AQMAN: LINEAR AND QUADRATIC PROGRAMMING MATRIX GENERATOR USING TWO-DIMENSIONAL GROUND-WATER FLOW SIMULATION FOR AQUIFER MANAGEMENT MODELING

By L. Jeff Lefkoff and Steven M. Gorelick

U.S. GEOLOGICAL SURVEY

Water-Resources Investigations Report 87-4061

Menlo Park, California

1987

DEPARTMENT OF THE INTERIOR

DONALD PAUL HODEL, Secretary

U. S. GEOLOGICAL SURVEY

Dallas L. Peck, Director

For additional information
write to:

Regional Hydrologist, WR
U.S. Geological Survey
345 Middlefield Road
Menlo Park, California 94025

Copies of this report can
be purchased from:

Books and Open-File Report
U.S. Geological Survey
Box 25425, Federal Center
Denver, Colorado 80225
(Telephone: (303) 236-7476)

PREFACE

This report describes a computer model that combines ground-water flow simulation with mathematical optimization in order to develop and evaluate aquifer management strategies. The computer program is intended for general application and may have to be modified by the user for specific field problems. Although the program will produce reliable calculations for a wide variety of problems, the user is cautioned that in some cases the accuracy of management solutions can be significantly affected by user discretion during implementation.

The user is requested to kindly notify the originating office of any errors found in this report or in the computer program. Updates may occasionally be made to both the report and the computer program. Users who wish to be added to the mailing list to receive updates, if any, may send a request to the following address:

AQMAN
U.S. Geological Survey
345 Middlefield Road, MS-421
Menlo Park, CA 94025

Copies of the computer program on tape are available at cost of processing from:

U.S. Geological Survey
WATSTORE Program Office
437 National Center
Reston, VA 22092
Telephone: (703) 648-5686

The use of computer and software brand names in this report is for identification purposes only and does not imply endorsement by the U. S. Geological Survey.

CONTENTS

ILLUSTRATIONS

LIST OF TABLES

CONVERSION FACTORS

For convenience of readers who prefer to use metric (International System) units, conversion factors for inch-pound units used in this report are listed below:

Multiply inch-pound unit	By	To obtain metric unit
foot (ft)	0.3048	meter (m)
cubic feet per second (ft^3/s)	0.02832	cubic meter per second (m^3/s)

ABSTRACT

A FORTRAN-77 computer program code that helps solve a variety of aquifer management problems involving the control of ground-water hydraulics is presented. It is intended for use with any standard mathematical programming package that uses Mathematical Programming System input format. The computer program creates the input files to be used by the optimization program. These files contain all the hydrologic information and management objectives needed to solve the management problem.

Used in conjunction with a mathematical programming code, the computer program identifies the pumping or recharge strategy that achieves a user's management objective while maintaining ground-water hydraulic conditions within desired limits. The objective may be linear or quadratic, and may involve the minimization of pumping and recharge rates or of variable pumping costs. The problem may contain constraints on ground-water heads, gradients, and velocities for a complex, transient hydrologic system.

Linear superposition of solutions to the transient, two-dimensional ground-water flow equation is used by the computer program in conjunction with the response matrix optimization method. A unit stress is applied at each decision well and transient responses at all control locations are computed using a modified version of the U.S. Geological Survey two dimensional aquifer simulation model. The program also computes discounted cost coefficients for the objective function and accounts for transient aquifer conditions.

INTRODUCTION

Numerical simulation models have become essential tools for hydrogeologists. Often models are utilized for evaluating different ground-water management strategies. Such strategies may be needed for aquifer management problems that involve:

- contaminant plume stabilization and removal
- coastal aquifer protection
- maximization of aquifer yields
- design of surface excavation dewatering systems
- development of policies for conjunctive use of surface and ground waters
- inspection of the impacts of water use regulation.

The difficulty with trying to use simulation models to study these problems is that potential strategies can be tested only on a trial and error basis. There is no guarantee that the best alternatives will be discovered. Simulation by itself is valuable for understanding system behavior and in some cases for predicting future responses, but it is far less valuable as a tool for discovering optimal aquifer management strategies. In order to effectively determine viable and efficient aquifer management plans, simulation can be combined with optimization procedures of operations research. Operations research is a general field of applied mathematics that is concerned with optimal decision making.

AQMAN is a FORTRAN-77 computer code that provides a link between ground-water simulation and two widely used techniques developed in operations research: linear programming and quadratic programming. These techniques provide a mathematical framework and efficient computational algorithms to determine the optimal allocation of scarce resources. Water, its cost of extraction, and the costs required to maintain its quality are the scarce resources of interest to the hydrologist. Combined simulation-management models can be used to identify pumping and recharge schedules that achieve some goal, such as minimizing ground-water production costs, while simultaneously protecting water quality and satisfying water demands. This procedure is known as **aquifer management modeling.**

Aquifer management modeling is concerned with the best selection of well locations and pumping and recharge rates that achieves certain goals with regard to aquifer yields, drawdowns, hydraulic heads, hydraulic gradients, and ground-water velocities. It also can involve recharge through streambeds, ground-water flow to streams and lakes, maintenance of surface-water levels, and other interactions between surface water and ground water. Aquifer management modeling is a a multistaged procedure. First, a simulation model is developed for a particular field site. The hydrologic behavior of the site should be well understood. Second, a management problem is formulated. Third, the simulation model is used to generate a compact simulator called a response matrix. Fourth, a special data file is created that contains the response matrix and represents the management formulation in a format that is required for solution by any of a number of available optimization codes. Fifth, a standard linear or quadratic programming code reads the special file and determines the optimal solution. Sixth, the effect of the pumping and recharge schedules prescribed by the optimal solution is verified using the original simulation model. Seventh, the sensitivity of the solution to uncertainties is explored.

The AQMAN program performs stages three and four: generation of the response matrix and construction of the optimization data file. It is also useful in stage seven, sensitivity analysis.

Given a field area for which a simulation model has been developed and for which a management problem has been formulated, AQMAN can be used to evaluate alternative management strategies. It is applicable to problems for which two-dimensional confined aquifer simulation models are appropriate. AQMAN uses the code of Trescott and others (1976) to simulate the set of system responses to pumping or recharge, as well as other system stresses. It then converts this information into a response matrix and creates the optimization data file. From that point on an optimization code is employed to solve the linear or quadratic programming problem.

The use of AQMAN for aquifer management modelling requires a thorough understanding of aquifer simulation and the principle of linear superposition. Experience with the optimization methods of linear and quadratic programming is helpful, but not essential. The user should be familiar with aquifer flow simulation and should be able to use the code of Trescott and others (1976) for the problem of interest. AQMAN's programming logic is quite general for aquifer management modelling, so that with appropriate modifications other ground-water flow simulators may be linked with AQMAN. However, this may require several changes in both the flow simulator and in the AQMAN code. These changes should be performed only by experienced ground-water modellers. In its current form, AQMAN is fully linked only with the Trescott code.

Aquifer simulation must be performed for systems that show a linear drawdown response to pumping and are sufficiently described by the equation that governs two-dimensional ground-water flow:

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

where H = hydraulic head [L],
T_{ij} = transmissivity tensor [L^2/T],
S = storage coefficient [L^0],
W = source (recharge) or sink (pumping) per unit area [L/T],
t = time [T],
x_i x_j = spatial coordinates [L].

GROUND-WATER MANAGEMENT MODELING

Background

Ground-water management modeling is a relatively new discipline in hydrology. Over the past 25 years, two types of management models have been developed: lumped parameter and distributed parameter models. **Lumped parameter models** have been used to study economic and policy matters that involve ground-water resources. They do not explicitly consider the governing equations of ground-water flow, but rather conceptualize aquifers with simple water mass balances. Examples of lumped parameter models are those of Chaudhry and others (1974), Anderson and others (1983), and Khepar and Chaturvedi (1982). **Distributed parameter management models** join aquifer simulation with optimization methods and explicitly solve the partial differential equation that governs flow. Gorelick (1983) discusses and critically evaluates the methods and applications of these models. They have been used to manage well fields, to evaluate efficient conjunctive use of stream-aquifer systems, and to inspect the impacts of water-resource policies upon the hydrology and economics of ground-water use. AQMAN is a tool that helps solve distributed parameter aquifer management problems.

AQMAN links a distributed parameter ground-water simulation model with mathematical optimization methods using a technique known as the **response matrix approach** (see Gorelick, 1983). This approach was initially developed for optimizing profits from oil production, and was presented in the petroleum engineering literature by Lee and Aronofsky (1958). During the 1970's the technique was expanded and brought into the hydrologic literature, principally by Wattenbarger (1970), Maddock (1972), Rosenwald and Green (1974), and Schwarz (1976). Applications of the response matrix approach have been presented by Larson and others (1977) and by Heidari (1982) to determine "safe yield" of aquifers; by Willis (1983) to determine the optimal pumping scheme to meet agricultural water demands; by Danskin and Gorelick (1985) to evaluate the efficiency of a surface-water recharge program; and by Colarullo and others (1984), Atwood and Gorelick (1985), Gorelick and Wagner (1986), and Lefkoff and Gorelick (1985; 1986) to contain plumes of contaminated ground water and design aquifer restoration systems.

The Response Matrix Method

The key idea behind the response matrix method is that because a ground-water system described by equation (1) is linear, the influence of each source or sink may be calculated separately and then superposed to compute the complete distribution of hydraulic heads over space and time under any pattern of pumping and recharge. The method fully accounts for the effects of initial conditions, which may vary over space, and the effects of boundary conditions, which may change over time.

A response matrix is an assemblage of coefficients, each of which relates pumping at one location to drawdown at another location. In order to see how a response matrix is developed using a simulation model, consider an example where head is to be controlled at a location P during two months by managing pumping at two wells, A and B. By use of a unit pumping rate, drawdown is calculated at the end of each month at location P.

Specifically,

(1) Reference-head at P is computed by a transient simulation with no pumping.

(2) A convenient unit pumping rate, 1.0 ft^3/s, is selected.

(3) A transient simulation is performed with only well A pumping at the unit rate during the first month and not pumping during the second month. Calculated head at P is subtracted from the reference head to obtain a drawdown for each month. For example, say that drawdown is 0.75 feet at the end of the first month and 0.20 feet at the end of the second month.

(4) Step (3) is repeated for well B. Say that the calculated drawdown at P from the reference head is 0.40 feet at the end of the first month and 0.25 feet at the end of the second month.

Because the system is linear, total drawdown at P induced by managed pumping during each month equals the sum of the drawdowns due to the two wells:

$$s_1 = 0.75\ Q_{A,1} + 0.40\ Q_{B,1}$$

$$s_2 = 0.20\ Q_{A,1} + 0.25\ Q_{B,1} + 0.75\ Q_{A,2} + 0.40\ Q_{B,2}$$

where s = total drawdown [feet] from the reference head at location P at the end of month 1 or 2 due to managed pumping,

Q = pumping rate [ft^3/s] at well A or B during month 1 or 2.

These two equations can be expressed in matrix form:

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} .75 & .40 & 0 & 0 \\ .20 & .25 & .75 & .40 \end{bmatrix} \begin{bmatrix} Q_{A,1} \\ Q_{B,1} \\ Q_{A,2} \\ Q_{B,2} \end{bmatrix}$$

vector of drawdowns = response matrix x vector of pumping rates

Response coefficients (0.75, 0.40, 0.20, 0.25) are stored in a matrix, thus the name **response matrix method**. Through matrix algebra, a large number of responses can conveniently be added over time and space. For any given set of initial and boundary conditions, total drawdown at a given location will be a linear function of pumping and recharge at all wells during all management periods.

The small example above demonstrates two general features of response matrices. First, the matrix contains zero elements in the upper triangular portion because pumping or recharge in some management period can never affect drawdowns in an earlier period. Second, matrix elements are repeated systematically. This is because the response to a unit of pumping depends only on the time since pumping began. That is, the response to a unit pumping rate that begins today is the same as the response to a unit pumping rate that begins next month, except that the responses are lagged by one month. The matrix in the example above has the following general structure:

drawdown response in month 1 to unit pumping in month 1	zero response
drawdown response in month 2 to unit pumping in month 1	**drawdown response in month 2 to unit pumping in month 2** (same as upper left)

When generating the response matrix, the number of computations performed in AQMAN is reduced by taking advantage of this matrix structure.

GENERAL PROBLEM FORMULATION FOR LINEAR AND QUADRATIC PROGRAMS

The process of solving a linear or quadratic programming problem begins with the formulation of a management problem as a mathematical model. As with all models, the mathematical formulation extracts the essence of the real-world system and does not consider every detail. Problem formulation may reveal inadequacies in the data. It may also happen that the formulation process, which is often avoided during trial and error simulation, will lead directly to a highly simplified management model, or even to an apparent solution.

Problem formulation is certainly the most important and often the most difficult part of management modelling. A management model consists of an **objective function** (or goal) which to be minimized or maximized, and a series of linear **constraints** (or restrictions) that must be obeyed. **Decision variables** are the unkown quantities of concern that can be controlled in a managed system. For our purposes, these are pumping and recharge rates at specific locations.

Linear and Quadratic Objectives

Linear programming is used for cases in which the objective function, whose value is F, is linear with respect to the decision variables. For example, one might maximize the sum of pumping rates, $Q_{i,n}$ at five wells (index i) during eight time periods (index n):

$$\text{Maximize} \quad F = \sum_{n=1}^{8} \sum_{i=1}^{5} Q_{i,n} \tag{2}$$

Quadratic programming is used for cases in which the objective is a quadratic function of the decision variables. For example, one might minimize the sum of squared differences between pumping rates and ideal target pumping rates, $Q^*_{i,n}$:

$$\text{Minimize} \quad F = \sum_{n=1}^{8} \sum_{i=1}^{5} (Q_{i,n} - Q^*_{i,n})^2 \tag{3}$$

Typical objective functions are:

- Minimize the cost of pumping
- Minimize total pumpage
- Maximize total pumpage
- Minimize the maximum pumping rate
- Maximize total recharge
- Minimize the maximum drawdown
- Maximize the minimum hydraulic head
- Minimize the sum of squared deviations from target heads, hydraulic gradients, velocities, drawdowns, or pumping rates
- Minimize the sum of the absolute value of deviations from target heads, hydraulic gradients, velocities, drawdowns, or pumping rates

Objectives may also involve fixed costs, such as well installation or capital investment in pumping capacity. Further, it is possible to formulate problems with multiple objectives. Aguado and Remson (1980) describe the fixed cost problem, and Cohon and Marks (1975) provide an excellent review of multi-objective analysis. AQMAN does not automatically generate fixed cost objectives nor multiple objective functions, but is nonetheless useful for solving such problems.

Constraints and System Linearity

Constraints derive from the physical, economic, or social mechanisms operating in the managed system. Typical ground-water constraints might involve

- definition of drawdowns as a linear function of system stresses using a response matrix
- definition of hydraulic heads, velocities, and gradients as a function of system stresses
- limitations on local drawdowns, hydraulic gradients, velocities or heads
- restrictions on local hydraulic gradients or velocities to certain magnitudes and directions
- limitations on pumping rates at individual or groups of wells
- restrictions on changes in pumping rates, drawdowns, or hydraulic heads over time
- balances between total pumping and total recharge

Constraints may be represented by placing **simple bounds** on individual decision variables, by **inequalities** or **equalities** placed on groups of decision variables, or by **restrictions** on hydraulic conditions. In all cases, a specified limit appears on the right-hand side. For example, the pumping rate Q at well 2 during period 3 must not exceed 1.5 ft^3/s:

$$Q_{2,3} \leq 1.5 \tag{4}$$

Total pumpage for three wells must supply at least 5.0 ft^3/s during period 1:

$$Q_{1,1} + Q_{2,1} + Q_{3,1} \geq 5.0 \tag{5}$$

Pumping rates at well 1 must be the same for periods 3 and 4:

$$Q_{1,3} - Q_{1,4} = 0 \tag{6}$$

Hydraulic head H at location 3 must not exceed 50 feet during period 1:

$$H_{3,1} \leq 50 \tag{7}$$

During period 4, the hydraulic gradient between locations 1 and 2 must not be less than 5 percent (L is the distance between the two locations):

$$\frac{H_{1,4} - H_{2,4}}{L} \geq 0.05 \tag{8}$$

The seepage velocity V at location 2 must be at least 0.03 ft/s during period 5:

$$V_{2,5} \geq 0.03 \quad (9)$$

At location 3 during period 1, the x-component of the seepage velocity must be at least twice the y-component:

$$V^{x}_{3,1} - 2V^{y}_{3,1} \geq 0 \quad (10)$$

In both linear and quadratic programming problem formulations all constraints must be linear. This means that the ground-water system must respond linearly to management decisions. The hydraulics of a confined aquifer can be managed successfully using linear or quadratic programming, since ground-water flow is governed by equation (1), which contains a linear relation between head changes and well pumpages.

AQMAN is useful only for linear management problems. Nonlinear constraints cannot be imposed. For example, nonlinearities resulting from dewatering of unconfined aquifers cannot be rigorously handled by AQMAN. However, in some cases it may be possible to linearize such systems if drawdown is small compared to saturated thickness, or by solving sequential linear problems where the saturated thickness is given by the last iterate (Danskin and Gorelick, 1985). There is a certain art in formulating problems to avoid nonlinearities. For instance, the absolute value function, which is in fact nonlinear, has a linear equivalent that can be easily incorporated into linear programming problems. The problem and methods for deriving linear formulations is discussed in various operations research text books such as Dantzig (1963), Hillier and Lieberman (1974), and Wagner (1975).

Advanced techniques are now available to manage nonlinear systems (Gorelick and others, 1984). Nonlinear management modelling problems are not restricted to aquifer hydraulics. They involve the solution of governing equations which are not linear with respect to management decisions. For instance, a problem which calls for the management of solute concentrations when those concentrations are affected by pumping and recharge decisions is nonlinear.

Time Parameters

Three time parameters are used in management modelling. The **planning horizon** is the total length of time over which a system is being managed. It consists of one or more **management periods** or **planning periods**, which define the time during which a particular decision variable is constant or a particular set of constraints applies. For instance, we may define a one-year planning horizon consisting of 12 one-month management periods. A separate decision variable would be defined for each well during each month, and a constraint that relates pumpage to drawdown would be formulated for each month. AQMAN requires that all management periods are equal in length. The third time parameter is the numerical **time step** used in the finite-difference simulation procedure.

Stress and Response

Drawdowns

Use of the response matrix method for ground-water management modeling relies on the concept of linear response to system stress. A **stress** is an excitation that induces a change in aquifer heads. An **unmanaged stress** is one that cannot be controlled due to physical limitations or social demands. Examples might be boundary conditions and pre-determined pumpage or recharge rates given for **unmanaged wells.** **Managed stresses** are those described by the decision variables: pumpage and recharge at **managed wells,** where management decisions must be made. In this context, "well" refers to any source or sink. In addition to actual pumping or injection wells, the term includes stresses such as recharge from a lake or from a river where flow is controlled. In order to strictly maintain system linearity, the flux through a managed well can not be head-dependent. **Unstressed heads** or **unmanaged heads** are those that would occur if no managed stresses were applied to the system (Figure 1). **Manageable drawdown** is the difference between unmanaged head and any limit on head imposed by the user. For example, suppose that head at a location must always be greater than 88.0 feet, and that the unmanaged head there is 93.8 feet in the first management period, and 92.4 feet in the second period. Then the manageable drawdown in the two periods are +5.8 feet and +4.4 feet, respectively.

A **unit stress** is a convenient quantity of managed stress, such as 0.1 ft^3/s or 1.0 ft^3/s. For each separate managed well, AQMAN applies a unit stress at the well during the first management period and then applies no managed stresses during subsequent periods. The transient ground-water flow equation (1) is solved to obtain hydraulic head at every control location during every period. The **drawdown response** is the difference between this head and unmanaged head. Every response appears in the management formulation as a **response coefficient.** The coefficient measures the hydraulic relation between stress at a particular managed well and drawdown at a particular control location. Through the use of linear superposition, the product of the response coefficient and the actual managed stress gives the actual drawdown at the control location induced by pumping or recharge at the managed well. **Total drawdown** equals the sum of the individual drawdowns caused by each managed well, plus the drawdown caused by initial and boundary conditions and unmanaged pumping and recharge. Total drawdown must be contained within the limits imposed by the manageable drawdown.

Gradients and velocities

A **control pair** is defined by two control locations across which a gradient or hydraulic ground-water velocity is being constrained. In order to define a gradient or a velocity, the **difference-in-drawdown** response is used in a manner similar to drawdown response. Each difference-in-drawdown response also appears in the management formulation as a response coefficient. The coefficient measures the hydraulic relation between stress at a decision well and the difference-in-drawdown within a control pair. The response coefficient multiplied by the managed pumping and recharge rates gives the difference in drawdown. The **total difference-in-drawdown** equals the sum of the influences due to managed stress, unmanaged stress, and initial and boundary conditions. This total difference is linearly proportional to the gradient and to the velocity at the control pair.

The total difference-in-drawdown must be contained within the limits imposed

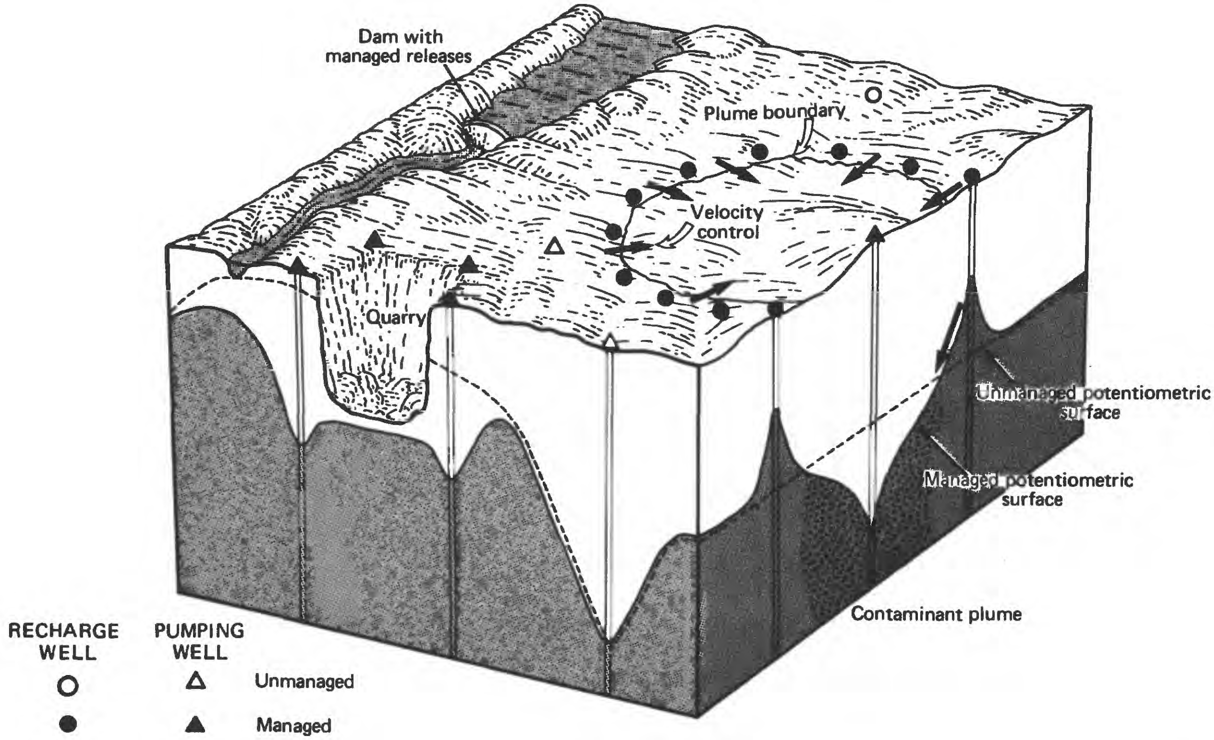

Figure 1. Managed and unmanaged potentiometric surfaces with several wells.

by the **manageable gradient** or **manageable velocity** for the control pair. The manageable gradient is the difference between the gradient due to unmanaged heads and any limit on the gradient imposed by the user. The manageable velocity is the difference between the velocity under the unmanaged gradient and any limit on velocity imposed by the user. In both cases, the user's limit on gradient or velocity is multiplied by a factor that converts it to a head difference. For example, say that control pair 2 is defined by control location 4 and 7 (These are indicated below by the subscripts "2", "4", and "7"). Ignoring the time dimension for the sake of clarity,

$$G_2 = H_4 - H_7 - \bar{G}_2 \ (L_2) \tag{11}$$

$$V_2 = H_4 - H_7 - \bar{V}_2 \left[\frac{\epsilon_2 L_2}{K_2}\right] \tag{12}$$

where G = manageable gradient within the control pair [L],

V = manageable velocity within the control pair [L],

H = unmanaged head at a control location [L],

$\bar{G}$ = limit on gradient imposed by the user $[L^o]$,

$\bar{V}$ = limit on velocity imposed by the user [L/T],

L = distance between the two locations of the control pair [L],

ϵ = effective porosity within the control pair $[L^o]$,

K = hydraulic conductivity within the control pair [L/T].

The conversion factor L or ($\epsilon L/K$) is read as input by AQMAN.

Head, gradient, and velocity definitions

Ground-water management problems are usually formulated in terms of drawdown or difference-in-drawdown. AQMAN also allows for **head definition**, **gradient definition**, or **velocity definition** at any control location or control pair as a supplement to drawdown information. The use of definitions may serve two purposes. First, the management solution will directly contain the heads, gradients, or velocities that result in response to optimal pumping and recharge rates. This helps the user characterize the optimal management strategy. Second, the difference between two gradients or two velocities can be easily controlled. This is particularly useful in comparing the magnitude of the two vector components of the gradient or velocity at a pair of control locations.

Definitions can be specified for head at a single control location or for difference-in-head within a control pair. In the first case, AQMAN automatically introduces a new decision variable and sets it equal to the difference between unmanaged head and total drawdown. In the second case, the new decision variable is set equal to the difference between: (1) the difference in unmanaged heads and (2) the total difference-in-drawdown. The user can then impose additional constraints on the new head, gradient, or velocity variables.

The MPS File and the Solution

AQMAN creates a data file that defines the objective and all constraint functions and contains all of the response coefficients and manageable drawdowns, gradients, and velocities. The file is written in MPS (Mathematical Programming System) format, which is required by most standard linear and quadratic programming packages. The optimization package reads this MPS file as input, and computes a solution to the management problem.

A **management solution** consists of a set of values for the decision variables. A solution is either infeasible, optimal, or unbounded. An **infeasible** solution violates one or more of the constraints. A feasible solution is **optimal** if it produces the best (maximum or minimum) possible value of the objective function. An optimal solution is usually what we hope to find! A problem may have more than one optimal solution, where several management strategies satisfy the constraints and produce the same optimal value for the objective. A solution is **unbounded** if the optimal value of the objective function goes to positive or negative infinity without violating any of the constraints.

PROBLEM FORMULATION WITH AQMAN

This section describes the general procedures required to utilize AQMAN, including manual changes to the MPS file. Detailed instructions for AQMAN input files are given in Appendix II.

Objective Function

Linear objective

AQMAN is written to handle either a linear or a quadratic objective function. If the linear option is in effect, AQMAN assumes that the linear objective is to minimize or maximize total pumping:

$$\sum_{n=1}^{N} \sum_{i=1}^{I} C_{i,n} Q_{i,n} \tag{13}$$

where I = total number of managed wells,

N = total number of management periods,

$Q_{i,n}$ = pumping rate at well i during period n $[L^3/T]$,

$C_{i,n}$ = cost coefficient for well i during period n $[L^0]$.

All cost coefficients are assumed to equal +1.0. These are automatically scaled by multiplying by the negative of the unit stress pumping or recharge rate. (See the sub-section below on "The Unit Stress and Scaling".) If no scaling is required and all unit stresses are either -1.0 (pumping) or +1.0 (recharge), cost coefficients will be +1.0 for pumping wells and -1.0 for recharge wells. These are written by AQMAN to the MPS file. Both pumping and recharge rates are non-negative by default in the MPS file.

If a linear objective other than (13) is desired, the user can make changes directly to the MPS file. For instance, say that the problem contains two decision wells, and management criteria specify that pumpage at the first well should be weighted twice as much as pumpage at the second well. Cost coefficients become weights, and the objective would be to optimize

$$\sum_{n=1}^{N} \left[2\, Q_{1,n} + 1\, Q_{2,n}\right] . \tag{14}$$

The cost coefficient written by AQMAN to the MPS file for Q_1 should be manually changed to twice its value for each management period n.

