $\alpha_o, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \beta_o$, and γ_o are known functions of the aquifer parameters.

The DDP approach begins with an estimate of the solution for each stage $(q_1, q_2, \ldots, q_n)$ referred to as nominal controls, which is used to solve the simulation (state equation) for the nominal states $(h_1, h_2, \ldots, h_{n+1})$. The nominal solution (h^*, q^*) results in a nominal objective Z^*. Next is to determine a quadratic approximation to the above optimal control problem (9.4.1) - (9.4.4).

Figure 9.4.1 Finite Differences

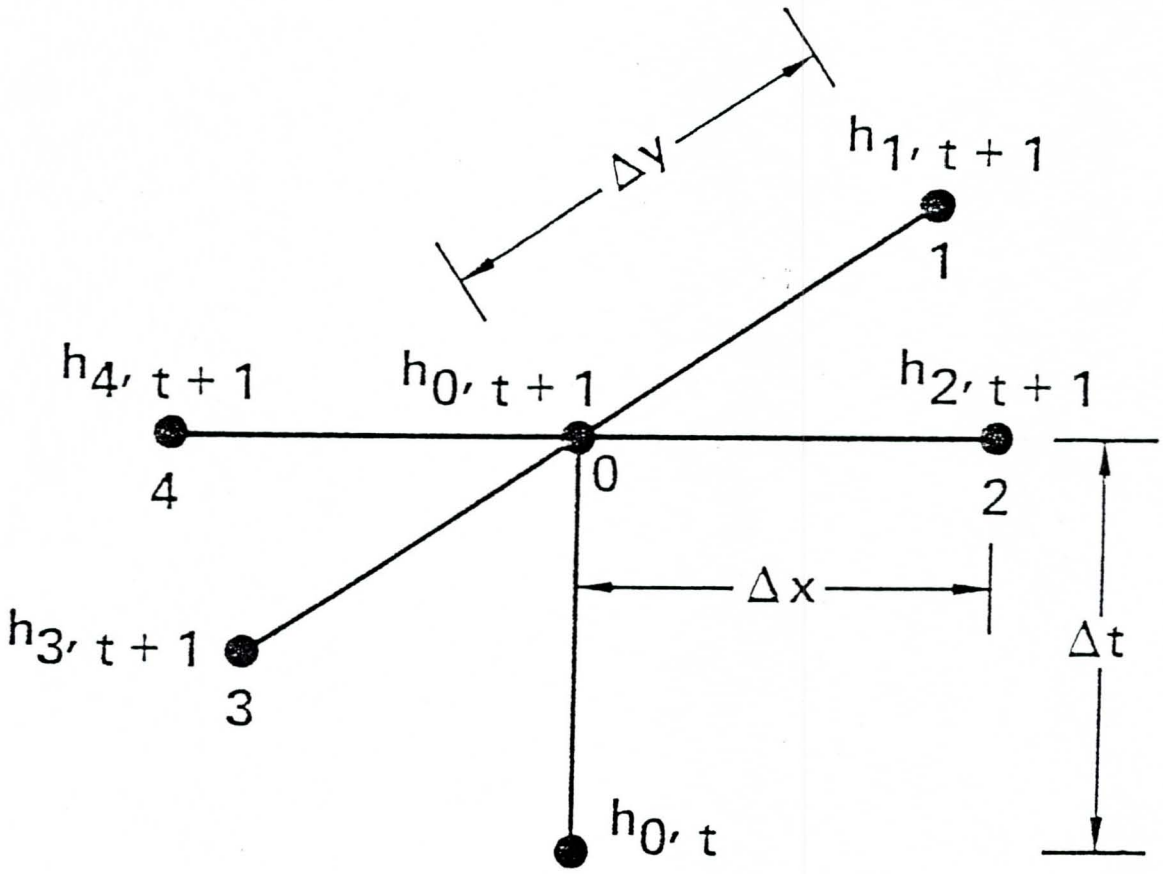


Fig. ~~1~~ 9.4.1. Finite difference approximation at node 0.