Quadratic objective

If the quadratic option is specified, AQMAN assumes that the objective is to minimize the present value of variable pumping costs. These are the costs associated with the energy required to operate pumps, and do not include capital (fixed) costs of well installation or pumping capacity. Variable pumping costs are a quadratic function of pumping decisions. This is because costs vary with pumping rates and pumping lifts, and lifts depend on pumping rates:

$$P = \sum_{n=1}^{N} \sum_{i=1}^{I} C_{i,n} Z_{i,n} Q_{i,n} \tag{15}$$

where P = total variable pumping cost [\$],

$Z_{i,n}$ = total lift at well i during period n [L],

$C_{i,n}$ = unit cost of pumping per unit lift at well i during period n [$\$/(L^3/T)/L$].

The total lift is a function of pumping during period n and during all previous periods at all wells. It consists of two parts (Figure 2). One part, $\bar{Z}$, is the unmanaged lift, which is the distance from the land surface down to the transient potentiometric surface that would occur if there was no managed stress applied to the aquifer. The second part, $\hat{Z}$, is the managed lift, which is induced by pumping at all wells during the current period and during all previous periods. Both parts of Z are fully computed by AQMAN. That is,

$$Z_{i,n} = \bar{Z}_{i,n} + \hat{Z}_{i,n} \tag{16a}$$

and

$$\hat{Z}_{i,n} = \sum_{k=1}^{n} \sum_{j=1}^{I} r_{i,j,(n-k)} Q_{j,k} \tag{16b}$$

where $r_{i,j,(n-k)}$ is the drawdown response at well i during period n induced by pumping a unit rate at well j during period k. If j is a recharge well, the response r will be negative.

Substitution of (16) into (15) gives

$$P = \sum_{n=1}^{N} \sum_{i=1}^{I} \left[C_{i,n} \bar{Z}_{i,n} Q_{i,n} + \sum_{k=1}^{n} \sum_{j=1}^{I} C_{i,n} r_{i,j,(n-k)} Q_{j,k} Q_{i,n} \right] \tag{17}$$

↑ cost due to linear component ↑ cost due to quadratic component

The first set of terms is linear, while the second set is quadratic with respect to the decision variables Q.

When the user specifies that the objective is quadratic, AQMAN uses equation (17) as the function to be minimized. For the cost due to unmanaged lift, undiscounted unit cost coefficients $C_{i,n}$ are read as input data and are scaled and discounted for each management period. The value of $\bar{Z}_{i,n}$ during each period is computed, and the scaled product $C_{i,n}\bar{Z}_{i,n}$ is written into the usual objective rows of the MPS file. For each management period, unit costs are discounted according to the

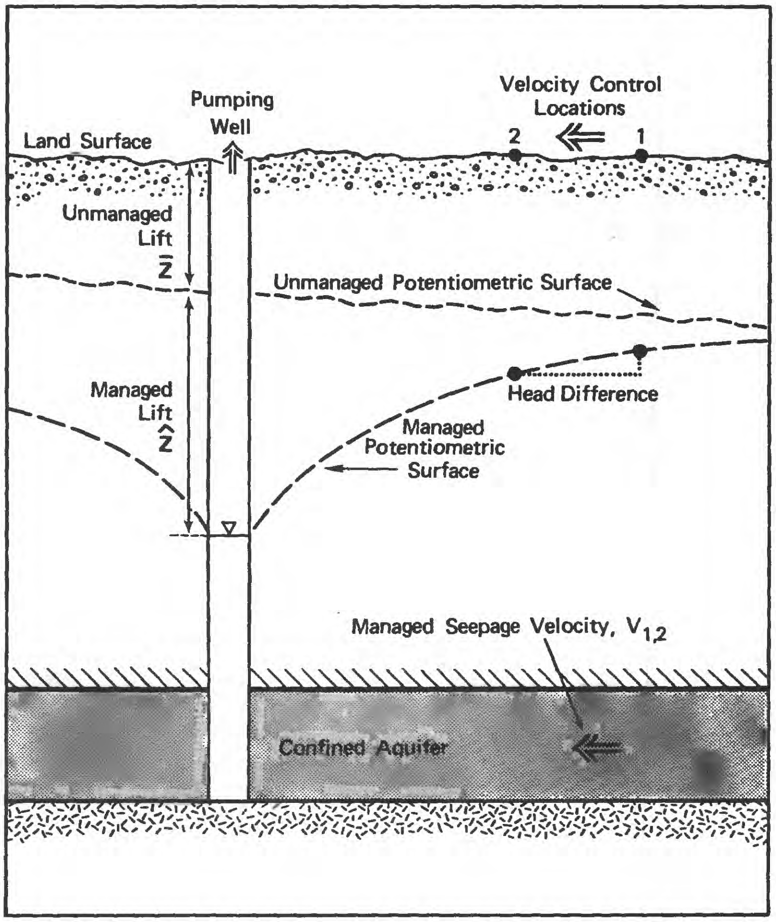

Figure 2. Lift, potentiometric surface, and seepage velocity induced by a pumping well.

number of whole months between the start of the planning horizon and the end of the period:

$$C_{i,n} = \frac{C'_{i,n}}{(1+r)^m} \qquad (18)$$

where r = monthly discount rate [L°] = (annual discount rate/12.0),
m = number of whole months between the start of the planning horizon and the end of period n.

Discounting adjusts future costs to the present, so that costs incurred at different times can be compared directly. Pumping costs are discounted monthly because the electricity required to operate pumps usually must be purchased on a monthly basis. Discounting is performed monthly regardless of the number of days in every management period, as specified by the user. Note that discounting monthly will decrease the present value of future costs slightly more than discounting annually. For instance, if the annual discount rate is 0.08, the first year of pumping cost will be discounted at an effective annual rate of 0.086988.

The quadratic portion of (17) is handled separately. The responses $r_{i,j,(n-k)}$ are written to a new output file. When a quadratic programming code is employed, a special subroutine must be included that reads from this file, discounts unit cost coefficients, and computes the value of the objective function at each iteration of the optimization algorithm. Subroutine FUNOBJ, described in Appendix III, is supplied for this purpose.

Constraint Set

Pumping and recharge constraints

Several types of constraints may be imposed directly on the decision variables. A **bound constraint** limits the value of an individual pumping or recharge decision. For instance,

$$Q_{4,1} \leq 3.0 \qquad (19a)$$

$$Q_{2,2} \geq 1.2 \qquad (19b)$$

$$Q_{3,1} = 0.74 \qquad (19c)$$

This type of constraint must be manually entered into the MPS file (in the BOUNDS or COLUMNS section) by the user. See the section in this document that describes MPS format conventions.

Demand constraints and **capacity constraints** limit the sum of pumping or recharge. For instance, a minimum total demand may have to be met in period 3:

$$\sum_{i=1}^{I} Q_{i,3} \geq 10.5 \qquad (20a)$$

There may be a limit on total pumping capacity during each period:

$$\sum_{i=1}^{I} Q_{i,n} \leq 22.1 \qquad n = 1,...N \qquad (20b)$$

Balance constraints compare pumping flow rates with recharge flow rates. Perhaps a pumping and injection system must be designed so that the total recharge at wells k never exceeds total pumping at wells i:

$$\sum_{i=1}^{I} Q_{i,n} - \sum_{k=1}^{K} Q_{k,n} \geq 0.0 \qquad n = 1,...N \qquad (21)$$

Where I and K are the total number of pumping and recharge wells, respectively.

For each demand, capacity, or balance constraint, the user must create a new row and enter its name into the ROWS section of the MPS file. (See the section below on "the MPS file" for a description of MPS format conventions.) In the COLUMNS section, a coefficient of +1.0 must be entered for each pumping decision that appears in these constraints. For balance constraints such as (21), recharge decisions are given a coefficient of -1.0. Values for the right-hand side of these constraints must be entered in the RHS section of the MPS file.

Head constraints

Hydraulic constraints on aquifer heads, gradients, or velocities are more complicated, since these depend on complex hydrogeologic phenomena and require simulation to obtain the appropriate response coefficients. AQMAN does most of the work, calculating coefficients and transforming the user's constraints into MPS format.

Consider the following problem. Decisions are made during one management period at two pumping wells, Q_1 and Q_2. Head H is constrained at three control locations, and only pumping is allowed:

$$H_1 \leq 70.0 \qquad (22a)$$

$$H_2 \leq 90.0 \qquad (22b)$$

$$H_3 \geq 50.0 \qquad (22c)$$

$$Q_1, Q_2 \geq 0.0 \qquad (22d)$$

The user provides AQMAN with the three control locations, the constraining values 70.0, 90.0, and 50.0 (feet), and the direction of the inequalities. As explained above, pumping constraints such as (22d) are entered directly into the MPS file. (Most optimization codes assume non-negativity for all decision variables unless stated otherwise in the MPS file.) AQMAN performs a series of calculations for the first three constraints:

(1) Compute the transient heads that would occur if there were no managed stresses on the system, that is $Q_1=Q_2=0$. These heads are shown in column 1 of Table 1.

CONTROL LOCATION	(1) UNMANAGED HEAD	(2) USER'S CONSTRAINT ON HEAD	(3) MANAGEABLE DRAWDOWN [(1) - (2)]	HEAD WITH UNIT STRESS APPLIED AT: Well 1 (4a)	Well 2 (4b)	RESPONSE COEFFICIENT Well 1 [(1)-(4a)]	Well 2 [(1)-(4b)]
1	80	≤70	+10	78	76	2	4
2	82	≤90	-8	79	75	3	7
3	75	≥50	+25	69	73	6	2

Table 1. Sample computations of manageable drawdown, and response coefficients. The last two columns form the response matrix.

(2) Subtract the user's constraints from unmanaged heads to obtain manageable drawdowns. See column 3 of Table 1.

(3) Compute the heads that occur in response to a unit rate of pumping at each well. In this example, the unit rate is 1.0 ft^3/sec. Example values of computed heads are shown in Table 1, columns 4a and 4b.

(4) Subtract these heads from unmanaged heads to obtain the drawdown responses. In this example, a unit rate of pumping at well 1 causes drawdowns of 2.0, 3.0, and 6.0 feet, while the same rate at well 2 induces drawdowns of 4.0, 7.0, and 2.0. These values appear in the last two columns of Table 1, and form the response matrix.

AQMAN transforms the first three constraints to:

$$2\ Q_1 + 4\ Q_2 \geq +10.0 \tag{23a}$$

$$3\ Q_1 + 7\ Q_2 \geq -8.0 \tag{23b}$$

$$6\ Q_1 + 2\ Q_2 \leq +25.0 \tag{23c}$$

The constraint on head at each control location is now expressed implicitly as a linear function of the pumping decisions. The non-negativity constraint will be assumed by the linear programming code:

$$Q_1\ ,\ Q_2 \geq 0.0 \tag{23d}$$

The MPS file shown in Figure 3a is written by AQMAN. Note that response coefficients appear in the COLUMNS section and manageable drawdowns appear in the RHS section.

Gradient and velocity constraints

Another type of hydraulic constraint limits the gradient or the seepage velocity between two control locations. Using the above example, the user defines control pair 1 as control locations 1 and 2, and restricts seepage velocity V from location 2 to location 1 to be greater than or equal to 0.01 ft/sec (Figure 2).

$$V_1 \geq 0.01 \tag{24}$$

Velocity is related to the difference in head.

$$V_1 = \frac{K(H_1 - H_2)}{\epsilon\ L} \tag{25a}$$

For the sake of simplicity of presentation, assume that the hydraulic conductivity K and effective porosity ϵ do not vary between the two locations. Head at each location is equal to the difference between unmanaged head H and drawdown s:

```
NAME          EXAMPLE
ROWS
 G  DR010001
 G  DR010002
 L  DR010003
 N  OBJ
COLUMNS
    QP1001    OBJ         1.0
    QP1001    DR010001     0.20000E+01   DR010002     0.30000E+01
    QP1001    DR010003     0.60000E+01
    QP1002    OBJ         1.0
    QP1002    DR010001     0.40000E+01   DR010002     0.70000E+01
    QP1002    DR010003     0.20000E+01
RHS
    RHS       DR010001     0.10000E+02   DR010002    -0.80000E+01
    RHS       DR010003     0.25000E+02
ENDATA
```

Figure 3a. MPS file written by AQMAN for the data shown in Table 1. There are three control locations, two decision wells, and one management period.

```
NAME          EXAMPLE
ROWS
 E  DR010001
 G  DR010002
 L  DR010003
 N  OBJ
COLUMNS
    QP1001    OBJ        1.0
    QP1001    DR010001    0.20000E+01   DR010002    0.30000E+01
    QP1001    DR010003    0.60000E+01
    QP1002    OBJ        1.0
    QP1002    DR010001    0.40000E+01   DR010002    0.70000E+01
    QP1002    DR010003    0.20000E+01
    H010001   DR010001    0.10000E+01
RHS
    RHS       DR010001    0.80000E+02   DR010002   -0.80000E+01
    RHS       DR010003    0.25000E+02
ENDATA
```

Figure 3b. Head definition is substituted at control location 1.

```
NAME          EXAMPLE
ROWS
 E  DR010001
 G  DR010002
 L  DR010003
 L  H010001
 N  OBJ
COLUMNS
    QP1001    OBJ        1.0
    QP1001    DR010001    0.20000E+01    DR010002    0.30000E+01
    QP1001    DR010003    0.60000E+01
    QP1002    OBJ        1.0
    QP1002    DR010001    0.40000E+01    DR010002    0.70000E+01
    QP1002    DR010003    0.20000E+01
    H010001   DR010001    0.10000E+01
    H010001   H010001    0.10000E+01
RHS
    RHS       DR010001    0.80000E+02    DR010002   -0.80000E+01
    RHS       DR010003    0.25000E+02
    RHS       H010001    0.700000E+02
ENDATA
```

Figure 3c. Head constraint at location 1 is manually inserted.

$$H_1 = \bar{H}_1 - s_1 \qquad (25b)$$

$$H_2 = \bar{H}_2 - s_2 \qquad (25c)$$

Substitution of (25) into (24a) gives

$$s_1 - s_2 \leq \bar{H}_1 - \bar{H}_2 - 0.01\ (\epsilon L/K) \qquad (24b)$$

The user supplies three types of input data to AQMAN:

(1) the location of the velocity-control pair,
(2) the minimum (in other problems, maximum) velocity, e.g., 0.01 ft/sec,
(3) the factor ($\epsilon L/K$) that converts the minimum seepage velocity to a difference in head. For this example, we assume ($\epsilon L/K$) equals 90.0 seconds.

AQMAN does the rest. Using the response information from Table 2 and the unmanaged heads from Table 1, the constraint (24b) is transformed for the optimization procedure:

$$\begin{array}{ccccccccc} s_1 & & s_2 & & \bar{H}_1 & & \bar{H}_2 & & (\epsilon L/K) \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ (2\ Q_1 + 4\ Q_2) & - & (3\ Q_1 + 7\ Q_2) & \leq & 80.0 & - & 82.0 & - & (0.01)\ (90.0) \end{array} \qquad (24c)$$

$$-\ Q_1 - 3\ Q_2 \leq -2.9 \qquad (24d)$$

With the addition of the velocity constraint, AQMAN writes the expanded MPS file shown in Figure 4a to include inequality (24d).

Note that if the user wishes to control the hydraulic gradient rather than the velocity, the distance L should be substituted for the factor ($\epsilon L/K$).

A ground-water management problem may involve a plume of contamination. If hydrodynamic dispersion is negligible and linear sorption can be assumed, movement of a contaminant front will be retarded relative to the bulk fluid flow (Freeze and Cherry, 1979). Under these conditions, AQMAN can be used to prescribe pumping and recharge rates that would control movement of the plume (Lefkoff and Gorelick, 1986). The user should incorporate a retardation factor R into the factor ($\epsilon L/K$) that converts seepage velocity to a difference in head. The retarded seepage velocity would then be controlled. In the above example, equation (24b) would become

$$s_1 - s_2 \leq H_1 - H_2 - 0.01\ \frac{\epsilon LR}{K} \qquad (26)$$

where R = retardation factor [L^o].

Head and velocity definitions

A head definition can be used to obtain additional management information in the final solution. Again referring to the above example, the user now wishes to know the optimal head at location 1. The variable KDEFHD is set equal to 1, and

```
NAME          EXAMPLE
ROWS
 L  DIF01001
 G  DR010001
 G  DR010002
 L  DR010003
 N  OBJ
COLUMNS
    QP1001    OBJ         1.0
    QP1001    DIF01001   -0.10000E+01
    QP1001    DR010001    0.20000E+01    DR010002     0.30000E+01
    QP1001    DR010003    0.60000E+01
    QP1002    OBJ         1.0
    QP1002    DIF01001   -0.30000E+01
    QP1002    DR010001    0.40000E+01    DR010002     0.70000E+01
    QP1002    DR010003    0.20000E+01
RHS
    RHS       DIF01001   -0.29000E+01
    RHS       DR010001    0.10000E+02    DR010002    -0.80000E+01
    RHS       DR010003    0.25000E+02
ENDATA
```

Figure 4a. MPS file written by AQMAN for the data shown in Table 1 with velocity control added at control pair 1.

```
NAME          EXAMPLE
ROWS
 E  DIF01001
 E  DIF01002
 G  DR010001
 G  DR010002
 L  DR010003
 N  OBJ
COLUMNS
    QP1001    OBJ       1.0
    QP1001    DIF01001  -0.10000E+01   DIF01002    0.30000E+01
    QP1001    DR010001   0.20000E+01   DR010002    0.30000E+01
    QP1001    DR010003   0.60000E+01
    QP1002    OBJ       1.0
    QP1002    DIF01001  -0.30000E+01   DIF01002   -0.50000E+01
    QP1002    DR010001   0.40000E+01   DR010002    0.70000E+01
    QP1002    DR010003   0.20000E+01
    G01001    DIF01001   0.10000E+01
    G01002    DIF01002   0.10000E+01
RHS
    RHS       DIF01001  -0.20000E+01   DIF01001   -0.70000E+01
    RHS       DR010001   0.10000E+02   DR010002   -0.80000E+01
    RHS       DR010003   0.25000E+02
ENDATA
```

Figure 4b. Head definition is specified at velocity control pair 1 and 2.

```
NAME          EXAMPLE
ROWS
 E  DIF01001
 E  DIF01002
 G  DR010001
 G  DR010002
 L  DR010003
 G  G01001
 G  G01002
 N  OBJ
COLUMNS
    QP1001    OBJ       1.0
    QP1001    DIF01001  -0.10000E+01   DIF01002    0.30000E+01
    QP1001    DR010001   0.20000E+01   DR010002    0.30000E+01
    QP1001    DR010003   0.60000E+01
    QP1002    OBJ       1.0
    QP1002    DIF01001  -0.30000E+01   DIF01002   -0.50000E+01
    QP1002    DR010001   0.40000E+01   DR010002    0.70000E+01
    QP1002    DR010003   0.20000E+01
    G01001    DIF01001   0.10000E+01   G01001      0.10000E+01
    G01001    G01002     0.57700E+00
    G01002    DIF01002   0.10000E+01   G01002     -0.10000E+01
RHS
    RHS       DIF01001  -0.20000E+01   DIF01001   -0.70000E+01
    RHS       DR010001   0.10000E+02   DR010002   -0.80000E+01
    RHS       DR010003   0.25000E+02
    RHS       G01001     0.90000E+00   G01002      0.00000E+00
ENDATA
```

Figure 4c. Velocity constraint at control pair 1 is manually inserted.

AQMAN will substitute

$$2Q_1 + 4Q_2 + H_1 = 80.0 \qquad (27)$$

for inequality (23a). The right side of this equation is the unmanaged head shown in column 1 of Table 1. AQMAN now writes the MPS file shown in Figure 3b rather than 3a. A value of H_1 will now appear in the management solution. In order to include the original constraint (22a), the user manually adds a row, a column entry, and a right-hand side into the MPS file. This is shown in Figure 3c. (See the section below on MPS formats.)

A velocity definition can be used to control the difference between two velocities. For example, the user defines control pair 2 as control locations 2 and 3, and restricts the vector sum of seepage velocities V_1 and V_2 to remain within 30° of the direction of V_1 (Figure 5). That is

$$\tan 30° \geq \frac{V_2}{V_1} \qquad (28a)$$

$$0.577\, V_1 - V_2 \geq 0 \qquad (28b)$$

The user now supplies as input data to AQMAN only the locations of the two velocities control pairs, with KDEFGR set equal to 1. The following equations replace (24b):

$$s_1 - s_2 + V_1 = \bar{H}_1 - \bar{H}_2 \qquad (29a)$$

$$s_3 - s_2 + V_2 = \bar{H}_3 - \bar{H}_2 \qquad (29b)$$

AQMAN now writes the MPS file shown in Figure 4b rather than 4a. In order to include the original constraint (24a) and the new constraint (28b), the user manually adds two rows, three column entries, and two right-hand sides into the MPS file. This is shown in Figure 4c.

Nonlinearities

AQMAN uses the Trescott code to solve the ground-water flow equation (1), which contains a linear relation between heads and well pumpages. The user may introduce some nonlinearities into the simulated system by specifying unconfined conditions or head-dependent leakage or evapotranspiration (Trescott and others, 1976, pp. 2-8). In these cases, AQMAN will write a warning message, but otherwise will operate normally, as if the system behaved linearly. The user must judge whether the consequent error in the linear programming solution is acceptable. For instance, if an unconfined aquifer is thick and highly conductive, transmissivity will not be a strong function of head, and linear treatment of the system may provide an acceptable management solution.

Time Parameters

The user must also select appropriate time parameters for the problem. Pumping periods in the Trescott code are synonymous with management periods in AQMAN, so

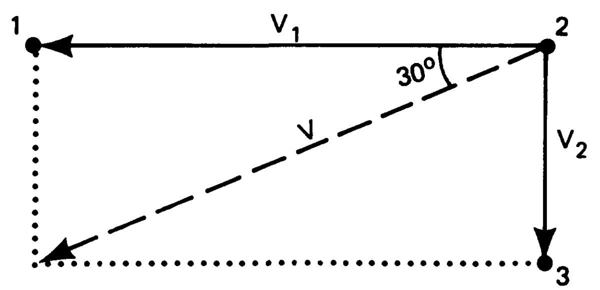

Figure 5. The vector sum V of velocity controls 1 and 2.

that management decisions are made for each pumping period specified. Hydraulic responses to applied stresses are written to the MPS file for the time corresponding to the end of each period. The response matrix formulation that AQMAN uses requires that **all pumping periods have the same length.** This assures that the time difference between implementation and effect of a decision is independent of pumping period. For instance, a decision for period 4 will cause an additional drawdown during period 6. The same decision in period 1 will cause the same additional drawdown during period 3. Transient responses are added over time. This fully utilizes the computational advantages of linear programming with response matrices.

For transient simulations, finite-difference computations are performed by time step within each pumping period. This is the numerical time step used in the finite-difference approximation, and should not be confused with the length of pumping periods. The numerical time step may have a variable length. Short time steps should be used while heads are rapidly recovering from the unit stress applied during the first period. Longer time steps may be used for later pumping periods in order to save computer time. The length of the initial time step within each period is read as the variable TIMINC. Subsequent time steps within a period are increased by a multiplying factor, CDELT. Time steps are automatically adjusted so that the end of each pumping period corresponds to the end of a time step.

The Unit Stress and Scaling

The value of the unit stress is read by AQMAN as the variable UNITQ. Its sign indicates whether the stress is a (-) sink (pumping) or a (+) source (recharge). AQMAN input follows the convention used in the Trescott code: negative for pumping, positive for recharge. AQMAN converts signs so that pumping and recharge rates are positive in the MPS file.

Many optimization codes automatically constrain decision variables to non-negative values. This is useful if each well is restricted to either pumping or recharge. In the input files for AQMAN, the user should specify a uniform positive unit stress at recharge wells and an equal negative stress at pumping wells. The management solution obtained with the optimization code will contain positive rates for both recharge and pumping.

Rather than place a restriction on the decision wells, the user may want to let the optimization code determine whether each well should pump or recharge. To accomplish this, assign a positive unit stress for all wells in AQMAN. Before executing the optimization code, override the automatic non-negativity constraints. (This is usually done in a specification file that gives general instructions to the optimization code.) The management solution will contain negative values that represent pumping rates, and positive values that indicate recharge rates.

Usually, all unit stresses are equal. A uniform, convenient unit stress, such as 1.0 ft^3/s, simplifies interpretation of the optimization results. However, in some cases this may cause a scaling problem, where the range in magnitude of the response coefficients is large. If the coefficients vary over several orders of magnitude, the optimization code may not be able to compute an optimal solution. If this occurs, the user will have to scale the formulation by specifying a different unit stress (UNITQ in the AQMAN input file) at certain decision wells. After re-executing AQMAN, the objective function coefficients in the new MPS file will be automatically adjusted to consistent units. Finally, the management solution computed by the optimization code must be scaled back to the original units.

For example, consider a problem containing two decision wells. The objective is to minimize the sum of the pumping rates. AQMAN is used with a uniform unit stress of -1.0 ft^3/s to create an MPS file. The file is used by a linear programming package, which reports an error message that relates to poor scaling. Examination of the MPS file reveals that well A has a very large effect on drawdowns, while well B has a very small effect. AQMAN is used to create a new MPS file, this time applying a unit stress of -0.001 ft^3/s at well A and -1.0 ft^3/s at well B. This means that every unit of pumping at well A represents only 1/1000 of each unit at well B. After re-executing AQMAN, the coefficient for well A in the objective row of the new MPS file will be changed to 0.001, while the coefficient for B will be left at 1.0. Say that the optimal solution to the scaled problem specifies pumping rates of 240 units at well A and 3.4 units at well B. This translates to 0.24 ft^3/s and 3.4 ft^3/s, respectively.

Quadratic Objective

If the objective function is the minimization of variable pumping costs, set the variable CASE equal to 'QUAD' in the AQMAN input file. Another output file, the quadratic response file, will be created. This file will be used by a subroutine in the quadratic programming package that evaluates the value and gradient of the quadratic objective for a given set of decisions. This subroutine must be supplied by the user. Two suggested versions, named Subroutine FUNOBJ, can be found in Appendix III.

A quadratic objective function will have both linear and quadratic terms, as in (17). Coefficients for the linear terms are written by AQMAN into the objective row of the MPS file. During execution with the optimization package, coefficients for the quadratic terms will be computed by FUNOBJ using input data read from the quadratic response file and from an additional input file. This additional file is constructed by the user, and contains information on well locations, unit pumping costs, and the discount rate. Instructions for preparing this file appear in Appendix III. Appendix III also describes how to specify pumping wells for AQMAN when the objective is quadratic.

The MPS File

User changes

Before using an optimization package, the user may want to make changes in the MPS file written by AQMAN. The most common changes will involve the addition of pumping and recharge constraints or changes in linear objective function coefficients. Examples are presented in (14), (19), and (20). Before editing the MPS file, the user should be familiar with MPS formats and the naming conventions used by AQMAN.

MPS format conventions

This section presents a basic description of the MPS format. More complete discussions can be found elsewhere (IBM Document No. H20-0476-2; Murtagh and Saunders, 1983).

Figure 6 shows the organization of the MPS file. The first line of the file must say "NAME" and the last line must say "ENDATA". The rest of the file is organized into sections. Each section is introduced by a line that contains one of the following words, starting in the first column: "ROWS", "COLUMNS", "RHS", "RANGES", "BOUNDS". The sections consist of data lines that either identify constraints or specify the numerical values that appear in the management problem. AQMAN writes all lines that are not optional.

The ROWS section names and defines the type of each constraint and names the objective function. Each line contains:

(1) A row name, in columns 5 through 12. The row names identify each constraint and the objective.

(2) A key to indicate the type of the row, in either column 2 or 3. There are four types of rows:

key	row type
G	greater than or equal to, $\geq$
L	less than or equal to, $\leq$
E	equality, =
N	unconstrained. (used for objective)

The key N is required for the objective row. It is also useful for providing head and velocity information at uncontrolled locations.

The COLUMNS section names the decision variables, and specifies the response coefficients and cost coefficients. Each coefficient links a decision variable (column) to either a constraint (row) or to the objective (row). Each line contains:

(1) A column name, in columns 5 through 12. The column names are the decision variables.

(2) A numerical response coefficient, in columns 25 through 36. The name of the row in which the coefficient appears for the decision variable named on this line is entered in columns 15 through 22.

(3) Optionally, another response coefficient and row name, respectively in columns 50 through 61 and columns 40 through 47. AQMAN uses this option in order to minimize the length of the file.

The RHS section specifies the values for the right-hand side of each constraint row. Each line contains:

(1) A right-hand side data set name, in columns 5 through 12. Usually, there is only one set of right-hand sides, in which case this name would be the same for all lines.

(2) The numerical value of a right-hand side, in columns 25 through 36, and the

```
NAME
ROWS
   :
   (Row Data)
   :
COLUMNS
   :
   (Column Data)
   :
RHS
   :
   (RHS Data)
   :
RANGES              \
   :                 |
   (Ranges Data)     |
   :                 |
BOUNDS               > Optional
   :                 |
   (Bounds Data)     |
   :                /
ENDATA
```

Figure 6. Organization of the MPS file.

name of its corresponding row, in columns 15 through 22.

(3) Optionally, the numerical value of another right-hand side and its corresponding row, respectively in columns 50 through 61 and columns 40 through 47. AQMAN uses this option in order to minimize the length of the file.

The RANGES section can be used during sensitivity analysis to allow the right-hand side of a constraint to range between upper and lower limits. This section is not required by most optimization packages, and is not written by AQMAN. If ranges for the constraints are desired, the user must manually enter the entire section.

A range for a constraint is defined by its RANGE value r, its RHS value c, and its row type. The sign of r should be positive. There are two possibilities:

Row type	Upper limit	Lower limit
G	c + r	c
L	c	c - r

Each line in the RANGES section contains:

(1) The range data set name, in columns 5 through 12. Usually, there is only one set of ranges, in which case this name would be the same for all lines.
(2) The numerical value of a range, in columns 25 through 36, and the name of the row in which it appears, in columns 15 through 22.
(3) Optionally, the numerical value of another range and its corresponding row, respectively in columns 50 through 61 and columns 40 through 47.

The BOUNDS section is used to specify upper and/or lower limits directly on the value of individual decision variables. This section is not required by most optimization packages, and is not written by AQMAN. If bounds on the decision variables are desired, the user must manually enter the entire section. (Many optimization packages automatically restrict decision variables to non-negative values.)

Each line in the BOUNDS section contains: (1) The type of bound, in columns 2 through 3. Six types of bounds are possible:

bound type	explanation
LO	lower bound
UP	upper bound
FX	fixed value
FR	free (-infinity to +infinity)
MI	lower bound = -infinity
PL	upper bound = +infinity

(2) The BOUNDS data set name, in columns 5 through 12. Usually, there is only one set of bounds, in which case this name would be the same for all lines.

(3) The numerical value of a bound, in columns 25 through 36, and the name of the decision variable (COLUMN name) to which it applies, in columns 15 through 22.

<u>Naming conventions</u>

There are three different types of rows named by AQMAN. The row for the objective function is simply named "OBJ". Row names that correspond to constraints on head begin with the letters "DR", followed by a 6-digit number. The last four digits identify the control location, and the first two digits refer to the management period. For example, DR030026 indicates:

```
                management
                  period
                    ↓
        DR          03       0026
         ↑                    ↑
        head               control
     (drawdown)            location
     constraint
```

Row names for constraints on a gradient or a velocity begin with the letters "DIF", followed by a 5-digit number. The last three digits identify the control pair, and the first two digits identify the management period. For example, DIF16002 indicates:

```
                management
                  period
                    ↓
        DIF         16      002
         ↑                   ↑
      gradient            control
         or                 pair
      velocity
     constraint
```

Appendix IV shows a sample MPS file written by AQMAN using these names. Note that these naming conventions for the rows restrict the number of control locations to less than 10,000, the number of control pairs to less than 1000, and the number of management periods to less than 100.

Column names begin with the letter "Q", followed by a 5-digit number. The last three digits identify the decision well, and the first two digits denote the management period. For example,

```
          management
            period
              ↓
    Q         01      015
    ↑                  ↑
decision              well
variable             number
```

The naming convention for columns restricts the number of decision wells to less than 1000, and the number of management periods to less than 100.

Column names for head definitions are identical to the corresponding row name, except that "DR" is replaced by "H" and "DIF" is replaced by "G". For example, if head definition is specified for location 119 during period 2, the row name would be DR020119 and the column name would be H020119.

Size of the MPS file

The MPS file written by AQMAN may be quite large. The approximate number of lines in the file is given by:

$$7 + MN + M'N + D'N + I\left[N + \frac{M'N\,(N+1)}{2}\right] \qquad (25)$$

where I = number of decision wells,

N = number of management periods,

M = number of control locations plus control pairs,

M'= M/2, rounded up to the nearest whole number,

D = number of head, gradient, and velocity definitions,

D'= D/2, rounded up to the nearest whole number.

The first term accounts for the seven required identification lines. The second, third, and fourth terms are the number of lines in the ROWS section, the RHS section, and the COLUMNS section, respectively.

For example, say that AQMAN is given a problem that contains 12 decision wells, 20 locations where head is controlled, 41 control pairs where velocity is controlled, and 4 management periods. The MPS file will contain approximately the following number of lines:

$$7 + (61)(4) + (31)(4) + (12)\,[4 + (31)(4)(5)/2] = 4143$$

PROGRAM DESCRIPTION

AQMAN consists of several subroutines combined with a modified version of the U.S. Geological Survey's two-dimensional, finite-difference aquifer model (Trescott and others, 1976). AQMAN uses the Trescott code as a subroutine (named TRES) to obtain hydraulic responses to management alternatives. The Trescott code has been modified for compatability with the AQMAN subroutines. AQMAN subroutines read and write data, perform simple calculations, and make repeated calls of Subroutine TRES to obtain results of aquifer simulation. The functions performed by each subroutine are described below. Program variables appear in capital letters and are defined in Appendix I.

Main

The main program makes several calls of Subroutine PRE, and performs simple calculations after each call. This is shown in Figure 7. Subroutine PRE is called once to read input files and to calculate transient, unmanaged heads, given initial and boundary conditions and any pre-determined pumping or recharge at unmanaged wells. Input data are checked for consistency, and unmanaged (unstressed) heads are stored in the HDUS vector and are written to unit 17. The user's desired limit on head, CONHD, at each control location is read from unit 14. These values are subtracted from the unstressed heads to obtain a vector of manageable drawdowns, MNGDRD, which will be written in the RHS section of MPS file.

If the objective function is quadratic, Subroutine QUAD is called to compute the cost coefficients for the linear part of the objective. If there are any gradients or velocities to be controlled, Subroutine GRADS is called to identify locations and to compute values for constraints on these controls.

Next, four tasks are performed for each managed well. First, Subroutine PRE is called to obtain hydraulic responses at all control locations to a unit stress applied at the well. Second, the head responses are converted to drawdown responses, DRDRES, by subtracting from unmanaged head at each control location. Third, if there are gradients or velocities to be managed, Subroutine GRADS is called to compute RDIF, which is the difference in drawdown response within a control pair. Fourth, Subroutine MPSFMT is called to write the response coefficients that correspond to this managed well into the COLUMNS section of the MPS file, unit 18. During the initial call of MPSFMT, the ROWS section of the MPS file is written.

Subroutine MPSFMT is called a final time to write the RHS section of the MPS file. Program execution then terminates.

Subroutine PRE

PRE controls simulations by calling the Trescott code as Subroutine TRES. Before each call, PRE assigns values to the vector of pumpages (variable QWELL), which is passed to TRES through the "AQMAN1" common block. PRE also initializes the pumping period counter (variable KPER) to 1. In addition, during the first call of PRE, data describing the management problem are read from unit 14. These include information on the type of problem and objective, the managed wells, the control locations, and time parameters.

After control returns to PRE from Subroutine TRES, the head vector (variable

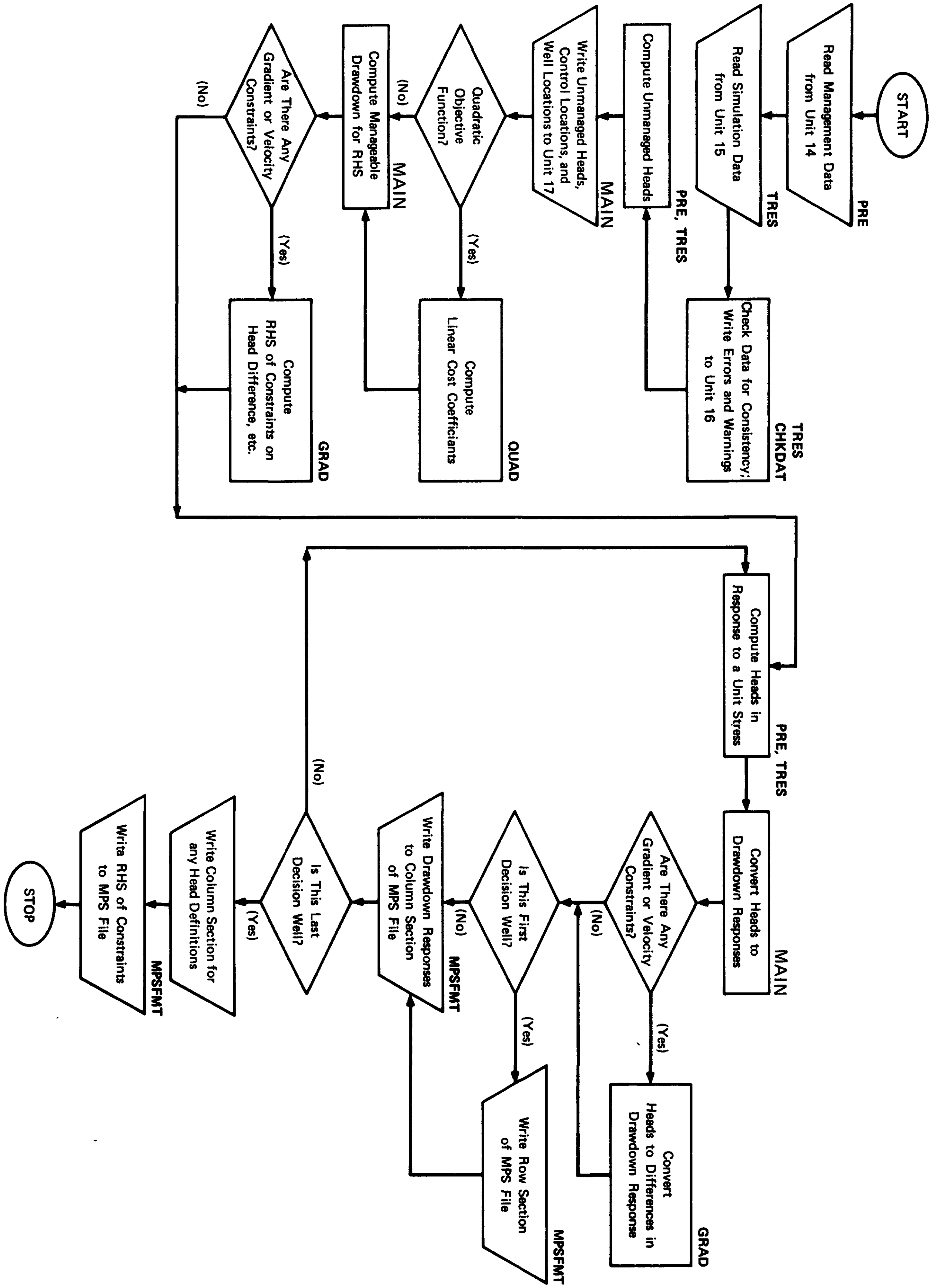

Figure 7. Flow chart showing major computations performed by AQMAN.

HD) is equated to the XHEAD vector. XHEAD contains solved heads at all control locations, and is stored by Subroutine WRITE1 in the "AQMAN1" common block after all Trescott computations. The HD vector is a subroutine argument that passes the computed heads to the Main program.

Subroutine CHKDAT

This subroutine checks input data for internal consistency. First, all control locations where KEYWL is not equal to zero are checked. The number of non-zero KEYWLs must equal NW, which is read from the usual Trescott data set on unit 15. NW is the number of well locations where head at the well, rather than average head for the finite-difference cell, is desired. Two conditions must hold for each location where KEYWL is not zero: (1) a well must exist (as specified by ILOCW and JLOCW) at that control location, and (2) the radius of the well (XRAD) must be positive, so that hydraulic head at the well radius will be computed correctly. If an error in the input data is found here, CHKDAT writes an error message to unit 16 and stops program execution.

CHKDAT also checks for nonlinearities introduced by unconfined conditions, evapotranspiration, or leakage. If any exist, a warning message is written to unit 16.

Subroutine GRADS

This subroutine is called if gradients or velocities are to be controlled. It reads information from unit 13 on each gradient or velocity control pair. These data are checked for consistency with input from unit 14. The locations specified as control pairs in unit 13 must also have been identified as "primary" control locations in unit 14. If not, GRADS writes an error message to unit 16 and stops execution.

GRADS computes the manageable drawdown for each control pair. This is the difference between the unstressed heads at the two control points, minus the product of GFACT and GCON. GCON is the user's desired limit on gradient or velocity, and GFACT is the factor that converts a gradient or a velocity into a head difference. (See equations 11, 12, and 24.) The difference in drawdown response (RDIF) between the two points of each control pair is also computed.

Subroutine MPSFMT

This subroutine writes all output in MPS format to the MPS file, unit 18. A row name for each constraint during every management period is written into the ROWS section. For each managed well, drawdown responses at control locations are converted to MPS input format, and are written in the COLUMNS section of the MPS file. For each constraint, the right-hand side values are written into the RHS section in MPS format. If the problem has a quadratic objective function, MPSFMT writes to unit 19 the drawdown responses at all pumping managed wells. These are identified as control locations at which KEYWL=1 and KEYGRD=1. Unit 19 is read by the optimization code in order to compute the value and gradient of the quadratic objective function for a given set of decisions (see Appendix III).

Subroutine QUAD

This subroutine is called only if the objective function is quadratic, indicated when the input variable CASE equals "QUAD". It reads from unit 13 the land surface elevation at each managed well, the annual discount rate, and the undiscounted unit pumping costs for each managed well during each period. For each management period, unit costs are discounted according to equation (18).

At each managed pumping well, QUAD computes pumping lifts under unstressed conditions. These are multiplied by the discounted unit pumping cost to obtain the cost coefficient for the linear part of the quadratic objective function. This will be written in the OBJ rows of the COLUMNS section in the MPS file by Subroutine MPSFMT. No cost coefficients are computed for recharge wells or for unmanaged wells.

Subroutine READ1

At the start of each management (pumping) period, this subroutine retrieves input data stored in common blocks and prepares them for transfer to Subroutine TRES. These data include information on the management period, well locations, well radii, and well pumpages.

Subroutine WRITE1

At the end of each management period, this subroutine enters the computed head at every control location into a vector XHEAD, which is then stored in the "AQMAN1" common block. Each of these heads is an average value for a cell in the finite-difference grid.

If pumping or recharge occurs in the cell, head at the well may be significantly different than the average cell value. The Trescott code uses the Thiem equation to compute head at the well radius (Trescott and others, 1976, pp.8-10). Subroutine WRITE1 tests each element of the XHEAD vector for whether the variable KEYWL equals zero. If not, head at the well radius is indicated, and this head is entered into XHEAD, replacing the average head previously stored. This does not affect future finite-difference calculations, as the average head in every cell is used at the start of the next management period.

Data Files

The purpose of each input and output file used by AQMAN is summarized below.

File Unit Number	Type	Description
13	Input	Management data for gradient/velocity controls and quadratic objective.
14	Input	Management data for all wells, control locations, and time periods.
15	Input	Hydrogeologic data for aquifer simulation (The usual Trescott data set).
16	Output	Error and warning messages.
17	Output	Unmanaged heads, user-defined limit stresses.
18	Output	The MPS file.

SAMPLE PROBLEM

In this section a simple example of the use of AQMAN is presented. The hypothetical aquifer to be managed is shown in Figure 8, and is similar to the sample problem of Trescott and others (1976). Modifications include the addition of a third pumping well and the relocation of the four recharge wells. The aquifer is confined, and the constant head boundary occurs at an undersea outcrop offshore.

Suppose that the following management criteria are specified. The objective is to maximize net water supply during two one-month management periods from the two wells located at cells (4,11) and (6,6). The third pumping well, located at cell (3,7), must pump at a fixed rate of 0.005 ft^3/s and 0.10 ft^3/s during the first and second months, respectively. At least a 1.0 percent southward gradient is to be maintained in order to prevent intrusion of salt water from the sea. Heads in the stippled area to the northeast must remain above 80.0 feet. Pumping at cell (6,6) must not pull head below the top of the aquifer, which has an elevation of 50.0 feet. The recharge wells must not raise heads above the height of the land surface, 250.0 feet. Finally, all recharge water must be supplied by the pumping wells, so total recharge can never exceed total pumpage.

The first step in formulating this problem mathematically is to identify the decision variables. These are pumping or recharge at six wells during two management periods. Denote these as $Q_{i,n}$, where i is the well index and n is the period index. Let i equal 1 to 6 for cells (4,11), (6,6), (7,8), (7,9), (7,10), and (7,11). Let n equal 1 to 2 for the first and second month. Restrict all decision variables to non-negative values; wells 1 and 2 can only pump, and wells 3 through 6 can only recharge.

The next step is to identify locations where head must be controlled. Denote head values as $H_{k,n}$, where k is the location index. To maintain southward gradients along the coast, heads must be controlled at four pairs of cells along the coast, as shown in Figure 8. In addition to these eight locations, head is controlled in the stippled area, at well 2, and at all four recharge wells. Let k equal 1 to 17 for cells (3,11), (3,12), (4,11), (4,12), (6,6), (7,8), (7,9), (7,10), (7,11), (8,8), (8,9), (8,10), (8,11), (9,8), (9,9), (9,10), and (9,11). Let L denote the distance (28.5 feet) between nodes in row 8 and nodes in row 9.

The user's linear programming formulation can now be stated:

$$\text{Maximize} \quad Q_{1,1} + Q_{1,2} + Q_{2,1} + Q_{2,2} - Q_{3,1} - Q_{3,2} - Q_{4,1} - Q_{4,2} - Q_{5,1} - Q_{5,2} - Q_{6,1} - Q_{6,2} \tag{31}$$

Subject to the constraints:

$$H_{1,n} \geq 80.0 \tag{32}$$

$$H_{2,n} \geq 80.0 \tag{33}$$

$$H_{3,n} \geq 80.0 \tag{34}$$

$$H_{4,n} \geq 80.0 \tag{35}$$

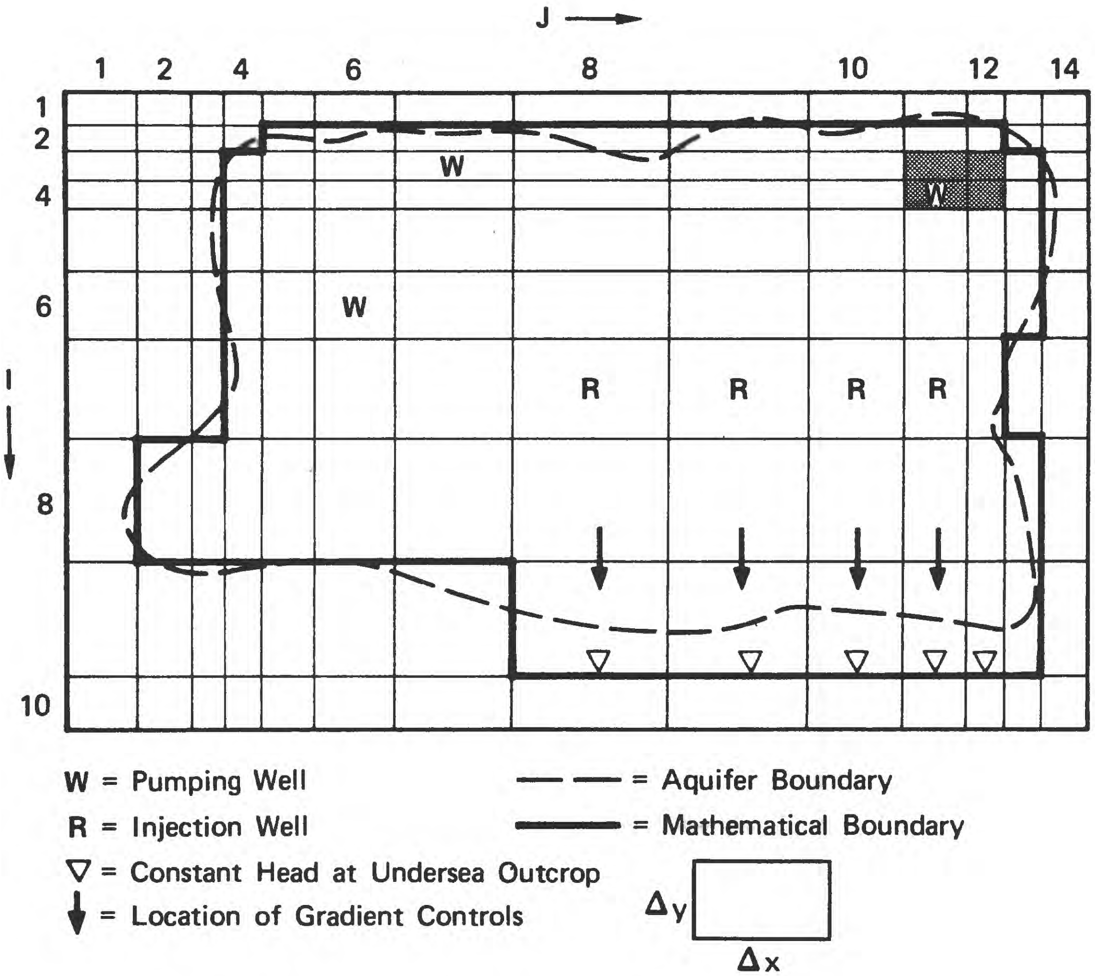

Figure 8. Finite-difference grid for the sample problem (after Trescott and others, 1976).

$$H_{5,n} \geq 50.0 \quad (36)$$

$$H_{6,n} \leq 250.0 \quad (37)$$

$$H_{7,n} \leq 250.0 \quad (38)$$

$$H_{8,n} \leq 250.0 \quad (39)$$

$$H_{9,n} \leq 250.0 \quad (40)$$

$$(H_{10,n} - H_{14,n})/L \geq 0.01 \quad (41)$$

$$(H_{11,n} - H_{15,n})/L \geq 0.01 \quad (42)$$

$$(H_{12,n} - H_{16,n})/L \geq 0.01 \quad (43)$$

$$(H_{13,n} - H_{17,n})/L \geq 0.01 \quad (44)$$

$$Q_{1,n} + Q_{2,n} - Q_{3,n} - Q_{4,n} - Q_{5,n} - Q_{6,n} \geq 0.0 \quad (45)$$

$$Q_{i,n} \geq 0.0 \; , \qquad \text{for } i = 1,...6 \quad (46)$$

All constraints must hold for n equals 1 and 2. Inequalities (32) through (40) are head constraints, (41) through (44) are gradient constraints, (45) is a balance constraint, and (46) is a non-negativity constraint.

At this point, AQMAN can be utilized to generate the response matrix needed for the problem and to create the MPS file required by standard optimization codes. Three input files, shown in Appendix IV, are needed to use AQMAN. Unit 15 is the original data set shown as Figure 33 in Trescott and others (1976). "Group IV" data will be ignored by AQMAN. In order to simulate a strictly linear system, unconfined conditions, leakage, and evapotranspiration have been omitted. (If these were not omitted, AQMAN would write warning messages to unit 16 but otherwise would operate normally). The value of NW has been changed to 6, since head at a well radius is managed at 6 locations. In order to demonstrate a transient problem, the number of pumping periods has been increased to two and a uniform storativity of 0.001 has been entered for all active nodes.

Unit 14 is the first of two data sets containing management information. (See Appendix II for a description of unit 14 formats.) The second line states that there are 7 wells, 17 control locations, 2 pumping periods, and 4 gradient control pairs. The fifth line specifies that one of the wells is an unmanaged well, operating at a pre-determined rate. This well is identified in the sixth line, with KEYQ equal to 1. The next two lines specify that this well must pump 0.005 ft^3/s during the first period and 0.01 ft^3/s during the second period. The two pumping decision wells are identified on lines 9 and 10. A unit stress of -1.0 ft^3/s will be applied at each pumping well in order to obtain responses at control locations. Lines 11 through 14 indicate that a unit stress of +1.0 ft^3/s will be applied at the four recharge wells. The 17 control locations are identified in lines 15 through 31. At the six control locations that require head of the well radius, KEYWL equals 1. Eight control locations involve only gradient control, so KEYGRD equals 1 at each of these. The next 17 lines specify the minimum or maximum head allowed at each of the control locations during the first period; these are followed by the same restrictions for the second period. Within each set of 17 lines, the first 9 lines specify the right-hand side and the direction of the inequalities for constraints (32) through (40). The remaining 8 lines correspond to control locations 10 through 17. Since heads are not controlled at these locations (gradient controls only), the values specified in these 8 lines are ignored. However, these "dummy" lines must be entered into the file.

Unit 13 contains information for the gradient constraints, (41) through (44). (See Appendix II for a description of unit 13 formats.) The first 4 lines define the gradient control pairs, using control locations that have already been defined in unit 14. The next 4 lines specify the minimum gradients allowed at each of the control pairs during the first period; these are followed by the same restrictions for the second period. Within each set of 4 lines, the variable GCON specifies the right-hand side of the constraints. The variable GFACT equals the distance between the two nodes of each control pair. It converts GCON to a difference in head, as in equation (11).

AQMAN was run with these three input files. Before examining the output, it is instructive to see how AQMAN internally re-formulates the problem before it constructs the MPS file. The re-formulated problem is shown in matrix form in Figure 9. Heads and gradients are converted to drawdown responses, and the user's limits on heads and gradients are converted to manageable drawdowns.

The output from AQMAN is also shown in Appendix IV. Unit 16 contains no messages.

Unit 17 provides intermediate results of the simulation. Unstressed heads may be useful in interpreting the hydrologic significance of the final management solution. The user's desired limits on head are provided as a check against input data.

Unit 18 is the MPS file. Before using it in an optimization model for this problem, several modifications are necessary in order to incorporate the balance constraint (45). These changes are all indicated with bold type in the modified MPS file, also shown in Appendix IV. One change is needed in the ROWS section. The two constraints on pumping and recharge balance must be named and identified as "greater than or equal to". In this case, the user chooses the names BAL1 and BAL2. In the COLUMNS section, for all 12 decision variables, a coefficient of +1.0 (pumping) or -1.0 (recharge) is added for the constraints on pumping and recharge balance. Finally, the right-hand sides of the two balance constraints are added to the RHS section.

The modified MPS file was used as input for an optimization code, MINOS (Murtaugh and Saunders, 1983). The results are shown in Appendix IV. The optimal solution is to pump wells 1 and 2 at rates of 0.415 and 0.972 ft^3/s during the first period, at rates of 0.408 and 1.166 ft^3/s during the second period, and to recharge well 3 at a rate of 0.308 ft^3/s during the second period. This solution satisfies all the constraints and yields a net water supply of 2.65 ft^3/s.