In order to develop the quadratic approximation, a first-order Taylor series approximation (linearization) of the groundwater simulation equations about the nominal solution ($\mathbf{h}^*$, $\mathbf{q}^*$) is developed ,

$$h_{i,t+1} = h^*_{i,t+1} + \sum_{j=1}^{n} \frac{\partial h_{i,t+1}}{\partial h_{i,t}} (h_{j,t} - h^*_{j,t}) + \sum_{j=1}^{n} \frac{\partial h_{i,t+1}}{\partial q_{j,t}} (q_{j,t} - q_{i,t})$$

where the deviations are evaluated at ($\mathbf{h}_i^*$, $\mathbf{q}_t^*$). The Taylor series approximation for the vector of heads is

$$\mathbf{h}_{t+1} = \mathbf{h}_{t+1} + R_1(\mathbf{h}_t - \mathbf{h}_t^*) + R_2(\mathbf{q}_t - \mathbf{q}_t^*) \tag{9.4.6}$$

which is a multivariate first-order Taylor series approximation of h_{t+1} as a function of h_t where R_1 and R_2 are Jacobian matrices. The i-th row, j-th column of R_1 is $\partial h_{i,\,t+1} / \partial h_{j,\,t}$ and the i-th row, j-th column of R_2 is $\partial h_{i,\,t+1} / \partial q_{j,\,t}$ both evaluated at (h_t^*,q_t^*).

Determination of the Jacobian matrices are found by implicit differentiation of the simulation equation, e.g. differentiation of (9.4.5) with respect to $h_{i,t}$ results in

$$\begin{aligned} &2\alpha_1 h_{1,\,t+1} \frac{\partial h_{1,\,t+1}}{\partial h_{j,\,t}} + 2\alpha_2 h_{2,\,t+1} \frac{\partial h_{2,\,t+1}}{\partial h_{j,\,t}} + 2\alpha_3 h_{3,\,t+1} \frac{\partial h_{3,\,t+1}}{\partial h_{j,\,t}} \\ &+ 2\alpha_4 h_{4,\,t+1} \frac{\partial h_{4,\,t+1}}{\partial h_{j,\,t}} + 2\alpha_o h_{o,\,t+1} \frac{\partial h_{o,\,t+1}}{\partial h_{j,\,t}} \\ &+ \beta_o \frac{\partial h_{o,\,t+1}}{\partial h_{i,\,t}} = \beta_o \frac{\partial h_{o,t}}{\partial h_{i,\,t}} + \gamma_o \frac{\partial q_{o,t}}{\partial h_{i,\,t}} \end{aligned} \tag{9.4.7}$$

Differentiating each simulation equation with respect to $h_{i,t}$ yields a simultaneous system of n equations for the j-th columns of the Jacobian matrix R_1. The Jacobian matrix is evaluated at $(\mathbf{h}^*_t, \mathbf{q}^*_t)$, $\mathbf{h}^*_{t+1}$ is used instead of $\mathbf{h}_{t+1}$ resulting in a system of n simultaneous linear equations for the j-th column of the Jacobian matrix R_1 evaluated at $(\mathbf{h}^*_t, \mathbf{q}^*_t)$. A system of n simultaneous linear equation is found for each $h_{j,t}$ and $q_{1,t}$ by the same procedure. The left-hand side coefficient matrix of the n simultaneous linear equation for the j-th column is the same, so that for n nodes and m wells the Jacobian matrices for (10.2.6) are found from the solution of n simultaneous linear equations with n+m right-hand sides.

The Taylor series approximation is made for each finite difference time step, dividing the time between stages into a number of finite differences time steps. The linear approximation for each finite difference time step allows simple substitution and matrix algebra to relate h_{t+1} to h_t and q_t to q_{t+1}. Through the use of the first-order approximation to the nonlinear simulation equation, h_{t+1} can be expressed as

$$\mathbf{h}_{t+1} = A_t \mathbf{h}_t + C_t \mathbf{q}_t + \mathbf{z}_t \tag{9.4.8}$$

where A_t is an nxn matrix, C_t is an nxn matrix, and $\mathbf{z}_t$ is an n vector.