Maximize:

$$Q_{1,1} + Q_{2,1} - Q_{3,1} - Q_{4,1} - Q_{5,1} - Q_{6,1} + Q_{1,2} + Q_{2,2} - Q_{3,2} - Q_{4,2} - Q_{5,2} - Q_{6,2}$$

Subject to:

8.2	19.4	-32.0	-17.4	-10.4	-7.8	0.0	0.0	0.0	0.0	0.0	0.0		≤	22.2
13.4	8.6	-17.2	-28.7	-19.3	-14.7	0.0	0.0	0.0	0.0	0.0	0.0	$Q_{1,1}$	≤	18.2
17.5	4.5	-9.6	-18.7	-29.0	-24.1	0.0	0.0	0.0	0.0	0.0	0.0		≤	15.3
15.7	2.6	-5.8	-11.7	-20.4	-26.2	0.0	0.0	0.0	0.0	0.0	0.0	$Q_{2,1}$	≤	11.5
179.9	6.7	-14.0	-26.3	-39.1	-45.3	0.0	0.0	0.0	0.0	0.0	0.0		≤	156.5
159.6	6.0	-13.0	-25.1	-39.1	-47.6	0.0	0.0	0.0	0.0	0.0	0.0	$Q_{3,1}$	≤	156.5
10.0	62.0	-29.0	-15.8	-9.7	-7.3	0.0	0.0	0.0	0.0	0.0	0.0		≤	159.7
155.6	6.0	-12.9	-25.2	-39.9	-49.3	0.0	0.0	0.0	0.0	0.0	0.0	$Q_{4,1}$	≤	155.8
431.9	6.7	-14.0	-26.6	-40.1	-47.1	0.0	0.0	0.0	0.0	0.0	0.0		≤	185.9
6.7	387.2	-27.4	-13.2	-7.6	-5.6	0.0	0.0	0.0	0.0	0.0	0.0	$Q_{5,1}$	≥	-7.3
14.0	27.4	-210.7	-28.6	-16.5	-12.3	0.0	0.0	0.0	0.0	0.0	0.0		≥	-18.7
26.6	13.2	-28.6	-209.9	-35.4	-25.9	0.0	0.0	0.0	0.0	0.0	0.0	$Q_{6,1}$	≥	-21.8
40.1	7.6	-16.5	-35.4	-214.6	-51.2	0.0	0.0	0.0	0.0	0.0	0.0		≥	-23.9
7.7	14.8	-9.2	-7.0	-5.7	-5.1	8.2	19.4	-32.0	-17.4	-10.4	-7.8	$Q_{1,2}$	≤	33.8
8.1	8.6	-6.3	-5.8	-5.3	-5.0	13.4	8.6	-17.2	-28.7	-19.3	-14.7		≤	26.2
7.9	5.6	-4.7	-4.9	-4.9	-4.8	17.5	4.5	-9.6	-18.7	-29.0	-24.1	$Q_{2,2}$	≤	21.5
6.3	3.7	-3.3	-3.6	-3.7	-3.7	15.7	2.6	-5.8	-11.7	-20.4	-26.2		≤	15.9
28.9	12.0	-12.2	-14.6	-15.9	-16.2	179.9	6.7	-14.0	-26.3	-39.1	-45.3	$Q_{3,2}$	≤	173.7
29.7	11.4	-12.0	-14.6	-16.1	-16.5	159.6	6.0	-13.0	-25.1	-39.1	-47.6		≤	173.5
13.8	37.5	-20.7	-14.3	-10.9	-9.4	10.0	62.0	-29.0	-15.8	-9.7	-7.3	$Q_{4,2}$	≤	182.7
28.9	11.2	-11.7	-14.2	-15.7	-16.1	155.6	6.0	-12.9	-25.2	-39.9	-49.3		≤	172.5
28.1	11.8	-12.0	-14.2	-15.5	-15.8	431.9	6.7	-14.0	-26.6	-40.1	-47.1	$Q_{5,2}$	≤	202.6
11.8	39.6	-20.6	-13.5	-9.8	-8.3	6.7	387.2	-27.4	-13.2	-7.6	-5.6		≥	19.3
12.0	20.6	-13.1	-10.3	-8.6	-7.7	14.0	27.4	-210.7	-28.6	-16.5	-12.3	$Q_{6,2}$	≥	-2.2
14.2	13.5	-10.3	-9.7	-9.1	-8.7	26.6	13.2	-28.6	-209.9	-35.4	-25.9		≥	-8.5
15.5	9.8	-8.6	-9.1	-9.3	-9.1	40.1	7.6	-16.5	-35.4	-214.6	-51.2		≥	-12.5

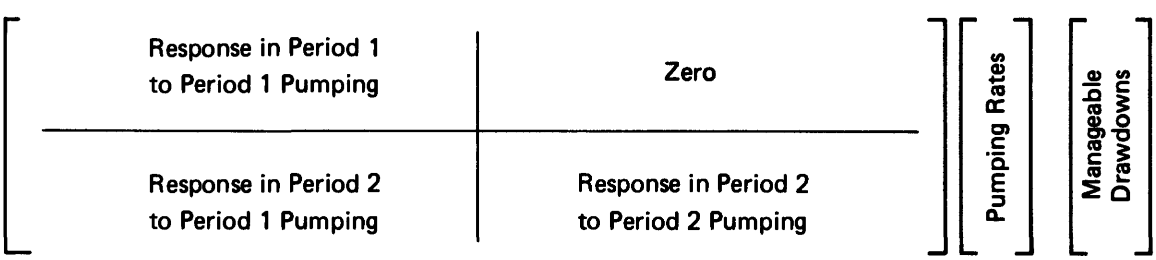

Figure 9. The sample problem, re-formulated with a response matrix.

On a PRIME 850 mini-computer, AQMAN used 76 CPU seconds for this problem. MINOS used 16 CPU seconds.

It is also instructive to see the effect of head definitions on the internally re-formulated problem. For example, the user wants the optimal head at locations 1 and 2 during both periods to appear directly in the solution, and KDEFHD is set equal to 1. The re-formulated problem is shown in Figure 10. Extra unity values have been added to the response matrix, inequalities are changed to equalities, and the value of the right-hand side is now unmanaged heads. The user will have to manually add constraints (32) and (33) to the MPS file.

Maximize:

$$Q_{1,1} + Q_{2,1} - Q_{3,1} - Q_{4,1} - Q_{5,1} - Q_{6,1} + Q_{1,2} + Q_{2,2} - Q_{3,2} - Q_{4,2} - Q_{5,2} - Q_{6,2}$$

Subject to:

8.2	19.4	-32.0	-17.4	-10.4	-7.8	0.0	0.0	0.0	0.0	0.0	0.0					≤	22.2
13.4	8.6	-17.2	-28.7	-19.3	-14.7	0.0	0.0	0.0	0.0	0.0	0.0					≤	18.2
17.5	4.5	-9.6	-18.7	-29.0	-24.1	0.0	0.0	0.0	0.0	0.0	0.0					≤	15.3
15.7	2.6	-5.8	-11.7	-20.4	-26.2	0.0	0.0	0.0	0.0	0.0	0.0					≤	11.5
179.9	6.7	-14.0	-26.3	-39.1	-45.3	0.0	0.0	0.0	0.0	0.0	0.0	1				=	236.5
159.6	6.0	-13.0	-25.1	-39.1	-47.6	0.0	0.0	0.0	0.0	0.0	0.0		1			=	236.5
10.0	62.0	-29.0	-15.8	-9.7	-7.3	0.0	0.0	0.0	0.0	0.0	0.0					≤	159.7
155.6	6.0	-12.9	-25.2	-39.9	-49.3	0.0	0.0	0.0	0.0	0.0	0.0					≤	155.8
431.9	6.7	-14.0	-26.6	-40.1	-47.1	0.0	0.0	0.0	0.0	0.0	0.0					≤	185.9
6.7	387.2	-27.4	-13.2	-7.6	-5.6	0.0	0.0	0.0	0.0	0.0	0.0					≥	-7.3
14.0	27.4	-210.7	-28.6	-16.5	-12.3	0.0	0.0	0.0	0.0	0.0	0.0					≥	-18.7
26.6	13.2	-28.6	-209.9	-35.4	-25.9	0.0	0.0	0.0	0.0	0.0	0.0					≥	-21.8
40.1	7.6	-16.5	-35.4	-214.6	-51.2	0.0	0.0	0.0	0.0	0.0	0.0					≥	-23.9
7.7	14.8	-9.2	-7.0	-5.7	-5.1	8.2	19.4	-32.0	-17.4	-10.4	-7.8					≤	33.8
8.1	8.6	-6.3	-5.8	-5.3	-5.0	13.4	8.6	-17.2	-28.7	-19.3	-14.7					≤	26.2
7.9	5.6	-4.7	-4.9	-4.9	-4.8	17.5	4.5	-9.6	-18.7	-29.0	-24.1					≤	21.5
6.3	3.7	-3.3	-3.6	-3.7	-3.7	15.7	2.6	-5.8	-11.7	-20.4	-26.2					≤	15.9
28.9	12.0	-12.2	-14.6	-15.9	-16.2	179.9	6.7	-14.0	-26.3	-39.1	-45.3			1		=	253.7
29.7	11.4	-12.0	-14.6	-16.1	-16.5	159.6	6.0	-13.0	-25.1	-39.1	-47.6				1	=	253.5
13.8	37.5	-20.7	-14.3	-10.9	-9.4	10.0	62.0	-29.0	-15.8	-9.7	-7.3					≤	182.7
28.9	11.2	-11.7	-14.2	-15.7	-16.1	155.6	6.0	-12.9	-25.2	-39.9	-49.3					≤	172.5
28.1	11.8	-12.0	-14.2	-15.5	-15.8	431.9	6.7	-14.0	-26.6	-40.1	-47.1					≤	202.6
11.8	39.6	-20.6	-13.5	-9.8	-8.3	6.7	387.2	-27.4	-13.2	-7.6	-5.6					≥	19.3
12.0	20.6	-13.1	-10.3	-8.6	-7.7	14.0	27.4	-210.7	-28.6	-16.5	-12.3					≥	-2.2
14.2	13.5	-10.3	-9.7	-9.1	-8.7	26.6	13.2	-28.6	-209.9	-35.4	-25.9					≥	-8.5
15.5	9.8	-8.6	-9.1	-9.3	-9.1	40.1	7.6	-16.5	-35.4	-214.6	-51.2					≥	-12.5

Variable vector: $[Q_{1,1}, Q_{2,1}, Q_{3,1}, Q_{4,1}, Q_{5,1}, Q_{6,1}, Q_{1,2}, Q_{2,2}, Q_{3,2}, Q_{4,2}, Q_{5,2}, Q_{6,2}, H_{1,1}, H_{2,1}, H_{1,2}, H_{2,2}]^T$

Figure 10. The re-formulated sample problem with four head definitions added.

REFERENCES

Aguado, E., and Remson, I., 1980, Ground-water management with fixed charges: Journal of Water Resources Planning and Management Division, American Society of Civil Engineering, v. 106, no. WR2, p. 375-382.

Anderson, T., Burt, O.R., and Fractor, D.T., 1983, Privatizing groundwater basins: a model and its application: in Water Rights, Scarce Resource Allocation, Bureaucracy, and the Environment, by T.L. Anderson (editor), Pacific Institute for Policy Research, San Francisco, 348 p.

Atwood, D.F., and Gorelick, S.M., 1985, Hydraulic gradient control for ground water contaminant removal: Journal of Hydrology, v. 76, p. 85-106.

Chaudry, M.T., Labadie, J.W., Hall, W.A., and Albertson, M.L., 1974, Optimal conjunctive use model for the Indus basin: Journal of Hydraulics Division, American Society of Civil Engineering, v. 100, no. HY5, p. 667- 687.

Cohon, J.L., and Marks, D.H., 1975, A review and evaluation of multi-objective programming techniques: Water Resources Research, v. 11, no. 2, p. 208- 220.

Colarullo, S.J., Heidari, M., and Maddock III, T., 1984, Identification of an optimal groundwater strategy in a contaminated aquifer: Water Resources Bulletin, v. 20, no. 5, p. 747-760.

Danskin, W.R., and Gorelick, S.M., 1985, A policy evaluation tool: management of a multiaquifer system using controlled stream recharge: Water Resources Research, v. 21, no. 11, p. 1731-1747.

Dantzig, G.B., 1963, Linear Programming and Extensions: Princeton University Press, Princeton, 627 p.

Freeze, R.A., and Cherry, J.A., 1979, Groundwater: Prentice-Hall, Englewood Cliffs, 604 p.

Gorelick, S.M., 1983, A review of distributed parameter groundwater management modeling methods: Water Resources Research, v. 19, no. 2, p. 305-319.

Gorelick, S.M., Voss, C.I., Gill, P.E., Murray, W., Saunders, M.A., and Wright, M.H., 1984, Aquifer reclamation design: the use of contaminant transport simulation combined with non-linear programming: Water Resources Research, v. 20, no. 4, p. 415-427.

Gorelick, S.M., and Wagner, B.J., 1986, Evaluating strategies for ground water contaminant plume stabilization and removal, Selected Papers in the Hydrologic Sciences: U.S. Geological Survey Water-Supply Paper 2290, pp. 81-89.

Heidari, M., 1982, Application of linear system's theory and linear programming to groundwater management in Kansas: Water Resources Bulletin, v. 18, no. 6, p. 1003-1012.

Hillier, F.S., and Lieberman, G.J., 1974, Introduction to Operations Research: Holden-Day, San Francisco, 829 p.

IBM, Mathematical programming system/360, Version 2, Linear and separable programming--User's Manual: Document No. H20-0476-2, IBM Corporation, New York.

Khepar, S.D., and Chaturvedi, M.C., 1982, Optimum cropping pattern and ground water management: Water Resources Bulletin, v. 18, no. 4, p. 655-660.

Larson, S.P., Maddock III, T., and Papadopulos, S., 1977, Optimization techniques applied to groundwater development: Memoirs of the International Association of Hydrogeology, v. 13, p. E57-E67.

Lee, A.S., and Aronofsky, J.S., 1958, A linear programming model for scheduling crude oil production: Journal of Petroleum Technology, v. 213, p. 51-54.

Lefkoff, L.J., and Gorelick, S.M., 1985, Rapid removal of a groundwater contaminant plume: Proceedings of the Symposium on Groundwater Contamination and Reclamation, American Water Resources Association, p. 125-131.

Lefkoff, L.J., and Gorelick, S.M., 1986, Design and cost analysis of rapid aquifer restoration systems using flow simulation and quadratic programming: Ground Water, v. 24, no. 6, p. 777-790.

Maddock III, T., 1972, Algebraic technological function from a simulation model: Water Resources Research, v. 8, no. 1, p. 129-134.

Murtagh, B.A., and Saunders, M.A., 1983, MINOS 5.0 User's Guide: Technical Report SOL 83-20, Department of Operations Research, Stanford University, Stanford, 118 p.

Rosenwald, G.W., and Green, D.W., 1974, A method for determining the optimal location of wells in a reservoir using mixed-integer programming: Society of Petroleum Engineering Journal, v. 14, p. 44-54.

Schwarz, J., 1976, Linear models for groundwater management: Journal of Hydrology, v. 28, p. 377-392.

Trescott, P.C., Pinder, G.F., and Larson, S.P., 1976, Finite-difference model for aquifer simulation in two dimensions with results of numerical experiments: U.S. Geological Survey Techniques of Water Resources Investigations, Book 7, Chap. C1, 116 p.

Wagner, 1975, Principles of Operations Research: Prentice-Hall, Englewood Cliffs, 1039 p.

Wattenbarger, R.A., 1970, Maximizing seasonal withdrawls from gas storage reservoirs: Journal of Petroleum Technology, p. 994-998.

Willis, R., 1983, A unified approach to regional groundwater management: in Groundwater Hydraulics, Water Resource Monograph Series, by J.S. Rosenshein and G.D. Bennett (edited), American Geophysical Union, Washington, 416 p.

APPENDIX I -- DEFINITION OF VARIABLES

ANNDIS = the annual discount rate, used only with quadratic objective function. Read by Subroutine QUAD from unit 13.

CASE = the type of objective function in a linear management problem. Must equal either "LINEAR" or "QUAD". Read by PRE from unit 14.

CDELT = multiplying factor for time steps. Read by subroutine PRE from unit 14. The Trescott variable CDLT is set equal to CDELT.

CONHD(JN) = user-defined limit on head at each control location JN. Read by Main from unit 14. JN goes from 1 to (NCNTR x NNPER). The CONHD vector is ordered by control location numbers within each sequential management period.

CONTYP(JN) = user-defined type of head constraint at control location JN. Read by Main from unit 14. Must equal either "L" for "less than or equal to", "G" for "greater than or equal to", or "E" for "equal to". Because constraints are re-formulated in terms of drawdowns, these inequalities are reversed in the MPS file. The vector CONTYP is ordered as is CONHD.

COSTC(IQ,N) = pumping cost per unit pumping rate per unit lift for well IQ during period N. Read as undiscounted unit cost from unit 13 by subroutine QUAD. Subroutine QUAD and MPSFMT apply a discount factor and a scaling factor to each unit pumping cost. IQ includes only quadratic (usually, pumping) managed wells.

COSTF = unit pumping cost like COSTC, but constant for all wells and all periods. If KEYCOS equals 0, pumping costs are constant and read by subroutine QUAD from unit 13 as COSTF. Subroutine QUAD then applies a discount factor and enters COSTF into the COSTC matrix.

DRDRES(JN) = drawdown response at each control location during each period. For each decision well, AQMAN's Main calls subroutine PRE to obtain heads HD, computes DRDRES(JN) = HDSS(JN) - HD(JN), and passes DRDRES to subroutine MPSFMT, which writes it into the column section of the MPS file.

FIXQ(I,N) = pre-determined pumping rate given for unmanaged well I during pumping period N. Read from unit 14 by subroutine PRE.

GCON(KN) = user-defined limit on either the gradient or the seepage velocity between the two locations of control pair KN during each period. Read from unit 13 by Subroutine GRADS. KN goes from 1 to (NGRAD x NPER). GCON is ordered by control pairs within each sequential pumping period. The convention adopted is GCON = Head(1) - Head(2).

GFACT(KN) = user-defined factor by which GCON is multiplied. Read from unit 13 by Subroutine GRADS. GFACT converts GCON to a difference in head. If simply the difference in head is managed, GFACT=1.0. If gradient is managed, GFACT = (the distance between the two locations). If velocity is managed, GFACT = (the distance between the two locations) x (effective aquifer porosity) / (hydraulic conductivity between the two locations). See equations

(11) and (12). KN goes from 1 to (NGRAD x NPER). GFACT is ordered as is GCON.

GRADE(KN) = manageable gradient or manageable velocity for control pair KN. Subroutine GRADS computes GRADE = HDUS(1) - HDUS(2) - (GCON x GFACT). GRADE(KN) is written in the RHS section of the MPS file for the row corresponding to constraint KN. The vector GRADE is ordered as is GCON.

GRATYP(KN) = user-defined type of gradient or velocity constraint for control pair KN. Must be either "L" for "less than or equal to", "G" for "greater than or equal to", or "E" for "equal to". Because constraints are re-formulated in terms of drawdowns, these inequalities are reversed in the MPS file. The GRATYP vector is ordered as is GCON.

HD(JN) = heads computed by subroutine TRES. JN goes from 1 to (NCNTR x NNPER). HD is ordered by control locations within pumping periods.

HDUS(JN) = unstressed, transient heads computed on first call of subroutine PRE. HDUS contains the heads that would occur over time if no managed wells were stressed.

ILOCC(J), JLOCC (J) = row (y-axis) and column (x-axis) location of control node J. Read by subroutine PRE from unit 14. J goes from 1 to NCNTR.

ILOCG1(K), JLOCG1(K) = row (y-axis) location and column (x-axis) location of the first node of control pair K. Read by subroutine GRADS from unit 13. Every ILOCG1 must correspond to some entry in the ILOCC vector, and every JLOCG1 must correspond to some entry in the JLOCC vector. K goes from 1 to NGRAD.

ILOCG2(K), JLOCG2(K) = row (y-axis) location and column (x-axis) location of the second node of control pair K. Read by subroutine GRADS from unit 13. Every ILOCG2 must correspond to some entry in the ILOCC vector, and every JLOCG2 must correspond to some entry in the JLOCC vector. K goes from 1 to NGRAD.

ILOCW(I), JLOCW (I) = row (y-axis) and column (x-axis) location of each managed or unmanaged well I. Read by subroutine PRE from unit 14. I goes from to NWLS.

KCALL, KCALLP = counter for number of times AQMAN's main has called subroutine PRE.

KDEFHD (J) = toggle (0 if no, 1 if yes) to indicate whether head definition is desired at control location J. Read by subroutine PRE from until 14. J goes from 1 to NCNTR. If KDEFHD (J) = 0, total drawdown from unmanaged conditions is constrained. If KDEFHD(J) = 1, a new decision variable is introduced and defined as unmanaged head minus total drawdown; the user may manually enter constraints on the new variable into the MPS file.

KDEFGR(K) = toggle (0 if no, 1 if yes) to indicate whether gradient or velocity definition is desired at control pair K. Read by subroutine PRE from unit 14. K goes from 1 to NGRAD. If KDEFGR(K) = 0, total difference-in-drawdown from unmanaged conditions is constrained. If KDEFGR(K) = 1, a new decision variable is introduced and defined as the unmanaged difference in head minus total difference-in-drawdown; the user may manually enter constraints on the new variable into the MPS file.

KEYCOS = toggle (0 if yes, 1 if no) to indicate whether undiscounted pumping cost COSTC is constant for all wells over all periods. Read from unit 13 by subroutine QUAD. Used only with a quadratic objective function. If KEYCOS=0, a single cost factor COSTF is read, discounted on a monthly basis, and entered into the COSTC matrix. If KEYCOS=1, variable COSTC is read and discounted.

KEYGRD(J) = toggle (0 if no, 1 if yes) to indicate whether primary control location J is only to be used for gradient or velocity control. Read from unit 14 by subroutine PRE. J goes from 1 to NCNTR. If KEYGRD(J) = 0, either (a) the head or (b) both the head and the gradient or velocity is controlled at location J, so that a drawdown response at J will be written in the MPS file by Subroutine MPSFMT. If KEYGRD(J) = 1, only the gradient or velocity is controlled at J, so that no drawdown response will be written in the MPS file; only a difference in drawdown response for the control pair to which control location J belongs will be written.

KEYQ(I) = toggle (0 if no, 1 if yes) to denote whether well I is a fixed, unmanaged well. Read by PRE from unit 14. I goes from 1 to NWLS. If there is a quadratic objective function, enter KEYQ=2 for non-quadratic (recharge) managed wells, and KEYQ=0 for quadratic (pumping) managed wells. When reading input data on the wells, all quadratic wells must be listed before any non-quadratic wells.

KEYWL(J) = toggle (0 if no, 1 if yes) to indicate whether control node J also contains a well, so that head at the well radius is needed. Read by Subroutine PRE from unit 14. J goes from 1 to NCNTR. The number of non-zero KEYWLs must equal the value of NW read by subroutine TRES from unit 15. For each non-zero value of KEYWL, XRAD at that node must be positive. For each zero value of KEYWL, XRAD must be zero.

KPER = pumping period counter. KPER is increased by 1 in subroutine TRES at line S2-730, after the new pumping period begins but before the SOLVE subroutines are called. KPER counts from 1 to NNPER.

MNGDRD(JN) = manageable drawdown at each control location during each period. Computed as MNGDRD(JN) = HDUS(JN) - CONHD(JN). MNGDRD is written by subroutine MPSFMT into the RHS section of the MPS file.

NNAME = the alphanumeric name given to the MPS data set. Read by subroutine PRE from unit 14. May contain up to 64 characters.

NCNTR = number of control locations. Read by Subroutine PRE from unit 14. Current maximum value (for storage of ILOCC, JLOCC, KEYWL) is 500.

NGRAD = number of gradient control pairs. Read by Subroutine PRE from unit 14. Each gradient pair consists of two control locations that have each been previously read as one of the NCNTR primary control nodes.

NKEYQ = the number of fixed (unmanaged) wells. Read by Subroutine PRE from unit 14.

NNPER = number of pumping periods. Read by Subroutine PRE from unit 14. The Trescott variable NPER is set equal to NNPER. The current maximum value of NNPER (for storage of TIMINC) is 50.

NWLS = total number of wells (managed plus unmanaged). Read by subroutine PRE from unit 14. NWLS is read instead of the Trescott variable NWEL. Current maximum value (for storage of XRAD, ILOCW, JLOCW) is 200.

QWELL(IN) = pumpage and recharge rates. IN goes from 1 to (NWEL x NNPER). QWELL is used instead of the Trescott variable WELL. The current maximum value for (NWEL x NNPER) is 500. QWELL is ordered by well numbers within pumping periods.

RDIF(KN) = the difference in drawdown response between the two locations of a control pair. KN goes from 1 to (NGRAD x NNPER). RDIF(KN) is written in the column section of the MPS file for the constraint corresponding to control pair KN.

SURF(IQ) = the elevation of the land surface at each quadratic (pumping) managed well. Read from unit 13 by subroutine QUAD. IQ goes from 1 to (number of quadratic decision wells). SURF is used to compute unmanaged lift required at these decision wells.

TIMINC(N) = number of hours in initial time step for each pumping period. Read by subroutine PRE from unit 14. N goes from 1 to NNPER. TIMINC is read instead of the Trescott variable DELT.

TIMPER = number of days in each pumping period. Read by subroutine PRE from unit 14. TIMPER is read instead of the Trescott variable TMAX.

UNITQ(I) = unit pumpage or recharge stress. Read by subroutine PRE from unit 14. UNITQ is negative for pumping, positive for recharge. If a problem requires scaling, the UNITQ vector can vary accordingly (eg., -1.0, 0.1, -0.001). All cost coefficients in the objective function will be automatically scaled (multiplied) by UNITQ at the appropriate well. The user must be careful to re-scale the final optimization results.

XHEAD(JN) = computed head at control locations. JN goes from 1 to (NCNTR x NNPER). XHEAD is ordered by locations within pumping periods.

XRAD(I) = well radius. Read by subroutine PRE from unit 14. I goes from 1 to NWLS. XRAD is read instead of the Trescott variable RADIUS. Non-zero values must be entered for any well that is also a control location, and zero values entered for any well that is not a control location.

APPENDIX II--DATA FILE INSTRUCTIONS

Two input files are always required for AQMAN. A third is needed if the problem involves gradients or velocities or if a quadratic objective function is desired. English or metric units can be used, as long as all input data are consistent.

1. **Logical unit 15** contains the usual Trescott data set. "Group IV" data may be included, but will be ignored by AQMAN. "Group IV" contains the pumping period parameters and the well locations, pumpages, and radii (Trescott and others, 1976, pp. 49-55).

2. **Logical unit 14** contains input read by subroutine PRE and Main:

Data Set	Number of Lines	Columns	Format	Variable	Definition
1	1	1-4	A4	CASE	The type of objective function in a linear problem. LINE for linear function, QUAD for quadratic.
		7-70	16A4	NNAME	Any title the user wishes to print on the first line of the MPS file.
2	1	1-10	I10	NWLS	Total number of wells (managed plus unmanaged) in the system. NWLS must be constant for all pumping periods.
		11-20	I10	NCNTR	Number of control locations, constant for all pumping periods. HD and XHEAD will contain the computed head at these nodal locations.
		21-30	I10	NNPER	Number of pumping periods to be simulated.
		31-40	G10.0	CDELT	Multiplying factor for DELT, constant for all pumping periods. Used instead of the Trescott variable CDLT.
		41-50	I10	NGRAD	Number of gradient controls. Each control consists of two locations, specified by ILOCG1, JLOCG1, ILOCG2, and JLOCG2. If there are no gradients, velocities, or head-differences to control, enter 0 for NGRAD.
3	1	1-80	8G10.0	TIMPER	Number of days in each time period.

Data Set	Number of Lines	Columns	Format	Variable	Definition
4	1+	1-80	8G10.0	TIMINC(N)	Number of hours in initial time step for each period. TIMINC is a vector containing NPER elements; if NPER exceeds 8, more than one line will be needed. See discussion of time parameters in text.
5	1	1-10	I10	NKEYQ	Number of fixed (unmanaged) wells. Cannot be greater than NWLS.
6	NWLS	1-10	I10	ILOCW(I)	Row (y-axis) location of well I. ILOCW is a vector containing NWLS elements.
		11-20	I10	JLOCW(I)	Column (x-ais) location of well I. JLOCW is a vector containing NWLS elements.
		21-40	G10.0	XRAD(I)	Radius [L] of well I. XRAD is a vector containing NWLS elements.
		31-40	I10	KEYQ(I)	Switch (0=no, 1=yes) to indicate whether I is an unmanaged well. For all cases, if KEYQ(I)=1 pumpage at well I is not a decision variable, and no drawdown response is computed for it. For a quadratic objective, KEYQ(I)=2 if I is a nonquadratic, managed well, and KEYQ(I)=0 if I is a quadratic, managed well.
		41-50	F10.0	UNITQ(I)	The unit pumping [L^3/T] rate used at well I obtain drawdown responses. A negative value indicates a pumping well; a positive value indicates a recharge well. **Use care if UNITQ is not constant for all decision wells. See discussion on scaling in text.

Note--If a KEYQ=1 is read for well I, the reading of the above five variables is interrupted, and FIXQ is read for each period at well I. <u>NNPER lines are required at each interruption</u>.

Data Set	Number of Lines	Columns	Format	Variable	Definition
6	NNPER	1-10	G10.0	FIXQ(I,N)	Pumpage or recharge [L^3/T] at the unmanaged well I during period N. There must be NNPER values for FIXQ following a KEYQ=1.
7	NCNTR	1-10	I10	ILOCC(J)	Row (y-axis) location of control node J. ILOCC is a vector containing NCNTR elements.
		11-20	I10	JLOCC(J)	Column (x-axis) location of control node J. JLOCC is a vector containing NCNTR elements.
		21-30	I10	KEYWL(J)	Switch to indicate whether control location J also contains a well. If KEYWL(J) does not equal zero, a well exists at J, and XRAD is used to compute head at the well, rather than a cell-averaged head.

Note--The number of non-zero KEYWLs must equal the value of NW read by the TRESCOTT code from unit 15. For each non-zero value of KEYWL, XRAD at that node must be positive.

Data Set	Number of Lines	Columns	Format	Variable	Definition
		31-40	I10	KEYGRD(J)	Switch to indicate whether control location J is to be used only for gradient or velocity control. If KEYGRD(J)=0, J will be used for head control and may be used for gradient or velocity control. If KEYGRD(J)=1, J will not be used for head control: no constraint row and no response coefficient for J will be written in the MPS file.

Data Set	Number of Lines	Columns	Format	Variable	Definition
		41-50	I10	KDEFHD(J)	Switch to indicate whether head definition is desired at location J If KDEFHD(J)=0, total drawdown is constrained. If KDEFHD(J)=1, head is defined and any constraint must be manually added to the MPS file.
8	NNPER x NCNTR	1-10	G10.0	CONHD(JN)	Desired upper or lower limit [L] on average cell head or head at a well radius at each control location during each pumping period. The first NCNTR lines are for period 1, the second NCNTR lines are for period 2, etc. Read dummy values for locations where head is not constrained.
		15	A1	CONTYP(JN)	Type of constraint on head at each control location. Must be either "L" for ≤, "G" for ≥, or, "E" for =

3. Logical unit 13 is used only if NGRAD is positive or CASE equals "QUAD". It contains input read by subroutines QUAD and GRADS. If the problem is not quadratic, omit the lines containing ANNDIS, COSTF, KEYCOS, COSTC, and SURF.

Data Set	Number of Lines	Columns	Format	Variable	Definition
1	1	1-10	G10.3	ANNDIS	Annual discount rate [L°], applied to pumping costs in each period on a monthly basis.
		11-20	G10.0	COSTF	Undiscounted pumping cost [$], constant for all wells in all periods. Used only if KEYCOS =0
		21-30	I10	KEYCOS	Switch to indicate whether pumping costs are constant (0) or variable (1). If KEYCOS=0, COSTC=COSTF. If KEYCOS=1, variable COSTC is read.
2	NNPER+	1-80	8G10.0	COSTC(IQ,N)	Undiscounted pumping cost [$] at quadratic managed well IQ during pumping period N. If IQ exceeds 8, more than one line will be needed for each period.

Note--Omit if KEYCOS=0

Data Set	Number of Lines	Columns	Format	Variable	Definition
3	1+	1-80	8G10.0	SURF(IQ,N)	Land surface elevation [L] at quadratic managed well. IQ goes from 1 to the number of managed wells included in the quadratic objective (usually, pumping wells). A new line is needed for every 8 entries.
4	NGRAD	1-10	I10	ILOCG1(K)	Row (y-axis) location of the first node of control pair K. Must correspond to some ILOCC, i.e., must already be specified as a head control location.
		11-20	I10	JLOCG1(K)	Column (x-axis) location of the first node of control pair K. Must correspond to some JLOCC, i.e., must already be specified as a head control location.
		21-30	I10	ILOCG2(K)	Row (y-axis) location of the second node of control pair K. Must correspond to some ILOCC, i.e., must already be specified as a head control location.
		31-40	I10	JLOCG2(K)	Column (x-axis) location of the second node of control pair K. Must correspond to some JLOCC, i.e., must already be specified as head control location.
		41-50	I10	KDEFGR(K)	Switch to indicate whether head definition is desired at control pair K. If KDEFGR(K) = 0 total difference-in-drawdown is constrained. If KDEFGR(K) = 1, difference in head is defined and any constraint must be manually added to the MPS file.
5	NNPER x NGRAD	1-15	G15.6	GFACT(KN)	Factor by which GCON is multiplied. Used to control velocities or gradients rather than head differences.

Data Set	Number of Lines	Columns	Format	Variable	Definition
		16-30	G15.6	GCON(KN)	Constraint on gradient [L/L] or or velocity [L/T] between the two locations of a control pair. Direction defined as (value at 1) (value at 2).
		35	A1	GRATYP(KN)	The type of gradient or velocity constraint at a control pair. Must be either "L" for ≤, "G" for ≥, or "E" for =.

Three output files must always be open for program execution. Unit 16 will contain error and warning messages. Unit 17 will contain the user-defined limits on head and the unmanaged heads for all control locations, and pumping or recharge rates at all wells. Unit 18 is the MPS file, to be used as input for a mathematical programming package.

A fourth output file is needed for quadratic problems. AQMAN writes to unit 19 the drawdown responses at all pumping (quadratic) managed wells. (These are recognized as control locations at which KEYWL=1 and KEYGRD=1). Unit 19 will be used by the optimization model to compute the value and gradient of the quadratic objective function for a given set of decisions. See Appendix III.

APPENDIX III -- QUADRATIC OBJECTIVE: SUBROUTINE FUNOBJ

In order to execute an optimization code that solves a quadratic objective, the user must supply to the code a subroutine that evaluates the value and the gradient of the quadratic objective function for a given set of decisions. The quadratic portion of the objective is given by:

$$F = \sum_{j=1}^{J} \sum_{n=1}^{N} \sum_{i=1}^{I} \sum_{k=1}^{n} C_{j,n}\, Q_{j,n}\, Q_{i,k}\, \beta_{i,j,(n-k)} \qquad \text{(A1)}$$

Where J = number of pumping managed wells,

I = number of managed (pumping + recharge) wells,

N = number of management periods,

C = unit cost of pumping per unit pumping rate per unit lift [$\$L^{-4}$ T],

Q = pumping or recharge rate at managed wells [L^3/T],

β = response coefficient between the head at well j and pumping or recharge at well i separated by (n-k) management periods [L^0]

The gradient is the partial derivative with respect to each of the decision variables Q:

$$\frac{\partial F}{\partial Q_{w,p}} = \sum_{j=1}^{J} \sum_{n=k}^{N} C_{j,n}\, Q_{j,n}\, \beta_{i,j,(n-k)} \begin{bmatrix} \text{for} \\ i=w \\ k=p \end{bmatrix}$$

$$+ \sum_{i=1}^{I} \sum_{k=1}^{n} C_{j,n}\, Q_{i,k}\, \beta_{i,j,(n-k)} \begin{bmatrix} \text{for} \\ j=w \\ n=p \end{bmatrix} \qquad \begin{matrix} w = 1,\ldots I \\ p = 1,\ldots N \end{matrix} \qquad \text{(A2)}$$

This appendix contains two versions of a subroutine that calculates (A1) and (A2). The subroutine, FUNOBJ, was written for use by the mathematical programming package MINOS (Murtagh and Saunders, 1983), but should be easily adaptable to other optimization codes. The two versions differ only in the amount of computer CPU time and storage required. The first version operates faster, and is recommended for most problems. This version reads all input data once, and needs CPU core storage space at least as large as the input files. The second version of FUNOBJ should only be used for very large problems. It reads input data repeatedly, requiring less storage but much more input/output time. The definition of a "large" problem depends on the computer facilities available; if the CPU can easily store all input files, the first version of FUNOBJ should be used. For example, if the input file that requires the most storage needs 0.5 megabyte of core and the problem is to be run on an 8.0 megabyte machine, the problem is not large and the fast FUNOBJ should be used.

In addition to the MPS file, two other input files are needed to execute a quadratic programming code. One of these is the quadratic response file, which is written by AQMAN as unit 19. It contains the drawdown response at "response wells" to a unit stress applied at all managed wells. "Response wells" consist of all pumping managed wells plus all recharge managed wells where head is controlled. Usually, head is controlled at all recharge wells to remain below some level, such as the land surface. In this case, all managed wells are response wells. There may be some exceptional problems in which head is not controlled at all recharge decision wells. If this is specified in the input files for AQMAN (units 14 and 13), the number of response wells written to unit 19 will be less than the number of decision wells.

During execution of the optimization code combined with Subroutine FUNOBJ, the quadratic response file will require more storage than any other input file. (The MPS file will be larger, but most optimization codes store it in a compact form.) The size of the quadratic response file determines whether the fast version of FUNOBJ can be used. The number of lines in the file equals the number of decision wells times the number of response wells times the number of management periods. The variable RBETA in FUNOBJ must be dimensioned large enough to store these numbers. For instance, say that a problem contains 50 managed wells and 25 periods, and that all of the managed wells are response wells. The file will contain 62,500 lines, and RBETA should be dimensioned 50 by 50 by 25. Since RBETA is double precision, this would reserve 0.5 megabytes of core. If the computer has a core memory of 8.0 megabytes, the fast FUNOBJ can be used.

In order for the data in the MPS file and the quadratic response file to be properly organized for the optimization code and for FUNOBJ, two precautions must be taken before execution of AQMAN. (1) Pumping wells, which all appear in the quadratic terms of the pumping cost objective, must be included as control locations (part of NCNTR) in order to obtain the well heads needed to compute induced lifts and cost coefficients. At each of these wells, if head-control is not desired, enter KEYGRD=1. This will prevent the writing of a constraint row for these wells to the MPS file. Remember that for each managed well at control locations, XRAD must be positive and KEYWL set equal to 1. (2) When entering well information into unit 14, all quadratic managed wells (usually, pumping wells) must be listed before non-quadratic managed wells (usually, recharge wells). This is needed for proper ordering of the responses written to unit 19. For non-quadratic managed wells, enter KEYQ=2. For quadratic managed wells, enter KEYQ=0. As always, enter KEYQ=1 for unmanaged wells.

The other input file needed by FUNOBJ is created by the user. It is unit number 12. Definitions and format instructions are given below. A listing of the two versions of Subroutine FUNOBJ follows.

Data Set	Number of Lines	Columns	Format	Variable	Definition
1	1	1-10	I10	NDWLS	The number of decision wells.
		11-20	I10	NRES	The number of response wells. This is all pumping managed wells (quadratics) plus all recharge managed wells where head is controlled. Usually, NRES equals NDWLS.
		21-30	I10	NPWLS	The number of pumping managed wells, where quadratic pumping costs are being minimized.
		***Note--NPWLS cannot be greater than NRES, and NRES cannot be greater than NDWLS.			
		31-40	I10	NPER	The number of management (pumping) periods.
2	1	1-80	G10.2	TIMPER	Number of days in each management period. TIMPER is used here for discounting future costs.
3	1	1-10	G10.3	ANNDIS	Annual discount rate [L^0], applied to pumping costs in each period.
		11-20	G10.3	COSTF	Undiscounted pumping cost [$], constant for all wells in all periods. Used only if KEYCOS = 0.
		21-30	I10	KEYCOS	Switch to indicate whether pumping costs are constant (0) or spatially variable (1). If KEYCOS=0, COSTC=COSTF. If KEYCOS=1, variable COSTC is read.
4	NNPER+	1-80	8G10.2	COSTC(IQ,N)	Undiscounted pumping cost [$] at quadratic managed well IQ during pumping period N. If I>8, more than one line needed for each period.
		** Note--Omit if KEYCOS=0			
5	NPWLS	1-10	G10.2	UNITQ(IP)	The unit pumping rate [$-L^3/T$] applied by AQMAN at each well IP. Used to scale cost coefficients in the quadratic objective.

```
C---Subroutine FUNOBJ for MINOS with quadratic objective function.
C---Calculates value and gradient of the objective function.
C---This version requires more storage, but runs must faster.
      SUBROUTINE FUNOBJ(MODE,NN,X,F,G,NSTATE,NPROB,Z,NWCORE)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(NN),G(NN),Z(NWCORE)
C*** Check dimensions!! Be sure that RBETA(I,K,N), COSTC(I,N), and
C*** Q(I,N) are large enough for the problem:
C    I=number of decision (managed) wells, NDWLS
C    K=number of response wells, NRES
C    N=number of management periods, NPER
      COMMON /QUADAQ/ RBETA(30,30,4),COSTC(30,4),Q(30,4),UNITQ(30)
C---Read data only if first call by MINOS.
      IF(NSTATE.NE.1) GO TO 170
C---Read number of decision (managed) wells, response (observation)
C----wells, pumping (quadratic) wells, and pumping periods.
C**** NOTE *** In ordering decision vector, all response wells must
C         be first decision wells, and all pumping wells must be
C         first response wells.
      READ(12,10)NDWLS,NRES,NPWLS,NPER
   10 FORMAT(4I10)
C---Read length (in days) of pumping periods
      READ(12,20)TIMPER
   20 FORMAT(G10.2)
C--Read annual discount rate, constant pumping cost, and key to
C----indicate whether costs are constant (0) or variable (1):
      READ(12,25)ANNDIS,COSTF,KEYCOS
   25 FORMAT(2G10.3,I10)
      DISMON=ANNDIS/12.0
      NDAYS=0
      DO 140 N=1,NPER
       NDAYS=NDAYS+TIMPER
       NMONTH=NDAYS/30
       DISFAC=1./((1.+DISMON)**NMONTH)
       IF(KEYCOS.EQ.0) GO TO 120
C--Read variable pumping costs into cost vector and discount according
C----to time period lengths by months:
       READ(12,30)(COSTC(I,N),I=1,NPWLS)
   30  FORMAT(8G10.2)
       DO 110 I=1,NPWLS
  110  COSTC(I,N)=DISFAC*COSTC(I,N)
       GO TO 140
C--Put constant pumping costs into costs vector and discount by time
C---period:
  120  DO 130 I=1,NPWLS
  130  COSTC(I,N)=COSTF*DISFAC
  140 CONTINUE
C--Read unit pumping rates used to obtain responses.  Be sure that each
C---unit rate corresponds to correct pumping well!  Remember: NEGATIVE
C---for pumping.
      DO 145 I=1,NPWLS
  145 READ(12,20)UNITQ(I)
C--Scale cost coefficients according to unit pumping rates.
```

```
      DO 150 N=1,NPER
      DO 150 I=1,NPWLS
  150 COSTC(I,N)=COSTC(I,N)*(-UNITQ(I))
C--Read matrix of responses between well I and well L for separated by
C----(NPER-1) pumping periods.
      DO 160 I=1,NDWLS
      DO 160 N=1,NPER
      DO 160 L=1,NRES
       READ(19,50)RBETA(I,L,N)
   50  FORMAT(20X,F15.7)
  160 CONTINUE
C--Convert pumpage vector into two dimensional matrix:
  170 DO 180 I=1,NDWLS
      DO 180 N=1,NPER
       KPUMP=N+(I-1)*NPER
       Q(I,N)=X(KPUMP)
  180 CONTINUE
C--Compute value of objective function.
      F=0.0
      DO 200 L=1,NPWLS
      DO 200 N=1,NPER
       FTEMP=0.0
       DO 190 I=1,NDWLS
       DO 190 K=1,N
        KT=N-K+1
        FTEMP=FTEMP+(Q(I,K)*RBETA(I,L,KT))
  190  CONTINUE
       F=F+(FTEMP*COSTC(L,N)*Q(L,N))
  200 CONTINUE
C-Compute value of gradient at each decision (managed) well during each
C---management period.
      DO 230 IWL=1,NDWLS
      DO 230 IPER=1,NPER
       KG=IPER+(IWL-1)*NPER
       G(KG)=0.0
       DO 225 I=1,NDWLS
       DO 225 K=1,NPER
        IF(I.NE.IWL .OR. K.NE.IPER)GO TO 225
        DO 220 L=1,NPWLS
        DO 220 N=K,NPER
         KT=N-K+1
         G(KG)=G(KG)+(COSTC(L,N)*Q(L,N)*RBETA(IWL,L,KT))
  220   CONTINUE
        GO TO 230
  225  CONTINUE
  230 CONTINUE
      DO 300 LWL=1,NPWLS
      DO 300 IPER=1,NPER
       G2=0.0
       DO 280 L=1,NPWLS
        DO 270 N=1,NPER
         IF(L.NE.LWL .OR. N.NE.IPER)GO TO 270
         DO 260 I=1,NDWLS
         DO 260 K=1,N
```

```
        KT=N-K+1                                                        FUNa1090
        G2=G2+(COSTC(L,N)*Q(I,K)*RBETA(I,LWL,KT))                       FUNa1100
260    CONTINUE                                                         FUNa1110
       KG=IPER+(LWL-1)*NPER                                             FUNa1120
       G(KG)=G(KG)+G2                                                   FUNa1130
       GO TO 300                                                        FUNa1140
270   CONTINUE                                                          FUNa1150
280  CONTINUE                                                           FUNa1160
300 CONTINUE                                                            FUNa1170
    RETURN                                                              FUNa1180
    END                                                                 FUNa1190
```

```
220    CONTINUE                                                          FUNb1080
       GO TO 230                                                         FUNb1090
225   CONTINUE                                                           FUNb1100
230  CONTINUE                                                            FUNb1110
     DO 300 LWL=1,NPWLS                                                  FUNb1120
      REWIND 19                                                          FUNb1130
      IF(LWL.EQ.1)GO TO 240                                              FUNb1140
      LSKIP=LWL-1                                                        FUNb1150
      DO 235 ISKIP=1,LSKIP                                               FUNb1160
235   READ(19,50)DUMMY                                                   FUNb1170
240   DO 250 I=1,NDWLS                                                   FUNb1180
      DO 250 N=1,NPER                                                    FUNb1190
       IF(I.EQ.1 .AND. N.EQ.1)GO TO 250                                  FUNb1200
       NSKIP=NRES-1                                                      FUNb1210
       DO 245 ISKIP=1,NSKIP                                              FUNb1220
245    READ(19,50)DUMMY                                                  FUNb1230
250   READ(19,50)RBETA(I,N)                                              FUNb1240
     DO 300 IPER=1,NPER                                                  FUNb1250
      G2=0.0                                                             FUNb1260
      DO 280 L=1,NPWLS                                                   FUNb1270
       DO 270 N=1,NPER                                                   FUNb1280
        IF(L.NE.LWL .OR. N.NE.IPER)GO TO 270                             FUNb1290
        DO 260 I=1,NDWLS                                                 FUNb1300
        DO 260 K=1,N                                                     FUNb1310
         KT=N-K+1                                                        FUNb1320
         G2=G2+(COSTC(L,N)*Q(I,K)*RBETA(I,KT))                           FUNb1330
260     CONTINUE                                                         FUNb1340
        KG=IPER+(LWL-1)*NPER                                             FUNb1350
        G(KG)=G(KG)+G2                                                   FUNb1360
        GO TO 300                                                        FUNb1370
270    CONTINUE                                                          FUNb1380
280   CONTINUE                                                           FUNb1390
300  CONTINUE                                                            FUNb1400
     RETURN                                                              FUNb1410
     END                                                                 FUNb1420
```

APPENDIX IV -- DATA FILES FOR SAMPLE PROBLEM

At the start of each input file, an extra line is inserted to indicate column locations. This line appears only here, and is not part of the actual input files used for the sample problem. Blank lines are also indicated.

UNIT 15, input file

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
--------- SAMPLE PROBLEM, based on Figure 33 in Trescott Documentation---------
------------
                    RECH SIP  CHEC                NUME HEAD
        10        14         6       800
(blank line)
         2         1      .003       .01                                      10
         1         1         1
(blank line)
(blank line)
(blank line)
       200
         1         1
(blank line)
                .001.001.001.001.001.001.001.001
            .001.001.001.001.001.001.001.001.001.001
            .001.001.001.001.001.001.001.001.001.001
            .001.001.001.001.001.001.001.001.001.001
            .001.001.001.001.001.001.001.001.001.001
            .001.001.001.001.001.001.001.001.001
    .001.001.001.001.001.001.001.001.001.001.001  -1
                              -1  -1  -1  -1  -1  -1
(blank line)
      .002         1
(blank line)
                   1   1   1       2   2   2   2
               1   1   1   1   2   2   2   2   2   2
               1   1   1   2   2   2   2   2   2   2
               2   2   2   2   2   2   2   2   2   2
               2   2   2   3   3   3   3   3   3   3
               3   3   3   3   4   4   4   4   4
       4   4   4   4   4   4   4   3   3   3   3   3
                               3   3   3   3   3   3
(blank line)
    .2E-07
        50         1
        20        14         9         9        14        21        31        41
        37        25        17        11         9        13
        50         1
        10         5         7        10        14        18        27        30
        31        12
         1         0         6         1         1       1.0        24
         4         4       .05
         5         4       .05
         6         4       .05
         7         4       .05
```

```
C---Subroutine FUNOBJ for MINOS with quadratic objective function.
C---Calculates value and gradient of the objective function.
C---This version requires less storage, but runs much slower.
      SUBROUTINE FUNOBJ(MODE,NN,X,F,G,NSTATE,NPROB,Z,NWCORE)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(NN),G(NN),Z(NWCORE)
C*** Check dimensions!! Be sure that RBETA(I,N), COSTC(I,N), and
C*** Q(I,N) are large enough for the problem:
C    I=number of decision (managed) wells, NDWLS
C    N=number of management periods, NPER
      COMMON /QUADAQ/ RBETA(200,50),COSTC(200,50),Q(200,50),UNITQ(200)
C---Read data only if first call by MINOS.
      IF(NSTATE.NE.1) GO TO 155
C---Read number of decision (managed) wells, response (observation)
C----wells, pumping (quadratic) wells, and pumping periods.
C**** NOTE *** In ordering decision vector, all response wells must
C         be first decision wells, and all pumping wells must be
C         first response wells.
      READ(12,10)NDWLS,NRES,NPWLS,NPER
   10 FORMAT(4I10)
C---Read length (in days) of pumping periods
      READ(12,20)TIMPER
   20 FORMAT(8G10.2)
C--Read annual discount rate, constant pumping cost, and key to
C----indicate whether costs are constant (0) or variable (1):
      READ(12,25)ANNDIS,COSTF,KEYCOS
   25 FORMAT(2G10.3,I10)
      DISMON=ANNDIS/12.0
      NDAYS=0
      DO 130 N=1,NPER
       NDAYS=NDAYS+TIMPER
       NMONTH=NDAYS/30
       DISFAC=1./((1.+DISMON)**NMONTH)
       IF(KEYCOS.EQ.0) GO TO 110
C--Read variable pumping costs into cost vector and discount according
C----to time period lengths by months:
       READ(12,30)(COSTC(I,N),I=1,NPWLS)
   30  FORMAT(8G10.2)
       DO 100 I=1,NPWLS
  100  COSTC(I,N)=DISFAC*COSTC(I,N)
       GO TO 130
C--Put constant pumping costs into costs vector and discount by time
C---period:
  110  DO 120 I=1,NPWLS
  120  COSTC(I,N)=COSTF*DISFAC
  130 CONTINUE
C--Read unit pumping rates used to obtain responses.  Be sure that each
C---unit rate corresponds to correct pumping well!  Remember: NEGATIVE
C---for pumping.
      DO 140 I=1,NPWLS
  140 READ(12,20)UNITQ(I)
C--Scale cost coefficients according to unit pumping rates.
```

```
      DO 150 N=1,NPER
      DO 150 I=1,NPWLS
  150 COSTC(I,N)=COSTC(I,N)*(-UNITQ(I))
C--Convert pumpage vector into two dimensional matrix:
  155 DO 160 I=1,NDWLS
      DO 160 N=1,NPER
       KPUMP=N+(I-1)*NPER
       Q(I,N)=X(KPUMP)
  160 CONTINUE
C--Compute value of objective function.
      F=0.0
      DO 200 L=1,NPWLS
       REWIND 19
       IF(L.EQ.1)GO TO 163
       LSKIP=L-1
       DO 162 ISKIP=1,LSKIP
  162  READ(19,50)DUMMY
  163  DO 170 I=1,NDWLS
       DO 170 N=1,NPER
        IF(N.EQ.1 .AND. I.EQ.1)GO TO 170
        NSKIP=NRES-1
        DO 165 ISKIP=1,NSKIP
  165   READ(19,50)DUMMY
  170  READ(19,50)RBETA(I,N)
   50  FORMAT(20X,F15.7)
       FTEMP2=0.0
       DO 190 N=1,NPER
        FTEMP1=0.0
        DO 180 I=1,NDWLS
        DO 180 K=1,N
         KRES=N-K+1
         FTEMP1=FTEMP1+Q(I,K)*RBETA(I,KRES)
  180   CONTINUE
       FTEMP2=FTEMP2+(COSTC(L,N)*Q(L,N)*FTEMP1)
  190  CONTINUE
       F=F+FTEMP2
  200 CONTINUE
C-Compute value of gradient at each decision (managed) well during each
C---decision period.
      REWIND 19
      DO 230 IWL=1,NDWLS
       DO 210 N=1,NPER
       DO 210 L=1,NRES
  210  READ(19,50)RBETA(L,N)
      DO 230 IPER=1,NPER
       KG=IPER+(IWL-1)*NPER
       G(KG)=0.0
       DO 225 I=1,NDWLS
       DO 225 K=1,NPER
        IF(I.NE.IWL .OR. K.NE.IPER)GO TO 225
        DO 220 L=1,NPWLS
        DO 220 N=K,NPER
          KT=N-K+1
          G(KG)=G(KG)+(COSTC(L,N)*Q(L,N)*RBETA(L,KT))
```

```
         4         11        -10
         6          6        -10
```

UNIT 14, input file

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
LINE  SAMPLE PROBLEM
         7         17          2       1.5         4
      30.0
      24.0       24.0
         1
         3          7       0.50         1
    -0.005
    -0.010
         4         11       0.50                -1.0
         6          6       0.50                -1.0
         7          8       0.25                 1.0
         7          9       0.25                 1.0
         7         10       0.25                 1.0
         7         11       0.25                 1.0
         3         11
         3         12
         4         11          1
         4         12
         6          6          1
         7          8          1
         7          9          1
         7         10          1
         7         11          1
         8          8                    1
         8          9                    1
         8         10                    1
         8         11                    1
         9          8                    1
         9          9                    1
         9         10                    1
         9         11                    1
      80.0    G
      80.0    G
      80.0    G
      80.0    G
      50.0    G
     250.0    L
     250.0    L
     250.0    L
     250.0    L
         0
         0
         0
         0
         0
         0
         0
         0
      80.0    G
```

```
    80.0   G
    80.0   G
    80.0   G
    50.0   G
   250.0   L
   250.0   L
   250.0   L
   250.0   L
       0
       0
       0
       0
       0
       0
       0
       0
```

UNIT 13, input file

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
         8         8         9         8
         8         9         9         9
         8        10         9        10
         8        11         9        11
       28.5           0.01    G
       28.5           0.01    G
       28.5           0.01    G
       28.5           0.01    G
       28.5           0.01    G
       28.5           0.01    G
       28.5           0.01    G
       28.5           0.01    G
```

UNIT 16, ouput

```
  *****WARNING*** THE UNITQ VECTOR AS READ FROM UNIT 14 IS NOT UNIFORM,
  SO DIFFERENT UNIT STRESSES WILL BE APPLIED  AT SOME WELLS.
 THE OBJECTIVE FUNCTION COEFFICIENTS IN THE MPS FILE WILL BE SCALED ACCORDINGLY.
```

UNIT 17, output

CONTROL LOCATIONS AND UNSTRESSED HEADS

	Period	Location #	I-Loc.	J-Loc.	Unmanaged Head	KEYWL
1	1	1	3	11	236.546565	0
2	1	2	3	12	236.446269	0
3	1	3	4	11	239.695702	1
4	1	4	4	12	235.842201	0
5	1	5	6	6	235.880777	1
6	1	6	7	8	242.667764	1
7	1	7	7	9	231.336352	1
8	1	8	7	10	228.216894	1
9	1	9	7	11	226.083924	1
10	1	10	8	8	222.510705	0
11	1	11	8	9	218.438309	0
12	1	12	8	10	215.595586	0

13	1	13	8	11	211.735532	0
14	1	14	9	8	200.000000	0
15	1	15	9	9	200.000000	0
16	1	16	9	10	200.000000	0
17	1	17	9	11	200.000000	0
18	2	1	3	11	253.715131	0
19	2	2	3	12	253.448525	0
20	2	3	4	11	262.752494	1
21	2	4	4	12	252.467227	0
22	2	5	6	6	252.636217	1
23	2	6	7	8	269.255351	1
24	2	7	7	9	247.825653	1
25	2	8	7	10	241.450164	1
26	2	9	7	11	237.517420	1
27	2	10	8	8	234.059561	0
28	2	11	8	9	226.531944	0
29	2	12	8	10	221.799685	0
30	2	13	8	11	216.178031	0
31	2	14	9	8	200.000000	0
32	2	15	9	9	200.000000	0
33	2	16	9	10	200.000000	0
34	2	17	9	11	200.000000	0

USER IMPOSED LIMITS ON HEAD AT CONTROL LOCATIONS

Period	Loc. #	I-Location	J-Location	Limit	Type	KEYGRD	KDEFHD
1	1	3	11	80.0	G	0	0
1	2	3	12	80.0	G	0	0
1	3	4	11	80.0	G	0	0
1	4	4	12	80.0	G	0	0
1	5	6	6	50.0	G	0	0
1	6	7	8	250.	L	0	0
1	7	7	9	250.	L	0	0
1	8	7	10	250.	L	0	0
1	9	7	11	250.	L	0	0
1	10	8	8	0.000		1	0
1	11	8	9	0.000		1	0
1	12	8	10	0.000		1	0
1	13	8	11	0.000		1	0
1	14	9	8	0.000		1	0
1	15	9	9	0.000		1	0
1	16	9	10	0.000		1	0
1	17	9	11	0.000		1	0
2	1	3	11	80.0	G	0	0
2	2	3	12	80.0	G	0	0
2	3	4	11	80.0	G	0	0
2	4	4	12	80.0	G	0	0
2	5	6	6	50.0	G	0	0
2	6	7	8	250.	L	0	0
2	7	7	9	250.	L	0	0
2	8	7	10	250.	L	0	0
2	9	7	11	250.	L	0	0
2	10	8	8	0.000		1	0
2	11	8	9	0.000		1	0

2	12	8	10	0.000	1	0
2	13	8	11	0.000	1	0
2	14	9	8	0.000	1	0
2	15	9	9	0.000	1	0
2	16	9	10	0.000	1	0
2	17	9	11	0.000	1	0

WELL LOCATIONS AND TYPE

Period	Loc. #	I-Loc.	J-Loc.	KEYQ	Fixed or Unit Rate
1	1	3	7	1	-.5000E-02
1	2	4	11	0	-1.000
1	3	6	6	0	-1.000
1	4	7	8	0	1.000
1	5	7	9	0	1.000
1	6	7	10	0	1.000
1	7	7	11	0	1.000
2	1	3	7	1	-.1000E-01
2	2	4	11	0	-1.000
2	3	6	6	0	-1.000
2	4	7	8	0	1.000
2	5	7	9	0	1.000
2	6	7	10	0	1.000
2	7	7	11	0	1.000

CONTROL PAIR LOCATIONS AND DEFINITIONS

Pair #	1st I-Loc.	1st J-Loc.	2nd I-Loc.	2nd J-Loc.	KDEFGR
1	8	8	9	8	0
2	8	9	9	9	0
3	8	10	9	10	0
4	8	11	9	11	0

USER IMPOSED LIMITS ON HEAD DIFFRENCE AT CONTROL PAIRS

Period	Pair #	Conversion Factor	Difference Limit	Type
1	1	28.5000	0.100000E-01	G
1	2	28.5000	0.100000E-01	G
1	3	28.5000	0.100000E-01	G
1	4	28.5000	0.100000E-01	G
2	1	28.5000	0.100000E-01	G
2	2	28.5000	0.100000E-01	G
2	3	28.5000	0.100000E-01	G
2	4	28.5000	0.100000E-01	G

UNIT 18, output -- The MPS file