The original problem (9.4.1) - (9.4.4) can now be expressed as

$$\underset{\mathbf{q}_t}{\text{Min}}\, Z = \sum_{t=1}^{T} \mathbf{h}_t^T U_t \mathbf{h}_t + \mathbf{q}_t^T V_t \mathbf{h}_t + \mathbf{q}_t^T W_t \mathbf{q}_t + X_t \mathbf{q}_t + Y_t^T \mathbf{h}_t \tag{9.4.9}$$

subject to

$$\mathbf{h}_{t+1} = A_t \mathbf{h}_t + C_t \mathbf{q}_t + \mathbf{z}_t \tag{9.4.10}$$

$$\underline{\mathbf{q}}_t \le \mathbf{q}_t \le \bar{\mathbf{q}}_t \tag{9.4.11}$$

where the matrices in (9.4.9) are $U_t = 0$, $V_t = -\hat{A}_t$, $W_t = -1/2\,(\hat{B} + \hat{B}^1)$, $X_t = L - \hat{z}_t$, and $Y_t = 0$.

The values of $\hat{\mathbf{h}}_{t+1}$ in (9.4.1) for nodes with wells are determined from

$$\hat{\mathbf{h}}_{t+1} = \hat{A}_t \mathbf{h}_t + \hat{B}_t \mathbf{q}_t + \hat{\mathbf{z}}_t \tag{9.4.12}$$

The optimization problem (9.4.9) - (9.4.11) can be referred to as the nominal control problem, which is an approximation to the original control problem (9.4.1) - (9.4.4) about the nominal solution. This problem would be an LQP problem if the two inequalities were not present. They can be placed in the objective through the use of penalty terms. Jones, et al. (1987) suggest incorporating (9.4.11) into the objective of the nominal control problem and solving it as an LQP problem using dynamic programming as described by Murray and Yakowitz (1979). The resulting recursive equation is

$$f_t(\mathbf{h}_t) = \underset{\mathbf{q}_t}{\mathrm{Min}} \Big\{ \mathbf{h}_t \hat{U}_t \mathbf{h}_t + \mathbf{q}_t^T \hat{V}_t \mathbf{h}_t + \mathbf{q}_t \hat{W}_t \mathbf{q}_t$$

$$+ \hat{X}_t^T \mathbf{q}_t + \hat{Y}^T \mathbf{h}_t + f_{t+1}\left(\mathbf{h}_{t+1}\right) - \lambda^T \left(1^T \mathbf{q}_t - \mathbf{d}_t \right) \Big\} \tag{9.4.13}$$

where λ is a Lagrange multiplier estimate and $f_{t+1}(\mathbf{h}_{t+1})$ is the optimal value that gives the optimal return for stages t+1 to T, which is a scalar for this problem. Using the nominal stage, $\mathbf{h}^*_t$ in place of $\mathbf{h}_t$ in (9.4.13), results in a QP problem

which is a subproblem in the background sweep. ~~Figure 9.4.1 presents a flowchart of the algorithm.~~

9.5 Groundwater Reclamation Models

The groundwater reclamation problem can be stated using the general form of the optimal contral problem as

$$\underset{\mathbf{u}}{\text{Min}} = \sum_{t=1}^{T} f_t(\mathbf{x}_t, \mathbf{u}_t, t) \tag{9.5.1}$$

subject to

$$\mathbf{x}_{t+1} = g_t(\mathbf{x}_t, \mathbf{u}_t, t) \qquad t = 1,\ldots, T \tag{9.5.2}$$

$$w(x_t, u_t, t) \leq 0 \qquad t = 1,\ldots,T \tag{9.5.3}$$

where the vector of state variables, x_t, is the vector of hydraulic heads, h_t, and concentrations, c_t,

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{h}_t \\ \mathbf{c}_t \end{bmatrix} \tag{9.5.4}$$

The vector of contral variables are the pumping rates at a set of m possible and or existing well locations. The state equation (9.5.2) defines the groundwater flow and contaminant transport for the aquifer. Constraint equation (9.5.3) includes constraints on both the state and contral. As an example, constraints on the state variable can include water quality requirements on the concentration levels and pumpage limitations on the contral variable.