```
NAME          SAMPLE PROBLEM
ROWS
 L  DIF01001
 L  DIF01002
 L  DIF01003
 L  DIF01004
 L  DIF02001
 L  DIF02002
 L  DIF02003
 L  DIF02004
 L  DR010001
 L  DR010002
 L  DR010003
 L  DR010004
 L  DR010005
 G  DR010006
 G  DR010007
 G  DR010008
 G  DR010009
 L  DR020001
 L  DR020002
 L  DR020003
 L  DR020004
 L  DR020005
 G  DR020006
 G  DR020007
 G  DR020008
 G  DR020009
 N  OBJ
COLUMNS
    Q01001    OBJ        1.0000
    Q01001    DIF01001   0.81642E+01   DIF01002   0.13411E+02
    Q01001    DIF01003   0.17481E+02   DIF01004   0.15720E+02
    Q01001    DIF02001   0.77105E+01   DIF02002   0.81007E+01
    Q01001    DIF02003   0.79189E+01   DIF02004   0.62541E+01
    Q01001    DR010001   0.17992E+03   DR010002   0.15956E+03
    Q01001    DR010003   0.99542E+01   DR010004   0.15561E+03
    Q01001    DR010005   0.43191E+03   DR010006   0.66832E+01
    Q01001    DR010007   0.14001E+02   DR010008   0.26553E+02
    Q01001    DR010009   0.40146E+02
    Q01001    DR020001   0.28949E+02   DR020002   0.29659E+02
    Q01001    DR020003   0.13823E+02   DR020004   0.28879E+02
    Q01001    DR020005   0.28078E+02   DR020006   0.11768E+02
    Q01001    DR020007   0.11951E+02   DR020008   0.14197E+02
    Q01001    DR020009   0.15493E+02
    Q02001    OBJ        1.0000
    Q02001    DIF02001   0.81642E+01   DIF02002   0.13411E+02
    Q02001    DIF02003   0.17481E+02   DIF02004   0.15720E+02
    Q02001    DR020001   0.17992E+03   DR020002   0.15956E+03
    Q02001    DR020003   0.99542E+01   DR020004   0.15561E+03
    Q02001    DR020005   0.43191E+03   DR020006   0.66832E+01
```

Q02001	DR020007	0.14001E+02	DR020008	0.26553E+02
Q02001	DR020009	0.40146E+02		
Q01002	OBJ	1.0000		
Q01002	DIF01001	0.19372E+02	DIF01002	0.86257E+01
Q01002	DIF01003	0.45041E+01	DIF01004	0.26437E+01
Q01002	DIF02001	0.14814E+02	DIF02002	0.85608E+01
Q01002	DIF02003	0.55619E+01	DIF02004	0.36689E+01
Q01002	DR010001	0.67215E+01	DR010002	0.60359E+01
Q01002	DR010003	0.62028E+02	DR010004	0.59800E+01
Q01002	DR010005	0.66834E+01	DR010006	0.38717E+03
Q01002	DR010007	0.27418E+02	DR010008	0.13228E+02
Q01002	DR010009	0.76126E+01		
Q01002	DR020001	0.11981E+02	DR020002	0.11428E+02
Q01002	DR020003	0.37489E+02	DR020004	0.11225E+02
Q01002	DR020005	0.11769E+02	DR020006	0.39594E+02
Q01002	DR020007	0.20561E+02	DR020008	0.13456E+02
Q01002	DR020009	0.98090E+01		
Q02002	OBJ	1.0000		
Q02002	DIF02001	0.19372E+02	DIF02002	0.86257E+01
Q02002	DIF02003	0.45041E+01	DIF02004	0.26437E+01
Q02002	DR020001	0.67215E+01	DR020002	0.60359E+01
Q02002	DR020003	0.62028E+02	DR020004	0.59800E+01
Q02002	DR020005	0.66834E+01	DR020006	0.38717E+03
Q02002	DR020007	0.27418E+02	DR020008	0.13228E+02
Q02002	DR020009	0.76126E+01		
Q01003	OBJ	-1.0000		
Q01003	DIF01001	-0.31994E+02	DIF01002	-0.17166E+02
Q01003	DIF01003	-0.95931E+01	DIF01004	-0.57806E+01
Q01003	DIF02001	-0.91502E+01	DIF02002	-0.63330E+01
Q01003	DIF02003	-0.47126E+01	DIF02004	-0.33140E+01
Q01003	DR010001	-0.14011E+02	DR010002	-0.12973E+02
Q01003	DR010003	-0.28951E+02	DR010004	-0.12909E+02
Q01003	DR010005	-0.14001E+02	DR010006	-0.27417E+02
Q01003	DR010007	-0.21074E+03	DR010008	-0.28579E+02
Q01003	DR010009	-0.16479E+02		
Q01003	DR020001	-0.12223E+02	DR020002	-0.11970E+02
Q01003	DR020003	-0.20667E+02	DR020004	-0.11720E+02
Q01003	DR020005	-0.11950E+02	DR020006	-0.20561E+02
Q01003	DR020007	-0.13104E+02	DR020008	-0.10295E+02
Q01003	DR020009	-0.85876E+01		
Q02003	OBJ	-1.0000		
Q02003	DIF02001	-0.31994E+02	DIF02002	-0.17166E+02
Q02003	DIF02003	-0.95931E+01	DIF02004	-0.57806E+01
Q02003	DR020001	-0.14011E+02	DR020002	-0.12973E+02
Q02003	DR020003	-0.28951E+02	DR020004	-0.12909E+02
Q02003	DR020005	-0.14001E+02	DR020006	-0.27417E+02
Q02003	DR020007	-0.21074E+03	DR020008	-0.28579E+02
Q02003	DR020009	-0.16479E+02		
Q01004	OBJ	-1.0000		
Q01004	DIF01001	-0.17407E+02	DIF01002	-0.28650E+02
Q01004	DIF01003	-0.18668E+02	DIF01004	-0.11718E+02
Q01004	DIF02001	-0.69626E+01	DIF02002	-0.57777E+01
Q01004	DIF02003	-0.48619E+01	DIF02004	-0.36042E+01
Q01004	DR010001	-0.26331E+02	DR010002	-0.25115E+02

Q01004	DR010003	-0.15836E+02	DR010004	-0.25183E+02
Q01004	DR010005	-0.26553E+02	DR010006	-0.13228E+02
Q01004	DR010007	-0.28579E+02	DR010008	-0.20992E+03
Q01004	DR010009	-0.35396E+02		
Q01004	DR020001	-0.14571E+02	DR020002	-0.14561E+02
Q01004	DR020003	-0.14342E+02	DR020004	-0.14227E+02
Q01004	DR020005	-0.14201E+02	DR020006	-0.13455E+02
Q01004	DR020007	-0.10295E+02	DR020008	-0.96834E+01
Q01004	DR020009	-0.91080E+01		
Q02004	OBJ	-1.0000		
Q02004	DIF02001	-0.17407E+02	DIF02002	-0.28650E+02
Q02004	DIF02003	-0.18668E+02	DIF02004	-0.11718E+02
Q02004	DR020001	-0.26331E+02	DR020002	-0.25115E+02
Q02004	DR020003	-0.15836E+02	DR020004	-0.25183E+02
Q02004	DR020005	-0.26553E+02	DR020006	-0.13228E+02
Q02004	DR020007	-0.28579E+02	DR020008	-0.20992E+03
Q02004	DR020009	-0.35396E+02		
Q01005	OBJ	-1.0000		
Q01005	DIF01001	-0.10410E+02	DIF01002	-0.19316E+02
Q01005	DIF01003	-0.28980E+02	DIF01004	-0.20445E+02
Q01005	DIF02001	-0.56896E+01	DIF02002	-0.53342E+01
Q01005	DIF02003	-0.48657E+01	DIF02004	-0.37299E+01
Q01005	DR010001	-0.39102E+02	DR010002	-0.39148E+02
Q01005	DR010003	-0.96869E+01	DR010004	-0.39893E+02
Q01005	DR010005	-0.40147E+02	DR010006	-0.76122E+01
Q01005	DR010007	-0.16480E+02	DR010008	-0.35396E+02
Q01005	DR010009	-0.21464E+03		
Q01005	DR020001	-0.15928E+02	DR020002	-0.16135E+02
Q01005	DR020003	-0.10893E+02	DR020004	-0.15749E+02
Q01005	DR020005	-0.15498E+02	DR020006	-0.98091E+01
Q01005	DR020007	-0.85883E+01	DR020008	-0.91081E+01
Q01005	DR020009	-0.92744E+01		
Q02005	OBJ	-1.0000		
Q02005	DIF02001	-0.10410E+02	DIF02002	-0.19316E+02
Q02005	DIF02003	-0.28980E+02	DIF02004	-0.20445E+02
Q02005	DR020001	-0.39102E+02	DR020002	-0.39148E+02
Q02005	DR020003	-0.96869E+01	DR020004	-0.39893E+02
Q02005	DR020005	-0.40147E+02	DR020006	-0.76122E+01
Q02005	DR020007	-0.16480E+02	DR020008	-0.35396E+02
Q02005	DR020009	-0.21464E+03		
Q01006	OBJ	-1.0000		
Q01006	DIF01001	-0.77557E+01	DIF01002	-0.14700E+02
Q01006	DIF01003	-0.24083E+02	DIF01004	-0.26166E+02
Q01006	DIF02001	-0.50783E+01	DIF02002	-0.50368E+01
Q01006	DIF02003	-0.47606E+01	DIF02004	-0.37032E+01
Q01006	DR010001	-0.45348E+02	DR010002	-0.47645E+02
Q01006	DR010003	-0.73425E+01	DR010004	-0.49326E+02
Q01006	DR010005	-0.47098E+02	DR010006	-0.55942E+01
Q01006	DR010007	-0.12294E+02	DR010008	-0.25909E+02
Q01006	DR010009	-0.51208E+02		
Q01006	DR020001	-0.16210E+02	DR020002	-0.16521E+02
Q01006	DR020003	-0.93659E+01	DR020004	-0.16119E+02
Q01006	DR020005	-0.15761E+02	DR020006	-0.82578E+01
Q01006	DR020007	-0.77416E+01	DR020008	-0.86745E+01

```
    Q01006    DR020009   -0.91486E+01
    Q02006    OBJ         -1.0000
    Q02006    DIF02001   -0.77557E+01   DIF02002   -0.14700E+02
    Q02006    DIF02003   -0.24083E+02   DIF02004   -0.26166E+02
    Q02006    DR020001   -0.45348E+02   DR020002   -0.47645E+02
    Q02006    DR020003   -0.73425E+01   DR020004   -0.49326E+02
    Q02006    DR020005   -0.47098E+02   DR020006   -0.55942E+01
    Q02006    DR020007   -0.12294E+02   DR020008   -0.25909E+02
    Q02006    DR020009   -0.51208E+02
RHS
    RHS       DIF01001    0.22226E+02   DIF01002    0.18153E+02
    RHS       DIF01003    0.15311E+02   DIF01004    0.11451E+02
    RHS       DIF02001    0.33775E+02   DIF02002    0.26247E+02
    RHS       DIF02003    0.21515E+02   DIF02004    0.15893E+02
    RHS       DR010001    0.15655E+03   DR010002    0.15645E+03
    RHS       DR010003    0.15970E+03   DR010004    0.15584E+03
    RHS       DR010005    0.18588E+03   DR010006   -0.73322E+01
    RHS       DR010007   -0.18664E+02   DR010008   -0.21783E+02
    RHS       DR010009   -0.23916E+02
    RHS       DR020001    0.17372E+03   DR020002    0.17345E+03
    RHS       DR020003    0.18275E+03   DR020004    0.17247E+03
    RHS       DR020005    0.20264E+03   DR020006    0.19255E+02
    RHS       DR020007   -0.21743E+01   DR020008   -0.85498E+01
    RHS       DR020009   -0.12483E+02
ENDATA
```

The Modified MPS file

The MPS file above printed by AQMAN is modified for the sample problem to include a constraint that balances total pumping and injection (equation 45). Changes were performed by manually editing the file, and are indicated below in **bold type**.

```
NAME          SAMPLE PROBLEM
ROWS
 L  DIF01001
 L  DIF01002
 L  DIF01003
 L  DIF01004
 L  DIF02001
 L  DIF02002
 L  DIF02003
 L  DIF02004
 L  DR010001
 L  DR010002
 L  DR010003
 L  DR010004
 L  DR010005
 G  DR010006
 G  DR010007
 G  DR010008
 G  DR010009
 L  DR020001
 L  DR020002
 L  DR020003
 L  DR020004
 L  DR020005
 G  DR020006
 G  DR020007
 G  DR020008
 G  DR020009
 G  BAL1
 G  BAL2
 N  OBJ
COLUMNS
    Q01001    OBJ         1.0000        BAL1        1.000
    Q01001    DIF01001    0.81642E+01   DIF01002    0.13411E+02
    Q01001    DIF01003    0.17481E+02   DIF01004    0.15720E+02
    Q01001    DIF02001    0.77105E+01   DIF02002    0.81007E+01
    Q01001    DIF02003    0.79189E+01   DIF02004    0.62541E+01
    Q01001    DR010001    0.17992E+03   DR010002    0.15956E+03
    Q01001    DR010003    0.99542E+01   DR010004    0.15561E+03
    Q01001    DR010005    0.43191E+03   DR010006    0.66832E+01
    Q01001    DR010007    0.14001E+02   DR010008    0.26553E+02
    Q01001    DR010009    0.40146E+02
    Q01001    DR020001    0.28949E+02   DR020002    0.29659E+02
    Q01001    DR020003    0.13823E+02   DR020004    0.28879E+02
    Q01001    DR020005    0.28078E+02   DR020006    0.11768E+02
    Q01001    DR020007    0.11951E+02   DR020008    0.14197E+02
```

Q01001	DR020009	0.15493E+02		
Q02001	OBJ	1.0000	**BAL2**	**1.000**
Q02001	DIF02001	0.81642E+01	DIF02002	0.13411E+02
Q02001	DIF02003	0.17481E+02	DIF02004	0.15720E+02
Q02001	DR020001	0.17992E+03	DR020002	0.15956E+03
Q02001	DR020003	0.99542E+01	DR020004	0.15561E+03
Q02001	DR020005	0.43191E+03	DR020006	0.66832E+01
Q02001	DR020007	0.14001E+02	DR020008	0.26553E+02
Q02001	DR020009	0.40146E+02		
Q01002	OBJ	1.0000	**BAL1**	**1.000**
Q01002	DIF01001	0.19372E+02	DIF01002	0.86257E+01
Q01002	DIF01003	0.45041E+01	DIF01004	0.26437E+01
Q01002	DIF02001	0.14814E+02	DIF02002	0.85608E+01
Q01002	DIF02003	0.55619E+01	DIF02004	0.36689E+01
Q01002	DR010001	0.67215E+01	DR010002	0.60359E+01
Q01002	DR010003	0.62028E+02	DR010004	0.59800E+01
Q01002	DR010005	0.66834E+01	DR010006	0.38717E+03
Q01002	DR010007	0.27418E+02	DR010008	0.13228E+02
Q01002	DR010009	0.76126E+01		
Q01002	DR020001	0.11981E+02	DR020002	0.11428E+02
Q01002	DR020003	0.37489E+02	DR020004	0.11225E+02
Q01002	DR020005	0.11769E+02	DR020006	0.39594E+02
Q01002	DR020007	0.20561E+02	DR020008	0.13456E+02
Q01002	DR020009	0.98090E+01		
Q02002	OBJ	1.0000	**BAL2**	**1.000**
Q02002	DIF02001	0.19372E+02	DIF02002	0.86257E+01
Q02002	DIF02003	0.45041E+01	DIF02004	0.26437E+01
Q02002	DR020001	0.67215E+01	DR020002	0.60359E+01
Q02002	DR020003	0.62028E+02	DR020004	0.59800E+01
Q02002	DR020005	0.66834E+01	DR020006	0.38717E+03
Q02002	DR020007	0.27418E+02	DR020008	0.13228E+02
Q02002	DR020009	0.76126E+01		
Q01003	OBJ	-1.0000	**BAL1**	**-1.000**
Q01003	DIF01001	-0.31994E+02	DIF01002	-0.17166E+02
Q01003	DIF01003	-0.95931E+01	DIF01004	-0.57806E+01
Q01003	DIF02001	-0.91502E+01	DIF02002	-0.63330E+01
Q01003	DIF02003	-0.47126E+01	DIF02004	-0.33140E+01
Q01003	DR010001	-0.14011E+02	DR010002	-0.12973E+02
Q01003	DR010003	-0.28951E+02	DR010004	-0.12909E+02
Q01003	DR010005	-0.14001E+02	DR010006	-0.27417E+02
Q01003	DR010007	-0.21074E+03	DR010008	-0.28579E+02
Q01003	DR010009	-0.16479E+02		
Q01003	DR020001	-0.12223E+02	DR020002	-0.11970E+02
Q01003	DR020003	-0.20667E+02	DR020004	-0.11720E+02
Q01003	DR020005	-0.11950E+02	DR020006	-0.20561E+02
Q01003	DR020007	-0.13104E+02	DR020008	-0.10295E+02
Q01003	DR020009	-0.85876E+01		
Q02003	OBJ	-1.0000	**BAL2**	**-1.000**
Q02003	DIF02001	-0.31994E+02	DIF02002	-0.17166E+02
Q02003	DIF02003	-0.95931E+01	DIF02004	-0.57806E+01
Q02003	DR020001	-0.14011E+02	DR020002	-0.12973E+02
Q02003	DR020003	-0.28951E+02	DR020004	-0.12909E+02
Q02003	DR020005	-0.14001E+02	DR020006	-0.27417E+02
Q02003	DR020007	-0.21074E+03	DR020008	-0.28579E+02

Q02003	DR020009	-0.16479E+02		
Q01004	OBJ	-1.0000	**BAL1**	**-1.000**
Q01004	DIF01001	-0.17407E+02	DIF01002	-0.28650E+02
Q01004	DIF01003	-0.18668E+02	DIF01004	-0.11718E+02
Q01004	DIF02001	-0.69626E+01	DIF02002	-0.57777E+01
Q01004	DIF02003	-0.48619E+01	DIF02004	-0.36042E+01
Q01004	DR010001	-0.26331E+02	DR010002	-0.25115E+02
Q01004	DR010003	-0.15836E+02	DR010004	-0.25183E+02
Q01004	DR010005	-0.26553E+02	DR010006	-0.13228E+02
Q01004	DR010007	-0.28579E+02	DR010008	-0.20992E+03
Q01004	DR010009	-0.35396E+02		
Q01004	DR020001	-0.14571E+02	DR020002	-0.14561E+02
Q01004	DR020003	-0.14342E+02	DR020004	-0.14227E+02
Q01004	DR020005	-0.14201E+02	DR020006	-0.13455E+02
Q01004	DR020007	-0.10295E+02	DR020008	-0.96834E+01
Q01004	DR020009	-0.91080E+01		
Q02004	OBJ	-1.0000	**BAL2**	**-1.000**
Q02004	DIF02001	-0.17407E+02	DIF02002	-0.28650E+02
Q02004	DIF02003	-0.18668E+02	DIF02004	-0.11718E+02
Q02004	DR020001	-0.26331E+02	DR020002	-0.25115E+02
Q02004	DR020003	-0.15836E+02	DR020004	-0.25183E+02
Q02004	DR020005	-0.26553E+02	DR020006	-0.13228E+02
Q02004	DR020007	-0.28579E+02	DR020008	-0.20992E+03
Q02004	DR020009	-0.35396E+02		
Q01005	OBJ	-1.0000	**BAL1**	**-1.000**
Q01005	DIF01001	-0.10410E+02	DIF01002	-0.19316E+02
Q01005	DIF01003	-0.28980E+02	DIF01004	-0.20445E+02
Q01005	DIF02001	-0.56896E+01	DIF02002	-0.53342E+01
Q01005	DIF02003	-0.48657E+01	DIF02004	-0.37299E+01
Q01005	DR010001	-0.39102E+02	DR010002	-0.39148E+02
Q01005	DR010003	-0.96869E+01	DR010004	-0.39893E+02
Q01005	DR010005	-0.40147E+02	DR010006	-0.76122E+01
Q01005	DR010007	-0.16480E+02	DR010008	-0.35396E+02
Q01005	DR010009	-0.21464E+03		
Q01005	DR020001	-0.15928E+02	DR020002	-0.16135E+02
Q01005	DR020003	-0.10893E+02	DR020004	-0.15749E+02
Q01005	DR020005	-0.15498E+02	DR020006	-0.98091E+01
Q01005	DR020007	-0.85883E+01	DR020008	-0.91081E+01
Q01005	DR020009	-0.92744E+01		
Q02005	OBJ	-1.0000	**BAL2**	**-1.000**
Q02005	DIF02001	-0.10410E+02	DIF02002	-0.19316E+02
Q02005	DIF02003	-0.28980E+02	DIF02004	-0.20445E+02
Q02005	DR020001	-0.39102E+02	DR020002	-0.39148E+02
Q02005	DR020003	-0.96869E+01	DR020004	-0.39893E+02
Q02005	DR020005	-0.40147E+02	DR020006	-0.76122E+01
Q02005	DR020007	-0.16480E+02	DR020008	-0.35396E+02
Q02005	DR020009	-0.21464E+03		
Q01006	OBJ	-1.0000	**BAL1**	**-1.000**
Q01006	DIF01001	-0.77557E+01	DIF01002	-0.14700E+02
Q01006	DIF01003	-0.24083E+02	DIF01004	-0.26166E+02
Q01006	DIF02001	-0.50783E+01	DIF02002	-0.50368E+01
Q01006	DIF02003	-0.47606E+01	DIF02004	-0.37032E+01
Q01006	DR010001	-0.45348E+02	DR010002	-0.47645E+02
Q01006	DR010003	-0.73425E+01	DR010004	-0.49326E+02

```
    Q01006    DR010005  -0.47098E+02   DR010006  -0.55942E+01
    Q01006    DR010007  -0.12294E+02   DR010008  -0.25909E+02
    Q01006    DR010009  -0.51208E+02
    Q01006    DR020001  -0.16210E+02   DR020002  -0.16521E+02
    Q01006    DR020003  -0.93659E+01   DR020004  -0.16119E+02
    Q01006    DR020005  -0.15761E+02   DR020006  -0.82578E+01
    Q01006    DR020007  -0.77416E+01   DR020008  -0.86745E+01
    Q01006    DR020009  -0.91486E+01
    Q02006    OBJ        -1.0000       BAL2        -1.000
    Q02006    DIF02001  -0.77557E+01   DIF02002  -0.14700E+02
    Q02006    DIF02003  -0.24083E+02   DIF02004  -0.26166E+02
    Q02006    DR020001  -0.45348E+02   DR020002  -0.47645E+02
    Q02006    DR020003  -0.73425E+01   DR020004  -0.49326E+02
    Q02006    DR020005  -0.47098E+02   DR020006  -0.55942E+01
    Q02006    DR020007  -0.12294E+02   DR020008  -0.25909E+02
    Q02006    DR020009  -0.51208E+02
RHS
    RHS       DIF01001   0.22226E+02   DIF01002   0.18153E+02
    RHS       DIF01003   0.15311E+02   DIF01004   0.11451E+02
    RHS       DIF02001   0.33775E+02   DIF02002   0.26247E+02
    RHS       DIF02003   0.21515E+02   DIF02004   0.15893E+02
    RHS       DR010001   0.15655E+03   DR010002   0.15645E+03
    RHS       DR010003   0.15970E+03   DR010004   0.15584E+03
    RHS       DR010005   0.18588E+03   DR010006  -0.73322E+01
    RHS       DR010007  -0.18664E+02   DR010008  -0.21783E+02
    RHS       DR010009  -0.23916E+02
    RHS       DR020001   0.17372E+03   DR020002   0.17345E+03
    RHS       DR020003   0.18275E+03   DR020004   0.17247E+03
    RHS       DR020005   0.20264E+03   DR020006   0.19255E+02
    RHS       DR020007  -0.21743E+01   DR020008  -0.85498E+01
    RHS       DR020009  -0.12483E+02
    RHS       BAL1          0.0000     BAL2          0.0000
ENDATA
```

Optimal Solution: output from MINOS, using modified MPS file as input

```
           M I N O S    ---    VERSION  4.9    MAR  1983
           = = = = =


 SPECS FILE
 ----------
          BEGIN SPECS FOR AWRA PROBLEM
             MAXIMIZE
             OBJECTIVE        OBJ
             RHS              RHS
             NONLINEAR  VARIABLES        0
             SUPERBASICS LIMIT           1
             DERIVATIVE LEVEL            3
             VERIFY LEVEL                1
             MULTIPLE PRICE              1
             ROWS                       50
             COLUMNS                   100
             ELEMENTS                 5000
             SCALE PRINT               YES
             FUNCTION PRECISION    1.0E-12
             MPS FILE                    9
             OLD BASIS FILE              0
             INSERT FILE                 0
             NEW BASIS FILE             55
             PUNCH FILE                 56
             BACKUP FILE                57
             SAVE FREQUENCY             10
             SOLUTION                  YES
             MAJOR ITERATIONS           10
             ITERATIONS                101
          END
1
 PARAMETERS
 ----------
 MPS INPUT DATA.
 ROW LIMIT..............        50      LIST LIMIT.............          0      LOWER BOUND DEFAULT....  0.00E-01
 COLUMN LIMIT...........       100      ERROR MESSAGE LIMIT....         10      UPPER BOUND DEFAULT....  1.00E+20
 ELEMENTS LIMIT (COEFFS)      5000      PHANTOM ELEMENTS.......          0      AIJ TOLERANCE..........  1.00E-10

 FILES.
 MPS FILE (INPUT FILE)..         9      OLD BASIS FILE (MAP)...          0      (CARD READER)..........         5
 SOLUTION FILE..........         0      NEW BASIS FILE (MAP)...         55      (PRINTER)..............         6
 INSERT FILE............         0      BACKUP BASIS FILE......         57      (SCRATCH FILE).........         8
 PUNCH FILE.............        56      LOAD FILE..............          0      DUMP FILE..............         0

 FREQUENCIES.
 LOG ITERATIONS.........         1      CHECK ROW ERROR........         30      CYCLE LIMIT............         1
 SAVE NEW BASIS MAP.....        10      FACTORIZE (INVERT).....         50      CYCLE TOLERANCE........  0.00E-01

 LP PARAMETERS.
```

```
ITERATIONS LIMIT.......        101     FEASIBILITY TOLERANCE..  1.00E-06     PARTIAL PRICE FACTOR...            1
CRASH OPTION...........          1     OPTIMALITY TOLERANCE...  1.00E-06     MULTIPLE PRICE.........            1
WEIGHT ON OBJECTIVE....   0.00E-01     PIVOT TOLERANCE........  3.67E-11     SCALE TOLERANCE........         0.90

NONLINEAR PROBLEMS.
NONLINEAR CONSTRAINTS..          0     HESSIAN DIMENSION......         0     FUNCTION PRECISION.....     1.00E-12
NONLINEAR JACOBIAN VARS          0     SUPERBASICS LIMIT......         1     DIFFERENCE INTERVAL....     1.00E-06
NONLINEAR OBJECTIV VARS          0     TRUNCATED CG METHOD....         1     CENTRAL DIFFCE INTERVAL     1.00E-04
PROBLEM NUMBER.........          0     LINESEARCH TOLERANCE...   0.10000     DERIVATIVE LEVEL.......            3
UNBOUNDED OBJECTV VALUE   1.00E+20     SUBSPACE TOLERANCE.....   0.20000     VERIFY LEVEL...........            1
UNBOUNDED STEP SIZE....   1.00E+10                                           EMERGENCY VERIFY LEVEL.            1

AUGMENTED LAGRANGIAN.
JACOBIAN...............     SPARSE     MAJOR ITERATIONS LIMIT.        10     RADIUS OF CONVERGENCE..     1.00E-02
LAGRANGIAN.............        YES     MINOR ITERATIONS LIMIT.        40     ROW TOLERANCE..........     1.00E-06
PENALTY PARAMETER......   0.00E-01     COMPLETION.............      FULL     PRINT LEVEL..(JFLXI)...            1
DAMPING PARAMETER......   2.00E+00

MISCELLANEOUS.
LU FACTOR TOLERANCE....   0.100000     WORKSPACE (USER).......         0     DEBUG LEVEL............            0
LU UPDATE TOLERANCE....   0.100000     WORKSPACE (TOTAL)......    200000     LINESEARCH DEBUG AFTER.       999999

REASONABLE WORKSPACE LIMITS ARE        0 ...     7917 ...   200000  WORDS
ACTUAL     WORKSPACE LIMITS ARE        0 ...   200000 ...   200000  WORDS
1
MPS FILE
--------
     1    NAME          SAMPLE P
     2    ROWS
    32    COLUMNS
   171    RHS
   187    ENDATA



NAMES SELECTED
--------------
OBJECTIVE     OBJ      (MAX)        1
RHS           RHS                  26
RANGES                              0
BOUNDS                              0



MATRIX STATISTICS
-----------------
              TOTAL      NORMAL        FREE       FIXED     BOUNDED
ROWS             29           2           1           0          26
COLUMNS          12          12           0           0           0

NO. OF MATRIX ELEMENTS       258     DENSITY       74.138
NO. OF REJECTED COEFFS         0      AIJTOL 1.00000E-10
BIGGEST AND SMALLEST COEFFS     4.31910E+02     1.00000E+00   (EXCLUDING OBJ AND RHS)
```

```
 LENGTH OF ROW-NAME HASH TABLE           101
 COLLISIONS DURING TABLE LOOKUP          125

 NONZEROS ALLOWED FOR IN LU FACTORS   132700
1
 ITERATIONS
 ----------
1

 SCALING
 -------
           MIN ELEM    MAX ELEM       MAX COL RATIO

 AFTER  0    1.00E+00    4.32E+02            431.91
 AFTER  1    1.22E-01    8.18E+00             66.91
 AFTER  2    1.28E-01    7.82E+00             61.22



 ROW SCALES  R(I)         (SCALED AIJ)  =  (ORIGINAL AIJ)  *  R(I)  /  C(J)
 ----------------

     1        0.06204     2         0.06237     3         0.07594     4         0.09511     5         0.07806
     6        0.08538     7         0.07594     8         0.09511     9         0.03008    10         0.03371
    11        0.04207    12         0.03429    13         0.01947    14         0.02055    15         0.01846
    16        0.01861    17         0.02409    18         0.03008    19         0.03371    20         0.04570
    21        0.03429    22         0.01947    23         0.02055    24         0.02477    25         0.02414
    26        0.02409    27         0.81297    28         0.85375    29         1.00000



 COLUMN SCALES  C(J)
 -------------------

     1        1.07463     2         1.07463     3         1.01703     4         1.01703     5         0.89464
     6        1.19301     7         0.92330     8         1.17375     9         0.89943    10         0.89943
    11        0.53492    12         0.53492


 CRASH OPTION  1
 FREE ROWS       1     FREE COLS      0     PASS2 (E ROWS)      0     PASS3      2     REMAINDER    26

 FACTORIZE     1    DEMAND      0    ITERATION      0    INFEAS       1    OBJECTV   0.000000000E-01
 SLACKS       27    LINEAR      2    NONLINEAR      0    ELEMS       70    DENSITY   8.32
 LROW         70    LCOL       70    LENL           0    LENU        70    INCRSE    0.00
 COMPRSSNS     0    MERIT     0.0    LMAX         0.0    UMAX   7.8E+00    GROWTH     1.0
 ITN    0 -- INFEASIBLE.  NUM =    1   SUM =   2.889086878E-01

   ITN PH PP NOPT  DJ,RG  +SBS -SBS  -BS  STEP     PIVOT        L     U NCP NINF  SINF,OBJECTIVE
     1  1  1    4 -7.7E+00    4    4    2 0.0E-01  1.1E+00     0    70   0    1  2.88908688E-01
     2  1  1    6 -1.5E+01   12   12   35 2.0E-02  1.5E+01     0    70   0    1  2.88908688E-01

 ITN    2 -- FEASIBLE SOLUTION.  OBJECTIVE =   4.303674377E-01
```

```
    3  2  1   2  1.2E+00   40   40   12 3.1E-02 -6.4E-01     1    83   0    0  4.66674543E-01
    4  2  1   3 -9.1E-01    2    2   34 4.7E-01  7.8E+00     1    70   0    0  8.99770302E-01
    5  2  1   2 -8.4E-01    3    3    4 2.9E-01  1.0E-01     3    83   0    0  1.14072090E+00
    6  2  1   1  1.2E+00   35   35   13 5.6E-01 -1.5E+00     6    95   0    0  1.80320632E+00
    7  2  1   2 -9.7E-01    4    4   17 6.7E-01  1.5E+00     7    95   0    0  2.45069218E+00
    8  2  1   1 -5.5E-01    6    6   36 3.7E-01 -3.4E+00     9   107   0    0  2.65296004E+00

 BIGGEST DJ = -5.675E-02 (VARIABLE    25)   NORM RG =  0.000E-01   NORM PI =  1.000E+00
1
 EXIT -- OPTIMAL SOLUTION FOUND.

 NO. OF ITERATIONS                         8      OBJECTIVE VALUE       2.6529600365290E+00


 NORM OF X                         1.135E+01      NORM OF PI                     1.000E+00


 BASIS MAP SAVED ON FILE  55    ITN =    8


 BASIS  PUNCHED  ON FILE  56
1
 PROBLEM NAME   SAMPLE P             OBJECTIVE VALUE    2.6529600365E+00


 STATUS         OPTIMAL SOLN         ITERATION   8     SUPERBASICS    0


 OBJECTIVE      OBJ      (MAX)
 RHS            RHS
 RANGES
 BOUNDS


 SECTION 1 - ROWS
```

NUMBER	...ROW..	STATE	...ACTIVITY...	SLACK ACTIVITY	..LOWER LIMIT.	..UPPER LIMIT.	.DUAL ACTIVITY	..I
13	DIF01001	UL	22.22600	0.00000	NONE	22.22600	-0.01098	1
14	DIF01002	BS	13.95642	4.19658	NONE	18.15300	0.00000	2
15	DIF01003	BS	11.63903	3.67197	NONE	15.31100	0.00000	3
16	DIF01004	BS	9.09877	2.35223	NONE	11.45100	0.00000	4
17	DIF02001	UL	33.77500	0.00000	NONE	33.77500	-0.05676	5
18	DIF02002	BS	21.97972	4.26728	NONE	26.24700	0.00000	6
19	DIF02003	BS	18.14874	3.36626	NONE	21.51500	0.00000	7
20	DIF02004	BS	13.89158	2.00142	NONE	15.89300	0.00000	8
21	HD010001	BS	81.25627	75.29373	NONE	156.55000	0.00000	9
22	HD010002	BS	72.13254	84.31746	NONE	156.45000	0.00000	10
23	HD010003	BS	64.44569	95.25431	NONE	159.70000	0.00000	11
24	HD010004	BS	70.43819	85.40181	NONE	155.84000	0.00000	12
25	HD010005	UL	185.88000	0.00000	NONE	185.88000	-0.00110	13
26	HD010006	BS	379.23743	-386.56963	-7.33220	NONE	0.00000	14
27	HD010007	BS	32.47382	-51.13782	-18.66400	NONE	0.00000	15
28	HD010008	BS	23.88915	-45.67215	-21.78300	NONE	0.00000	16
29	HD010009	BS	24.07354	-47.98954	-23.91600	NONE	0.00000	17
30	HD020001	BS	100.56284	73.15716	NONE	173.72000	0.00000	18
31	HD020002	BS	91.53432	81.91568	NONE	173.45000	0.00000	19
32	HD020003	BS	114.61925	68.13075	NONE	182.75000	0.00000	20
33	HD020004	BS	89.35787	83.11213	NONE	172.47000	0.00000	21
34	HD020005	UL	202.64000	0.00000	NONE	202.64000	-0.00137	22

35	HD020006	BS	489.62079	-470.36579	19.25500	NONE	0.00000	23
36	HD020007	LL	-2.17430	0.00000	-2.17430	NONE	0.00396	24
37	HD020008	BS	36.51745	-45.06725	-8.54980	NONE	0.00000	25
38	HD020009	BS	36.18865	-48.67165	-12.48300	NONE	0.00000	26
39	BAL1	BS	1.38768	-1.38768	0.00000	NONE	0.00000	27
40	BAL2	BS	1.26528	-1.26528	0.00000	NONE	0.00000	28
41	OBJ	BS	2.65296	-2.65296	NONE	NONE	1.00000	29

1

SECTION 2 - COLUMNS

NUMBER	.COLUMN.	STATE	...ACTIVITY...	.OBJ GRADIENT.	..LOWER LIMIT.	..UPPER LIMIT.	REDUCED GRADNT	M+J
1	Q01001	BS	0.41534	1.00000	0.00000	NONE	0.00000	30
2	Q02001	BS	0.40750	1.00000	0.00000	NONE	0.00000	31
3	Q01002	BS	0.97234	1.00000	0.00000	NONE	0.00000	32
4	Q02002	BS	1.16626	1.00000	0.00000	NONE	0.00000	33
5	Q01003	LL	0.00000	-1.00000	0.00000	NONE	-0.15368	34
6	Q02003	BS	0.30847	-1.00000	0.00000	NONE	0.00000	35
7	Q01004	LL	0.00000	-1.00000	0.00000	NONE	-0.40999	36
8	Q02004	LL	0.00000	-1.00000	0.00000	NONE	-0.08884	37
9	Q01005	LL	0.00000	-1.00000	0.00000	NONE	-0.53557	38
10	Q02005	LL	0.00000	-1.00000	0.00000	NONE	-0.41940	39
11	Q01006	LL	0.00000	-1.00000	0.00000	NONE	-0.58801	40
12	Q02006	LL	0.00000	-1.00000	0.00000	NONE	-0.54392	41

ENDRUN

APPENDIX V -- AQMAN PROGRAM LISTING

Modifications to subroutines that were originally part of the Trescott code include conversion to FORTRAN-77 and the specification of all variables as double precision. All other changes are denoted within the code with a "$" in column 73.

**NOTE:

Before executing AQMAN, make sure that the management problem does not require more CPU storage than is currently allocated by the specification statements at the start of each added AQMAN subroutine. It may be necessary to increase the size of arrays in the following COMMON blocks: RESP, WELLS, DIFFS, DEF, LOCATS, STRESS, PARAMS, TYP GRINFO, QUAD1. The user may also need to increase the size of the arrays HD and Q.

```
C
C*************************************************************
C                      A Q M A N                             *
C      COMPUTER CODE FOR AQUIFER MANAGEMENT PROBLEMS         *
C                                                            *
C           BY  S.M.GORELICK AND L.J.LEFKOFF                 *
C                 U.S. GEOLOGICAL SURVEY                     *
C                     OCTOBER, 1986                          *
C*************************************************************
C
C---Check Dimensions!!!!  Current limits are:
C---- NWLS = no. of wells = 200,
C---- NCNTR = no. of control nodes = 500,
C---- NNPER = no. of pumping periods = 50,
C---- NGRAD = no. of control pairs = 400,
C---- NNPER*NWLS = 2000
C---- NNPER*NCNTR = 2500
C---- NNPER*NGRAD = 1000
C---List of dimensioned variables with their minimum size:
C    HD(NNPER*NCNTR) = solved heads
C    Q(NNPER*NWLS) = stresses
C    HDUS(NNPER*NCNTR) = unstressed heads
C    CONHD(NNPER*NCNTR) = user's limits on head
C    DRDRES(NNPER*NCNTR) = drawdown responses
C    MNGDRD(NNPER*NCNTR) = manageable drawdown
C    UNITQ(NWLS) = unit stresses
C    FIXQ(NWLS,NNPER) = fixed stress at non-decision wells
C    KEYGRD(NGRAD) = key for gradient/velocity only
C    GRADE(NNPER*NGRAD) = manageable difference-in-drawdown
C    RDIF(NNPER*NGRAD) = difference-in-drawdown response
C    KDEFHD(NCNTR) = key for head definition
C    KDEFGR(NGRAD) = key for gradient or velocity definition
C    ILOCW(NWLS) = I-location of wells
C    JLOCW(NWLS) = J-location of wells
C    ILOCC(NCNTR) = I-location of control points
C    JLOCC(NCNTR) = J-location of control points
C    KEYQ(NWLS) = key for non-decision wells
C    KEYWL(NCNTR) = key for well at control location
C    XRAD(NWLS) = well radius
C    QWELL(NNPER*NWLS) = stress at wells
C    XHEAD(NNPER*NCNTR) = solved heads
C    TIMINC(NNPER) = length of initial times steps
C    CONTYP(NNPER*NCNTR) = type of constraint on head
C    GRATYP(NNPER*NGRAD) = type of constraint on gradient or velocity
C    SURF(NWLS) = land surface elevation at wells
C    COSTC(NWLS,NNPER) = unit pumping cost
C--The remaining variables appear only in Subroutine GRADS:
C    GFACT(NNPER*NGRAD) = conversion factor for gradient or velocity
C    GCON(NNPER*NGRAD) = user's constaint on gradient or velocity
C    ILOCG1(NGRAD) = I-location of first half of control pair
C    JLOCG1(NGRAD) = J-location of first half of control pair
C    ILOCG2(NGRAD) = I-location of second half of control pair
C    JLOCG2(NGRAD) = J-location of second half of control pair
```

```
C    KEEPC1(NNPER*NGRAD) = vector pointer within DRDRES
C    KEEPC2(NNPER*NGRAD) = vector pointer within DRDRES
C
C--Main program to generate LINEAR responses to unit pumping or
C----injection stresses.
      IMPLICIT REAL *8 (A-H,O-Z)
      REAL *8 MNGDRD
      INTEGER *4 CASE,TYPE,NNAME
      CHARACTER *1 CONTYP,GRATYP
      DIMENSION HD(2500),TYPE(4)
      COMMON /RESP/ HDUS(2500),CONHD(2500),DRDRES(2500),MNGDRD(2500)
      COMMON /WELLS/ UNITQ(200),FIXQ(200,50)
      COMMON /DIFFS/ NGRAD,KEYGRD(400),GRADE(1000),RDIF(1000)
      COMMON /DEF/ KDEFHD(500),KDEFGR(400)
      COMMON /LOCATS/ ILOCW(200),JLOCW(200),ILOCC(500),JLOCC(500),
     & KEYQ(200),KEYWL(500),XRAD(200)
      COMMON /STRESS/ QWELL(2000),XHEAD(2500)
      COMMON /PARAMS/ KCALLP,NWLS,NCNTR,NNPER,KPER,CDELT,TIMPER,
     & TIMINC(50)
      COMMON /TYP/ CONTYP(2500),GRATYP(1000)
      COMMON /ALPHAS/ NNAME(16),CASE
      DATA TYPE/'LINE','NONL','LINE','QUAD'/
C--Call PRE to compute transient, unmanaged (unstressed) heads:
      CALL PRE(1,Q,HD)
C--Store unstessed (includes boundary conditions and fixed wells)
C---head vector:
      NPER=NNPER
      WRITE(17,3)
    3 FORMAT(/19X,'CONTROL LOCATIONS AND UNSTRESSED HEADS'/6X,'Period',
     & 3x,'Location #  I-Loc.   J-Loc.   Unmanaged Head  KEYWL')
      DO 100 N=1,NNPER
       DO 100 I=1,NCNTR
       K=(N-1)*NCNTR+I
        HDUS(K)=HD(K)
        WRITE(17,5)K,N,I,ILOCC(I),JLOCC(I),HDUS(K),KEYWL(I)
    5   FORMAT(I5,I5,5X,I5,5X,I5,5X,I5,G20.9,I5)
  100 CONTINUE
C--If the management problem contains a quadratic objective function,
C---call QUAD to compute the cost coefficients for linear part of
C---objective:
      IF(CASE.EQ.TYPE(4)) CALL QUAD
C--Read user's desired limit on head and type of constraint at
C----each control location for all pumping periods:
      WRITE(17,15)
   15 FORMAT(//10X,'USER IMPOSED LIMITS ON HEAD AT CONTROL LOCATIONS'/,
     &3X,'Period',4X,'Loc. #  I-Location  J-Location ',4x,'Limit',
     &6X,'Type  KEYGRD  KDEFHD')
      DO 200 N=1,NPER
        DO 210 I=1,NCNTR
          K=(N-1)*NCNTR+I
          READ(14,20)CONHD(K),CONTYP(K)
   20     FORMAT(G10.0,4X,A1)
          IF(KDEFHD(I).EQ.0)WRITE(17,25)N,I,ILOCC(I),JLOCC(I),CONHD(K),
     &     CONTYP(K),KEYGRD(I),KDEFHD(I)
```

```
          IF(KDEFHD(I).NE.0)WRITE(17,26)N,I,ILOCC(I),JLOCC(I),CONHD(K),
     &     KEYGRD(I),KDEFHD(I)
   25     FORMAT(3X,I3,7X,I4,5X,I5,5X,I5,5X,G15.3,4X,A1,4X,I2,6X,I2)
   26     FORMAT(3X,I3,7X,I4,5X,I5,5X,I5,5X,G15.3,4X,'E',4X,I2,6X,I2)
  210   CONTINUE
  200 CONTINUE
C--Write well information:
      WRITE(17,30)
   30 FORMAT(//18X,'WELL LOCATIONS AND TYPE'/2X,'Period   Loc. #',4x,
     & 'I-Loc.   J-Loc.     KEYQ   Fixed or Unit Rate')
      DO 220 N=1,NNPER
      DO 220 I=1,NWLS
       IF(KEYQ(I).EQ.1)WRITE(17,35)N,I,ILOCW(I),JLOCW(I),KEYQ(I),
     &  FIXQ(I,N)
       IF(KEYQ(I).NE.1)WRITE(17,35)N,I,ILOCW(I),JLOCW(I),KEYQ(I),
     &  UNITQ(I)
   35  FORMAT(5(I5,5x),G10.4)
  220 CONTINUE
C--Compute the available managable drawdown at each control location
C----for all periods by subtracting the desired limits on heads
C----from the un-stressed heads.
      LENGTH=NPER*NCNTR
      DO 250 K=1,LENGTH
        MNGDRD(K)=HDUS(K)-CONHD(K)
  250 CONTINUE
C--If gradient controls are included, Call GRADS to adjust right-
C----hand-sides of gradient constraints by unstressed heads:
      IF(NGRAD.NE.0) CALL GRADS(1)
C--Call PRE to get the unit respose for each managed  well. Then
C----subtract this from the unstressed heads to obtain the drawdown
C----response vector, which is passed to MPSFMT:
      KCALL=1
      KMPS=0
      DO 300 IWLS=1,NWLS
        KCALL=KCALL+1
C--If the well is an unmanaged (fixed) well, do not call PRE, do
C----not store drawdown responses in the DRDRES vector, and do not
C----call MPSFMT:
        IF(KEYQ(IWLS).EQ.1)GO TO 320
        CALL PRE(KCALL,Q,HD)
        DO 310 K=1,LENGTH
          DRDRES(K)=HDUS(K)-HD(K)
  310   CONTINUE
        IF (NGRAD.NE.0) CALL GRADS(2)
        KMPS=KMPS+1
        CALL MPSFMT(KMPS,IWLS,1)
C--After PRE and MPSFMT have been called for the last well, call MPSFMT
C-----again to construct Right-Hand Side (RHS) of constraints:
  320   IF(IWLS.EQ.NWLS)CALL MPSFMT(KMPS,IWLS,2)
  300 CONTINUE
C
      STOP
      END
C-------------------------------------------------------------------
```

```
C--AQMAN's Main enters here. (For non-linear problem, optimization
C---code enters here; this in future versions of AQMAN).
C
C--- *** Check Dimensions!!  Current limits are listed at start of
C----- main program.
C
      SUBROUTINE PRE(KCALL,Q,HD)
      IMPLICIT REAL *8 (A-H,O-Z)
      INTEGER *4 CASE,TYPE,NNAME
      CHARACTER *1 CONTYP,GRATYP
      DIMENSION HD(2500),Q(2000),TYPE(4)
      COMMON /WELLS/ UNITQ(200),FIXQ(200,50)
      COMMON /DIFFS/ NGRAD,KEYGRD(400),GRADE(1000),RDIF(1000)
      COMMON /DEF/ KDEFHD(500),KDEFGR(400)
      COMMON /LOCATS/ ILOCW(200),JLOCW(200),ILOCC(500),JLOCC(500),
     & KEYQ(200),KEYWL(500),XRAD(200)
      COMMON /STRESS/ QWELL(2000),XHEAD(2500)
      COMMON /PARAMS/ KCALLP,NWLS,NCNTR,NNPER,KPER,CDELT,TIMPER,
     & TIMINC(50)
      COMMON /TYP/ CONTYP(2500),GRATYP(1000)
      COMMON /ALPHAS/ NNAME(16),CASE
      DATA TYPE/'LINE','NONL','LINE','QUAD'/
C--Read data from Unit 14 if this is first call of PRE.
      IF(KCALL.NE.1)GO TO 200
      READ(14,5)CASE,NNAME
    5 FORMAT(A4,2X,16A4)
  100 CONTINUE
C--Check that CASE has been correctly entered:
      IF(CASE.EQ.TYPE(3).OR.CASE.EQ.TYPE(4))GO TO 101
      WRITE(16,12)
   12 FORMAT(/2X,'PROGRAM TERMINATED BY SUBROUTINE PRE--CASE INCORRECTLY
     1 ENTERED'/)
      STOP
  101 CONTINUE
C--Read pumping period data:
      READ(14,15)NWLS,NCNTR,NNPER,CDELT,NGRAD
   15 FORMAT(3I10,F10.0,I10)
      READ(14,20)TIMPER
      READ(14,20)(TIMINC(N),N=1,NNPER)
   20 FORMAT(8G10.0)
C--Read data on wells.
      IKEYQ=0
      DO 150 I=1,NWLS
C---Read the number of fixed (unmanaged) wells.
  120    IF(I.EQ.1)READ(14,30)NKEYQ
   30    FORMAT(I10,F10.0)
C---For each well, read location, radius, KEYQ and UNITQ.
C----If KEYQ(I) equals 0 or 2, I is a managed  well.
C----If KEYQ(I) equals 1, I is an unmanaged   (fixed) well.
C----UNITQ(I) is the unit stress applied to (managed)  well I.
C----A positive value for UNITQ indicates recharge; negative UINTQ
C----indicates a pumping well.
C-*****WARNING**** Use care if UNITQ varies with I: Objective function
C---coefficients of MPS file will be scaled to compensate for variable
```

```
C---stresses.  Remember this when interpreting optimization results.
        READ(14,25)ILOCW(I),JLOCW(I),XRAD(I),KEYQ(I),UNITQ(I)
   25   FORMAT(2I10,F10.0,I10,F10.0)
        IF(KEYQ(I).EQ.0 .OR. KEYQ(I).EQ.2)GO TO 150
        IKEYQ=IKEYQ+1
        DO 130 N=1,NNPER
  130   READ(14,35)FIXQ(I,N)
   35   FORMAT(G10.0)
  150 CONTINUE
C--Check for non-uniform unit stresses
      ISW=0
      DO 160 I=1,NWLS
        IF(KEYQ(I).EQ.1)GO TO 160
        ISW=ISW+1
        USW2=UNITQ(I)
        IF(ISW.EQ.1)GO TO 155
        IF(USW2.NE.USW1)WRITE(16,32)
   32   FORMAT(//2X,'*****WARNING*** THE UNITQ VECTOR AS READ FROM UNIT
     114 IS NOT UNIFORM,'/'  SO DIFFERENT UNIT STRESSES WILL BE APPLIED
     2 AT SOME WELLS.'/1X,'THE OBJECTIVE FUNCTION COEFFICIENTS IN THE MP
     3S FILE WILL BE SCALED ACCORDINGLY.')
  155   USW1=UNITQ(I)
  160 CONTINUE
C--Check data on unmanaged wells:
      IF(IKEYQ.EQ.NKEYQ)go to 170
      WRITE(16,40)IKEYQ,NKEYQ
   40 FORMAT(/2X,'PROGRAM TERMINATED BY SUBROUTINE PRE--'/2X,'THE NUMBER
     1 OF UNIT KEYQ's,',I3,' ,DOES NOT EQUAL NKEYQ,',I3/)
      STOP
  170 CONTINUE
C--Read data on control locations.
      DO 180 I=1,NCNTR
        READ(14,45)ILOCC(I),JLOCC(I),KEYWL(I),KEYGRD(I),KDEFHD(I)
   45   FORMAT(5I10)
  180 CONTINUE
  200 CONTINUE
C--Compute QWELL (pumpage and injection rates) vector:
      IUNIT=KCALL-1
      IF(KCALL.NE.1.AND.KEYQ(IUNIT).EQ.1)WRITE(16,50)IUNIT
   50 FORMAT(/2X,'WARNING--PRE HAS BEEN CALLED BY AQMAN  FOR A NON-DECIS
     1ION WELL, WELL NUMBER',I4/)
  205 DO 210 N=1,NNPER
      DO 210 I=1,NWLS
        IN=(N-1)*NWLS+I
        QWELL(IN)=0.0
        IF(KEYQ(I).EQ.1)QWELL(IN)=FIXQ(I,N)
C--No unit stresses applied on unstressed or steay-state runs or in
C---non-first periods.
        IF(KCALL.LE.1 .OR. N.NE.1)GO TO 210
C--Enter unit stress into pumpage/injection vector.
        IF(I.EQ.IUNIT)QWELL(IUNIT)=UNITQ(I)
  210 CONTINUE
  240 CONTINUE
C--Initialize KPER:
```

```
      KPER=1
C--Call the TRESCOTT:
      CALL TRES(KCALL)
C--After TRESCOTT simulates all pumping/management periods, convert
C---XHEAD vector from "AQMAN1" Common to HD vector, which is passed
C---back to AQMAN's Main:
      KTOT=NCNTR*NNPER
      DO 300 K=1,KTOT
  300 HD(K)=XHEAD(K)
      RETURN
      END
C-----------------------------------------------------------------------
C--Check input data from unit 14 to be sure that control nodes
C----indicated to also be well nodes (by KEYWL not equal 0) are
C----indeed well nodes,having a non-zero well radius (for computation
C----of in-well head).
C--Also check for non-linearities introduced by problem specification.
      SUBROUTINE CHKDAT(NW,WATER,CONVRT,EVAP,LEAK)
      IMPLICIT REAL *8 (A-H,O-Z)
      INTEGER *4 CHK,WATER,CONVRT,EVAP,LEAK,CASE,TYPE,NNAME
      DIMENSION TYPE(4)
      COMMON /LOCATS/ ILOCW(200),JLOCW(200),ILOCC(500),JLOCC(500),
     & KEYQ(200),KEYWL(500),XRAD(200)
      COMMON /PARAMS/ KCALLP,NWLS,NCNTR,NNPER,KPER,CDELT,TIMPER,
     & TIMINC(50)
      COMMON /SARRAY/ VF4(11),CHK(15)
      COMMON /ALPHAS/ NNAME(16),CASE
      DATA TYPE/'LINE','NONL','LINE','QUAD'/
      KSUM=0
      DO 110 KC=1,NCNTR
        IF(KEYWL(KC).EQ.0)GO TO 110
        KSUM=KSUM+1
        IC=ILOCC(KC)
        JC=JLOCC(KC)
        DO 100 KW=1,NWLS
          KWM=KW
          IF(ILOCW(KW).NE.IC)GO TO 100
          IF(JLOCW(KW).NE.JC)GO TO 100
          KWM=KW-1
          IF(XRAD(KW).GT.0)GO TO 110
          WRITE(16,2)KW
          STOP
  100   CONTINUE
        IF(KWM.EQ.NWLS)WRITE(16,4)KC
        STOP
  110 CONTINUE
      IF(KSUM.EQ.NW)GO TO 120
      WRITE(16,6)KSUM,NW
      STOP
 120  CONTINUE
      IF(WATER.EQ.CHK(2) .OR. CONVRT.EQ.CHK(7))WRITE(16,12)
      IF(CONVRT.EQ.CHK(7))WRITE(16,12)
      IF(EVAP.EQ.CHK(6))WRITE(16,14)
      IF(LEAK.EQ.CHK(9))WRITE(16,16)
```