Culver and Shoemaker (1992) present a groundwater remediation model based upon a DDP approach that they refer to as a successive approximation

linear quadratic regulator (SALQR). Their definition of SALQR is that it "differs from DDP only in that the nonlinear simulation equations are linearized in the optimization step." SALQR and DDP are identical if the simulation equations are linear. The optimal contral problem solved by Culver and Shoemaker (1992) with flexible management periods is defined as

$$\text{Min } Z = \sum_{k=1}^{k} F_k (x_k, u_k, k) \tag{9.5.5}$$

subject to

$$x_{k+1} = Y (x_k, u_k, k) \qquad k = 1,..., K \tag{9.5.6}$$

$$L (x_k, u_k, k) \leq 0 \qquad k = 1,..., k \tag{9.5.7}$$

where k refers to the management period; K is the total number of management periods defined as K = T/d; T is the total number of simulation steps; and d is the number of simulation periods per management period. The above problem (9.5.5) - (9.5.7) is directly analogous to (9.5.1) - (9.5.4). The state equation (9.5.6) describes the change in x over a management period instead of over a simulation step. The algorithm defined by Culver and Shoemaker (1992) to solve this problem is presented in Figure (9.5.1).

The cost function used by Culver and Shoemaker (1992) defines the total operating costs of pumping and treatment during each time period given as

$$f_k (x_k, u_k, k) = \sum_{i=1}^{m} a_{1i} u_{ki} + \sum_{i=1}^{m} a_{2i} u_{ki} (\bar{h} - h_{k+1}) \tag{9.5.8}$$

in which the first term defines water treatment costs on a linear function of the amount of water pumped (extracted) and the second term defines the pumping

$\frac{\partial g}{\partial x}$ $\frac{\partial g}{\partial u}$ $\frac{\partial^2 g}{\partial x^2}$ $\frac{\partial g}{\partial u^2}$ $\frac{\partial^2 g}{\partial x \partial u}$ all have to be computed, so

Successive approximation linear quadratic regulator (SALQR)

when Taylor Series is used, the 2nd order terms become zero.

Simplifies things when 2nd derivatives are difficult to compute.

Read this paper.

Notice this backward step is a linearization of the non-linear transition eqn using Taylor Series.

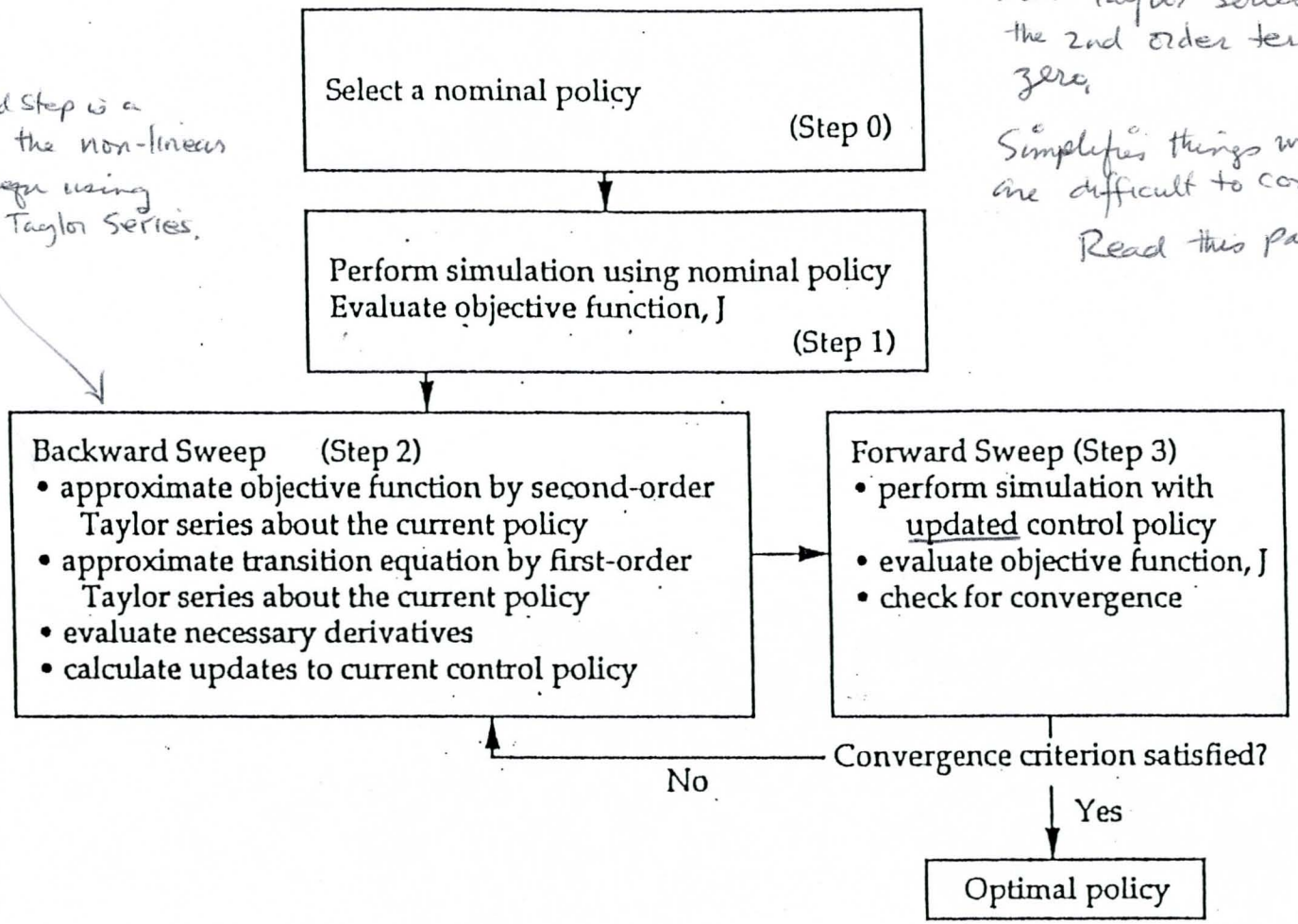


Fig. ~~1.~~ 1 Diagram of optimization algorithm for SALQR. Step numbers correspond to the mathematical description in the appendix. (Culver and Shoemaker, 1992)
9.5.1

costs related to the product of the extraction rate and the left $(\bar{h} - h_{k+1})$. $\bar{h}$ is the vector of distances from the ground surface to the lower boundary of the aquifer.

The constraint equation (9.5.7) on the contral and state variables are incorporated into the objective function (10.3.8) utilizing a penalty function so that

$$\hat{f}_k(x_k, u_k, k) = \sum_{i=1}^{m} a_{1i} u_{ki} + \sum_{i=1}^{m} a_{2i} u_{ki} (\bar{h} - h_{k+1})$$

$$+ \sum_{j=1}^{J} y_{ki}(r_{ki}) \qquad (9.5.9)$$

where y_{ki} ( ) is the penalty (cost) associated with the violation, k_i of constraint j in period k. Culver and Shoemaker (1992) used the following penalty function

$$y_{ki}(c_{ki}) = \varepsilon_{kj} \qquad \varepsilon_{ki} \leq 1 \qquad (9.5.10)$$

$$y_{ki}(c_{ki}) = a\,\varepsilon_{ki}^2 + b\,(\varepsilon_{ki})^{1/2} + c \qquad \varepsilon_{ki} > 1 \qquad (9.5.11)$$

gives the following hyperbolic penalty function by Lin (1990)

$$\varepsilon_{ki} = (\alpha_{ki}^2 r_{ki}^2 + \beta^2)^{1/2} + \alpha_{ki} r_{ki} \qquad (9.5.12)$$

where α_{ki} is the weighting coefficient of the j-th constraint; β_{ki} is a shape parameter of the hyperbolic function ε_{ki}; and a, b, and c are constant coefficients.

Culver and Shoemaker (1992) point out that with the inclusion of equations (9.5.10) and (9.5.11) ie. the penalty function, the objective function (9.5.9) is not quadratic. During the backward sweep (refer to Figure 9.5.1), the objective (9.5.9) is approximated as a quadratic by a second order Taylor series expansion about the current policy as suggested by Yakowitz and Rutherford (1984). The first

See paper on convergence.

but it is approximated by the Taylor Series.

derivatives of the state equation over a management period requires the first derivatives of the state equation over a simulation period.

corrected in Class 11/12/92

$$\left(\frac{\partial g}{\partial x}\right)_k = \prod_{t=p}^{p+d-1} \left(\frac{\partial g}{\partial x}\right)_t \tag{9.5.12}$$

$$\left(\frac{\partial g}{\partial u}\right)_k = \left(\frac{\partial g}{\partial u}\right)_{p+d-1} + \left(\frac{\partial g}{\partial x}\right)_{p+d-1}\left(\frac{\partial g}{\partial u}\right)_{p+d-2} + \left(\frac{\partial g}{\partial x}\right)_{p+d-1}\left(\frac{\partial g}{\partial x}\right)_{p+d-2}\left(\frac{\partial g}{\partial u}\right)_{p+d-3}$$

$$+\ldots+ \prod_{t=p+1}^{p+d-1} \left[\frac{\partial g}{\partial x_t}\right] \left(\frac{\partial g}{\partial u_p}\right)_p \tag{9.5.13}$$

where p = *k-1) d + 1. These derivatives for $\left(\frac{\partial g}{\partial x}\right)_k$ and $\left(\frac{\partial g}{\partial u}\right)_k$ above are based upon the product rule of differentiation.

The derivatives can be determined analytically over a simulation period as shown by Chang (1990) who compute the derivatives from the equations of the finite element model, ISOQUAD by Pinder (1979). The algorithm used by Culver and Shoemaker (1992) is basically the same as the one described in Section (?) with the exception of equations (?) - (?) in which

$$1/2 \sum_{i=1}^{n} \left(Q_{t+1}\right)_i \left(\frac{\partial^2 g}{\partial x_i^2}\right) = 0 \tag{9.5.14}$$

$$\sum_{i=1}^{n} \left(Q_{t+1}\right)_i \left(\frac{\partial^2 g}{\partial x_i \partial u_i}\right) = 0 \tag{9.5.15}$$

$$1/2 \sum_{i=1}^{n} \left(Q_{t+1}\right)_i \left(\frac{\partial^2 g}{\partial u_i^2}\right) = 0 \tag{9.5.16}$$

Liao,

Christine Shoemaker.

References

Bellman R. and S. Dreyfus, Applied Dynamic Programming, Princeton, U.P., Princeton, N.J., 1962.

Chang, L. C., The application of constrained optimal control algorithms to groundwater remediation, Ph.D. dissertation, Cornell Univ., Ithaca, N. Y. 1990.

Teresa

Culver, T. B., Dynamic optimal control of groundwater remediation with management periods: Linearized and quasi-Newton approaches, Ph.D. dissertation, Cornell Univ., Ithaca, N.Y., 1991.

Culver, T. and C. Shoemaker, Dynamic Optimal Control for Groundwater Remediation with Flexible Management Periods, Water Resources Research, Vol. 28, No. 3, pp. 629-641, March 1992.

Dyer P. and S. McReynolds, The Computational Theory of Optimal Control, Academic, New York, 1970.

Gershwin S. and D. Jacobson, A discrete-time differential dynamic programming algorithm with application to optimal orbit transfer, AIAA J. 8:1616-1626 (1970).

Jacobson D. and D. Mayne, Differential Dynamic Programming, Elsevier, New York, 1970.

Jones, L. C., R. Willis, and W. W. Yeh, Optimal control of nonlinear groundwater hydraulics using differential dynamic programming, Water Resour. Res., 23(11), 2097-2217, 1987.

Mayne D., A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems, Internat. J. Control 3:85-95 (1966).

Murray M. and Yakowitz, The application of optimal control methodology to nonlinear programming problems, Math. Programming, 21:331-347 (1981).

Murray, D. M., and S. J. Yakowitz, Constrained differential dynamic programming and its application to multireservoir control, Water Resour. Res., 15(5), 1017-1027, 1979. (10 reservoirs)

Ohno K., A new approach of differential dynamic programming for discrete time systems, IEEE Trans. Automat. Control AC-23:37-47 (1978).

Pinder, G. F., Galerkin finite element models for aquifer simulation, Rep. 76-WR-5, Dep of Civ. Eng., Princeton Univ., Princeton, N. J., 1979.

Yakowitz, S. and B. Rutherford, Computational Aspects of Discrete-Time Optimal Control, Applied Mathematics and Computation, Vol. 15, pp. 29-45, 1984.

Yakowitz, S., Algorithms and Computational Techniques in Differential Dynamic Programming, Control and Dynamic Systems, Academic Press Inc., Vol. 31, pp. 75-91, 1989.