```
      RETURN
    2 FORMAT(/2X,'PROGRAM TERMINATED BY SUBROUTINE CHKDAT:'/2X,' WELL ',
     1I3,' IS SPECIFIED BY KEYWL TO BE A CONTROL LOCATION,'/3X,
     2'  BUT HAS A NON-POSITIVE WELL RADIUS'/)
    4 FORMAT(/2X,'PROGRAM TERMINATED BY SUBROUTINE CHKDAT:'/2X,'CONTROL
     1LOCATION',I3,' IS SPECIFIED BY KEYWL TO REQUIRE IN-WELL HEAD,'/
     23X,'  BUT NO WELL EXISTS THERE'/)
    6 FORMAT(/2X,'PROGRAM TERMINATED BY SUBROUTINE CHKDAT:'/2X,'THE NUMB
     1ER OF NON-ZERO KEYWLs,',I3,'   DOES NOT EQUAL NW,',I3/)
   12 FORMAT(//2X,'*****WARNING*** A NON-LINEARITY HAS BEEN INTRODUCED B
     1Y THE SPECIFICATION'/'   OF UNCONFINED CONDITIONS <from unit 15, l
     2ine 3>.'/'  ADDITION OF RESPONSE MATRICES BY A LINEAR PROGRAM MAY
     3GIVE ERRONEOUS RESULTS'/)
   14 FORMAT(//2X,'*****WARNING*** A NON-LINEARITY HAS BEEN INTRODUCED B
     1Y THE SPECIFICATION'/'   OF EVAPORTRANSPIRATION <from unit 15, lin
     2e 3>.'/2x,'ADDITION OF RESPONSE MATRICES BY A LINEAR PROGRAM MAY G
     3IVE ERRONEOUS RESULTS'/)
   16 FORMAT(//2X,'*****WARNING*** A NON-LINEARITY HAS BEEN INTRODUCED B
     1Y THE SPECIFICATION'/'   OF LEAKAGE CONDITIONS <from unit 15, line
     2 3>.'/2x,'ADDITION OF RESPONSE MATRICES BY A LINEAR PROGRAM MAY GI
     3VE ERRONEOUS RESULTS'/)
      END
C--------------------------------------------------------------------

C--Subroutine for control of head differences, gradients, or
C---velocities.
      SUBROUTINE GRADS(NENT)
      IMPLICIT REAL *8 (A-H,O-Z)
      REAL *8 MNGDRD
      CHARACTER *1 CONTYP,GRATYP
      COMMON /GRINFO/ KEEPC1(1000),KEEPC2(1000),ILOCG1(400),JLOCG1(400),
     &ILOCG2(400),JLOCG2(400),GFACT(1000),GCON(1000)
      COMMON /RESP/ HDUS(2500),CONHD(2500),DRDRES(2500),MNGDRD(2500)
      COMMON /DIFFS/ NGRAD,KEYGRD(400),GRADE(1000),RDIF(1000)
      COMMON /DEF/ KDEFHD(500),KDEFGR(400)
      COMMON /LOCATS/ ILOCW(200),JLOCW(200),ILOCC(500),JLOCC(500),
     & KEYQ(200),KEYWL(500),XRAD(200)
      COMMON /PARAMS/ KCALLP,NWLS,NCNTR,NNPER,KPER,CDELT,TIMPER,
     & TIMINC(50)
      COMMON /TYP/ CONTYP(2500),GRATYP(1000)
      IF (NENT.EQ.2) GO TO 300
C--Read and Write the I and J location of the two nodes of each
C---head-difference control pair.
      WRITE(17,10)
   10 FORMAT(//13X,'CONTROL PAIR LOCATIONS AND DEFINITIONS'/2X,
     & 'Pair #   1st I-Loc.  1st J-Loc.   2nd I-Loc.  2nd J-Loc.    '
     & 'KDEFGR')
      DO 100 NG=1,NGRAD
      READ(13,20)ILOCG1(NG),JLOCG1(NG),ILOCG2(NG),JLOCG2(NG),KDEFGR(NG)
   20 FORMAT(5I10)
  100 WRITE(17,25)NG,ILOCG1(NG),JLOCG1(NG),ILOCG2(NG),JLOCG2(NG),
     & KDEFGR(NG)
   25 FORMAT(3X,I3,7X,I3,8X,I3,11X,I3,9X,I3,9X,I2)
      WRITE(17,30)
```

```
   30 FORMAT(//1X,'USER IMPOSED LIMITS ON HEAD DIFFRENCE AT CONTROL '
     & 'PAIRS'/2X,'Period   Pair #     Conversion Factor    Difference '
     & 'Limit    Type')
      DO 120 N=1,NNPER
      DO 120 NG=1,NGRAD
       IG=NG+(N-1)*NGRAD
C--Read the factor to convert a gradient or velocity to a head
C---difference, and read the value and direction of the desired limit
C---on head difference, gradient, or velocity for all periods:
      READ(13,35)GFACT(IG),GCON(IG),GRATYP(IG)
   35 FORMAT(2G15.6,4X,A1)
      IF(KDEFGR(NG).EQ.0)WRITE(17,37)N,NG,GFACT(IG),GCON(IG),GRATYP(IG)
      IF(KDEFGR(NG).NE.0)WRITE(17,38)N,NG,GFACT(IG),GCON(IG)
   37 FORMAT(2X,I3,7X,I3,8X,G15.6,4X,G15.6,6X,A1)
   38 FORMAT(2X,I3,7X,I3,8X,G15.6,4X,G15.6,6X,'E')
  120 CONTINUE
C--Use the GFACT, GCON and HDUS vectors to compute the head difference
C---to be used as the RHS of the constraints in the MPS file and
C---optimization model.
      DO 280 N=1,NNPER
        DO 270 NGD=1,NGRAD
          NG=NGD+(N-1)*NGRAD
          IC=1
  210     IF(ILOCG1(NGD).EQ.ILOCC(IC))GO TO 230
  220     IC=IC+1
          IF(IC.LE.NCNTR) GO TO 210
          WRITE(16,40)NGD
   40     FORMAT(/2X,'EXECUTION TERMINATED BY GRADS--',2X,'LOCATION OF F
     1IRST NODE IN CONTROL PAIR',I4,' IS NOT ALSO A PRIMARY CONTROL LOCA
     2TION'/)
          STOP
  230     IC1=IC
          IF(JLOCG1(NGD).NE.JLOCC(IC1))GO TO 220
          KEEPC1(NG)=(N-1)*NCNTR+IC1
          IC=1
  240     IF(ILOCG2(NGD).EQ.ILOCC(IC))GO TO 260
  250     IC=IC+1
          IF(IC.LE.NCNTR) GO TO 240
          WRITE(16,45)NGD
   45     FORMAT(/2X,'EXECUTION TERMINATED BY GRADS--',2X,'LOCATION OF S
     1ECOND NODE IN CONTROL PAIR',I4,' IS NOT ALSO A PRIMARY CONTROL LOC
     2ATION'/)
          STOP
  260     IC2=IC
          IF(JLOCG2(NGD).NE.JLOCC(IC2))GO TO 250
          KEEPC2(NG)=(N-1)*NCNTR+IC2
C--Head difference between locations 1 & 2 is defined as
C---head(1)-head(2):
          KCON1=(N-1)*NCNTR+IC1
          KCON2=(N-1)*NCNTR+IC2
C--GRADE is vector of size = NPER*NGRAD.
C--If constraint is a definition use unmanaged gradients only -- This
C---used at lines MPS3060 to MPS3240
          GRADE(NG)=HDUS(KCON1)-HDUS(KCON2)-(GFACT(NG)*GCON(NG))
```

```
        IF(KDEFGR(NG).EQ.1)GRADE(NG)=HDUS(KCON1)-HDUS(KCON2)
  270   CONTINUE
  280 CONTINUE
      RETURN
  300 CONTINUE
      DO 320 N=1,NNPER
        DO 310 NGD=1,NGRAD
          NG=NGD+(N-1)*NGRAD
          KDRD1=KEEPC1(NG)
          KDRD2=KEEPC2(NG)
C--Difference in drawdown response to pumping between locations 1 & 2
C----is defined as DRDRES(1)-DRDRES(2):
          RDIF(NG)=DRDRES(KDRD1)-DRDRES(KDRD2)
  310   CONTINUE
  320 CONTINUE
      RETURN
      END
C-------------------------------------------------------------------
C---This subroutine takes drawdown response coefficients and converts
C-----them to MPS format. Response functions are repeated and staggered
C-----as necessary to simulate multiple pumping periods. It is assumed
C-----that the response functions are contained in a vector, ordered
C-----by pumping well, then observation time (pumping period), then
C-----control location.
C-----NWLS=no. of pumping wells;
C-----NCNTR=no. of control locations/observation wells;
C-----NPER=no. of pumping periods;
C-----DRDRES=vector of responses
C-----MNGDRD=vector of constraints on drawdown
      SUBROUTINE MPSFMT(KMPS,IWLS,NENT)
      IMPLICIT REAL *8 (A-H,O-Z)
      REAL *8 MNGDRD
      INTEGER *4 IROW,IROW1,IROW2,NNAME,CASE,TYPE
      CHARACTER *1 CONTYP,GRATYP
      DIMENSION TYPE(4)
      COMMON /RESP/ HDUS(2500),CONHD(2500),DRDRES(2500),MNGDRD(2500)
      COMMON /WELLS/ UNITQ(200),FIXQ(200,50)
      COMMON /DIFFS/ NGRAD,KEYGRD(400),GRADE(1000),RDIF(1000)
      COMMON /DEF/ KDEFHD(500),KDEFGR(400)
      COMMON /LOCATS/ ILOCW(200),JLOCW(200),ILOCC(500),JLOCC(500),
     & KEYQ(200),KEYWL(500),XRAD(200)
      COMMON /PARAMS/ KCALLP,NWLS,NCNTR,NNPER,KPER,CDELT,TIMPER,
     & TIMINC(50)
      COMMON /QUAD1/ SURF(200),COSTC(200,50)
      COMMON /TYP/ CONTYP(2500),GRATYP(1000)
      COMMON /ALPHAS/ NNAME(16),CASE
      DATA TYPE/'LINE','NONL','LINE','QUAD'/
      NPER=NNPER
C--If this is final call of MPSFMT, skip to RHS section:
      IF(NENT.EQ.2)GO TO 400
C--If this is not first call of MPSFMT, skip to COLUMN section:
      IF(KMPS.NE.1)GO TO 150
C--Check for too many digits in number of observation sites and
C---periods:
```

```
      IF(NCNTR.LE.9999)GO TO 5
      WRITE(16,1705)
 1705 FORMAT(/2X,'NUMBER OF OBSERVATION WELLS EXCEEDS 9999--PROGRAM TERM
     1INATED BY SUBROUTINE MPSFMT'/)
      STOP
    5 CONTINUE
      IF(NPER.LE.99)GO TO 10
      WRITE(16,1710)
 1710 FORMAT(/2X,'NUMBER OF PUMPING PERIODS EXCEEDS 99--PROGRAM TERMINAT
     1ED BY SUBROUTINE MPSFMT'/)
      STOP
   10 CONTINUE
C---Check for too many digits in number of pumping/injection sites:
      IF(NWLS.LE.999)GO TO 15
      WRITE(16,1715)
 1715 FORMAT(/2X,'NUMBER OF PUMPING WELLS EXCEEDS 999--PROGRAM TERMINATE
     1D BY SUBROUTINE MPSFMT'/)
      STOP
   15 CONTINUE
C---Write problem name:
      WRITE(18,1720)NNAME
 1720 FORMAT('NAME',10X,16A4)
C---Write row section:
      WRITE(18,1725)
 1725 FORMAT('ROWS')
      IF(NGRAD.EQ.0)GO TO 122
C---Write row names for gradient pairs:
      DO 120 KT=1,NPER
      DO 120 KG=1,NGRAD
       IG=KG+(KT-1)*NGRAD
       IROW=1000*KT+KG
       IF(KDEFGR(KG).EQ.1)GO TO 105
       IF(KDEFGR(KG).GT.1 .OR. KDEFGR(KG).LT.0)GO TO 118
       IF(GRATYP(IG).NE.'L')GO TO 100
       IF(KT.LE.9)WRITE(18,1731)IROW
       IF(KT.GE.10)WRITE(18,1732)IROW
 1731  FORMAT(' G  ','DIF0',I4)
 1732  FORMAT(' G  ','DIF',I5)
       GO TO 120
  100  IF(GRATYP(IG).NE.'E')GO TO 110
  105  IF(KT.LE.9)WRITE(18,1733)IROW
       IF(KT.GE.10)WRITE(18,1734)IROW
 1733  FORMAT(' E  ','DIF0',I4)
 1734  FORMAT(' E  ','DIF',I5)
       GO TO 120
  110  IF(GRATYP(IG).NE.'G')GO TO 115
       IF(KT.LE.9)WRITE(18,1735)IROW
       IF(KT.GE.10)WRITE(18,1736)IROW
 1735  FORMAT(' L  ','DIF0',I4)
 1736  FORMAT(' L  ','DIF',I5)
       GO TO 120
  115  WRITE(16,1737)KG,KT
 1737  FORMAT(2X,'PROGRAM EXECUTION TERMINATED --'/4X,'CONSTRAINT TYPE
     &AT CONTROL PAIR',I4,' DURING PERIOD',I3,' IS NEITHER L,E, NOR G')
```

```
      STOP
  118 WRITE(16,1740)KG,KT
 1740 FORMAT(2X,'PROGRAM EXECUTION TERMINATED --'/4X,'KDEFGR AT '
    & 'CONTROL PAIR',I4,' DURING PERIOD',I3,' IS IMPROPERLY DEFINED')
      STOP
  120 CONTINUE
  122 CONTINUE
C---Write row names for individual control locations.  Direction of
C----inequalities is reversed since constraints are expressed
C----in terms of drawdown.
      DO 140 KT=1,NPER
      DO 140 KW=1,NCNTR
       IF(NGRAD.EQ.0)GO TO 125
       IF(KEYGRD(KW).EQ.1)GO TO 140
  125  CONTINUE
       IT=KW+(KT-1)*NCNTR
       IROW=10000*KT+KW
       IF(KDEFHD(KW).EQ.1)GO TO 130
       IF(KDEFHD(KW).LT.0 .OR. KDEFHD(KW).GT.1)GO TO 138
       IF(CONTYP(IT).NE.'L')GO TO 128
       IF(KT.LE.9)WRITE(18,1741)IROW
       IF(KT.GE.10)WRITE(18,1742)IROW
 1741  FORMAT(' G  ','DRO',I5)
 1742  FORMAT(' G  ','DR',I6)
       GO TO 140
  128  IF(CONTYP(IT).NE.'E')GO TO 132
  130  IF(KT.LE.9)WRITE(18,1743)IROW
       IF(KT.GE.10)WRITE(18,1744)IROW
 1743  FORMAT(' E  ','DRO',I5)
 1744  FORMAT(' E  ','DR',I6)
       GO TO 140
  132  IF(CONTYP(IT).NE.'G')GO TO 135
       IF(KT.LE.9)WRITE(18,1745)IROW
       IF(KT.GE.10)WRITE(18,1746)IROW
 1745  FORMAT(' L  ','DRO',I5)
 1746  FORMAT(' L  ','DR',I6)
       GO TO 140
  135  WRITE(16,1747)KW,KT
 1747  FORMAT(2X,'PROGRAM EXECUTION TERMINATED --'/4X,'CONSTRAINT TYPE
     &AT CONTROL LOCATION',I4,' DURING PERIOD',I3,' IS NEITHER L,E, NOR
     &G')
       STOP
  138  WRITE(16,1749)KW,KT
 1749  FORMAT(2X,'PROGRAM EXECUTION TERMINATED --'/4X,'KDEFHD AT '
     & 'CONTROL LOCATION',I4,' DURING PERIOD',I3,' IS IMPROPERLY DEFIN
     &ED')
       STOP
  140 CONTINUE
C---End row section with objective row:
      WRITE(18,1750)
 1750 FORMAT(' N ',' OBJ')
C---Start column section:
      WRITE(18,1755)
 1755 FORMAT('COLUMNS')
```

```
  150 CONTINUE
C---Write all responses (by location and by time) from pumping well 1
C-----for all time periods, and then responses from next pumping well
C-----for all time periods, etc.
C---Row DR010006 is drawdown (response) during pumping period 1 at
C-----control location 6;  Row DIF03012 is difference-in-drawdown
C-----response during period 3 at gradient pair 12.  Column Q02003 is
C-----pumping period 2, pumping well 3.
      DO 350 IPER=1,NPER
       KILOPR=IPER*1000
       NUMWEL=KMPS+KILOPR
C---Write scaled objective coefficient into column section for linear
C----objective function:
       IF(CASE.EQ.TYPE(4)) GO TO 160
       COEFF=-UNITQ(IWLS)
       IF(IPER.LE.9)WRITE(18,1760)NUMWEL,COEFF
       IF(IPER.GE.10)WRITE(18,1765)NUMWEL,COEFF
 1760  FORMAT(4X,'Q0',I4,4X,'OBJ',7X,G12.5)
 1765  FORMAT(4X,'Q',I5,4X,'OBJ',7X,G12.5)
       IF(NGRAD.EQ.0)GO TO 300
       GO TO 190
C---Write objective into column section for linear part of quadratic
C----objective function:
  160  SCOST=COSTC(KMPS,IPER)*(-UNITQ(KMPS))
       IF(IPER.LE.9)WRITE(18,1767)NUMWEL,SCOST
       IF(IPER.GE.10)WRITE(18,1768)NUMWEL,SCOST
 1767  FORMAT(4X,'Q0',I4,4X,'OBJ',7X,G12.5)
 1768  FORMAT(4X,'Q',I5,4X,'OBJ',7X,G12.5)
       IF(NGRAD.EQ.0)GO TO 300
C---Write column section for gradient control pairs:
  190  DO 230 KT=IPER,NPER
       DO 230 KG=1,NGRAD
        IF(MOD(KG,2).EQ.1)GO TO 210
        IROW2=1000*KT+KG
        KRES2=KG+(KT-1)*NGRAD-(IPER-1)*NGRAD
        IF(IPER.GE.10)GO TO 205
        IF(KT.LE.9)WRITE(18,1770)NUMWEL,IROW1,RDIF(KRES1),IROW2,
     &   RDIF(KRES2)
        IF(KT.GE.10)WRITE(18,1775)NUMWEL,IROW1,RDIF(KRES1),IROW2,
     &   RDIF(KRES2)
 1770   FORMAT(4X,'Q0',I4,4X,'DIF0',I4,2X,E12.5,3X,'DIF0',I4,2X,E12.5)
 1775   FORMAT(4X,'Q0',I4,4X,'DIF',I5,2X,E12.5,3X,'DIF',I5,2X,E12.5)
        GO TO 230
  205   IF(KT.LE.9)WRITE(18,1780)NUMWEL,IROW1,RDIF(KRES1),IROW2,
     &   RDIF(KRES2)
        IF(KT.GE.10)WRITE(18,1785)NUMWEL,IROW1,RDIF(KRES1),IROW2,
     &   RDIF(KRES2)
 1780   FORMAT(4X,'Q',I5,4X,'DIF0',I4,2X,E12.5,3X,'DIF0',I4,2X,E12.5)
 1785   FORMAT(4X,'Q',I5,4X,'DIF',I5,2X,E12.5,3X,'DIF',I5,2X,E12.5)
        GO TO 230
  210   IROW1=1000*KT+KG
        KRES1=KG+(KT-1)*NGRAD-(IPER-1)*NGRAD
        IF(KG.NE.NGRAD)GO TO 230
C---Write single last value if it exists:
```

```
      IF(IPER.GT.9)GO TO 220                                            MPS1990
      IF(KT.LE.9)WRITE(18,1790)NUMWEL,IROW1,RDIF(KRES1)                 MPS2000
      IF(KT.GE.10)WRITE(18,1795)NUMWEL,IROW1,RDIF(KRES1)                MPS2010
 1790 FORMAT(4X,'Q0',I4,4X,'DIF0',I4,2X,E12.5)                          MPS2020
 1795 FORMAT(4X,'Q0',I4,4X,'DIF',I5,2X,E12.5)                           MPS2030
      GO TO 230                                                         MPS2040
  220 IF(KT.LE.9)WRITE(18,1805)NUMWEL,IROW1,RDIF(KRES1)                 MPS2050
      IF(KT.GE.10)WRITE(18,1810)NUMWEL,IROW1,RDIF(KRES1)                MPS2060
 1805 FORMAT(4X,'Q',I5,4X,'DIF0',I4,2X,E12.5)                           MPS2070
 1810 FORMAT(4X,'Q',I5,4X,'DIF',I5,2X,E12.5)                            MPS2080
  230 CONTINUE                                                          MPS2090
  300 CONTINUE                                                          MPS2100
C---Write column section for individual control locations.              MPS2110
      DO 340 KT=IPER,NPER                                               MPS2120
       ISNGL=0                                                          MPS2130
       KW=0                                                             MPS2140
       DO 330 KKW=1,NCNTR                                               MPS2150
  305   KW=KW+1                                                         MPS2160
        IF(KW.GT.NCNTR .AND. ISNGL.EQ.0)GO TO 340                       MPS2170
        IF(KW.GT.NCNTR .AND. ISNGL.EQ.1)GO TO 322                       MPS2180
        IF(KEYGRD(KW).EQ.1)GO TO 305                                    MPS2190
        IF(MOD(KKW,2).EQ.1)GO TO 320                                    MPS2200
        ISNGL=0                                                         MPS2210
        IROW2=10000*KT+KW                                               MPS2220
        KRES2=KW+(KT-1)*NCNTR-(IPER-1)*NCNTR                            MPS2230
        IF(IPER.GE.10)GO TO 315                                         MPS2240
        IF(KT.LE.9)WRITE(18,1815)NUMWEL,IROW1,DRDRES(KRES1),IROW2,      MPS2250
     &   DRDRES(KRES2)                                                  MPS2260
        IF(KT.GE.10)WRITE(18,1820)NUMWEL,IROW1,DRDRES(KRES1),IROW2,     MPS2270
     &   DRDRES(KRES2)                                                  MPS2280
 1815   FORMAT(4X,'Q0',I4,4X,'DR0',I5,2X,E12.5,3X,'DR0',I5,2X,E12.5)    MPS2290
 1820   FORMAT(4X,'Q0',I4,4X,'DR',I6,2X,E12.5,3X,'DR',I6,2X,E12.5)      MPS2300
        GO TO 330                                                       MPS2310
  315   IF(KT.LE.9)WRITE(18,1825)NUMWEL,IROW1,DRDRES(KRES1),IROW2,      MPS2320
     &   DRDRES(KRES2)                                                  MPS2330
        IF(KT.GE.10)WRITE(18,1830)NUMWEL,IROW1,DRDRES(KRES1),IROW2,     MPS2340
     &   DRDRES(KRES2)                                                  MPS2350
 1825   FORMAT(4X,'Q',I5,4X,'DR0',I5,2X,E12.5,3X,'DR0',I5,2X,E12.5)     MPS2360
 1830   FORMAT(4X,'Q',I5,4X,'DR',I6,2X,E12.5,3X,'DR',I6,2X,E12.5)       MPS2370
        GO TO 330                                                       MPS2380
  320   IROW1=10000*KT+KW                                               MPS2390
        KRES1=KW+(KT-1)*NCNTR-(IPER-1)*NCNTR                            MPS2400
        ISNGL=1                                                         MPS2410
        IF(KW.NE.NCNTR)GO TO 330                                        MPS2420
C---Write single last value if it exists:                               MPS2430
  322   IF(IPER.GT.9)GO TO 325                                          MPS2440
        IF(KT.LE.9)WRITE(18,1835)NUMWEL,IROW1,DRDRES(KRES1)             MPS2450
        IF(KT.GE.10)WRITE(18,1840)NUMWEL,IROW1,DRDRES(KRES1)            MPS2460
 1835   FORMAT(4X,'Q0',I4,4X,'DR0',I5,2X,E12.5)                         MPS2470
 1840   FORMAT(4X,'Q0',I4,4X,'DR',I6,2X,E12.5)                          MPS2480
        GO TO 340                                                       MPS2490
  325   IF(KT.LE.9)WRITE(18,1845)NUMWEL,IROW1,DRDRES(KRES1)             MPS2500
        IF(KT.GE.10)WRITE(18,1850)NUMWEL,IROW1,DRDRES(KRES1)            MPS2510
 1845   FORMAT(4X,'Q',I5,4X,'DR0',I5,2X,E12.5)                          MPS2520
```

```
 1850    FORMAT(4X,'Q',I5,4X,'DR',I6,2X,E12.5)
         GO TO 340
  330   CONTINUE
  340  CONTINUE
C---If objective function is quadratic, write responses at control
C----locations that are also managed  wells to unit 19.
       IF(CASE.NE.TYPE(4) .OR. IPER.GT.1) GO TO 350
       DO 345 KTQ=1,NPER
       DO 345 KC=1,NCNTR
        IF(KEYWL(KC).NE.1)GO TO 345
        KRES=KC+(KTQ-1)*NCNTR
        WRITE(19,1852)KMPS,KC,KTQ,DRDRES(KRES)
 1852   FORMAT(6X,I4,2I5,F15.7)
  345  CONTINUE
  350 CONTINUE
C---Return to MAIN program to obtain unit response for next well.
      RETURN
C---If constraints are definitions (KDEFHD =1 and/or KDEFGR=1) add
C---Identity matrix to define H as Drawdown + Unstressed heads,
C---and/or defines G as Diff. in Drawdown + Unstressed Diff. in Head
C---Entry point is line 400 if all response info. (rest of COLUMN
C---section has been written.
  400 CONTINUE
C---Write columns section DEFINING gradient variables
      ONE=1.0D0
      IF(NGRAD.EQ.0)GO TO 540
      DO 530 KT=1,NPER
       DO 520 IG=1,NGRAD
        IF(KDEFGR(IG).NE.1) GO TO 520
        IROW1=1000*KT+IG
C---Write single value
  510   IF(KT.LE.9)WRITE(18,1423)IROW1,IROW1,ONE
        IF(KT.GE.10)WRITE(18,1424)IROW1,IROW1,ONE
 1423   FORMAT(4X,'G0',I4,4X,'DIF0',I4,2X,E12.5)
 1424   FORMAT(4X,'G',I5,4X,'DIF',I5,2X,E12.5)
  520  CONTINUE
  530  CONTINUE
C---Write DEFS for for individual control locations.
  540 DO 580 KT=1,NPER
       DO 570 KKW=1,NCNTR
        IF(KDEFHD(KKW).NE.1)GO TO 570
  560    IROW1=10000*KT+KKW
C---Write single HEAD DEF
  565    IF(KT.LE.9)WRITE(18,1435)IROW1,IROW1,ONE
         IF(KT.GE.10)WRITE(18,1440)IROW1,IROW1,ONE
 1435    FORMAT(4X,'H0',I5,3X,'DR0',I5,2X,E12.5)
 1440    FORMAT(4X,'H',I6,3X,'DR',I6,2X,E12.5)
  570  CONTINUE
  580 CONTINUE
C---Write Right-Hand-Side section:
  600 WRITE(18,1855)
 1855 FORMAT('RHS')
C--RHS for gradient pairs.
C--For gradient definitions (KDEFGR=1), GRADE is defined as unstressed
```

```
C---gradient without any user-defined constraint.  See line GRA 700
      IF(NGRAD.EQ.0)GO TO 640
      DO 630 KT=1,NPER
       DO 620 KG=1,NGRAD
        IF(MOD(KG,2).EQ.1)GO TO 610
        IROW2=1000*KT+KG
        KCON2=KG+(KT-1)*NGRAD
        IF(KT.LE.9)WRITE(18,1860)IROW1,GRADE(KCON1),IROW2,GRADE(KCON2)
        IF(KT.GE.10)WRITE(18,1865)IROW1,GRADE(KCON1),IROW2,GRADE(KCON2)
 1860   FORMAT(4X,'RHS',7X,'DIF0',I4,2X,E12.5,3X,'DIF0',I4,2X,E12.5)
 1865   FORMAT(4X,'RHS',7X,'DIF',I5,2X,E12.5,3X,'DIF',I5,2X,E12.5)
        GO TO 620
  610   IROW1=1000*KT+KG
        KCON1=KG+(KT-1)*NGRAD
        IF(KG.NE.NGRAD)GO TO 620
        IF(KT.LE.9)WRITE(18,1870)IROW1,GRADE(KCON1)
        IF(KT.GE.10)WRITE(18,1875)IROW1,GRADE(KCON1)
 1870   FORMAT(4X,'RHS',7X,'DIF0',I4,2X,E12.5)
 1875   FORMAT(4X,'RHS',7X,'DIF',I5,2X,E12.5)
  620  CONTINUE
  630 CONTINUE
  640 CONTINUE
C---Write RHS for individual control locations.
      DO 680 KT=1,NPER
       ISNGL=0
       KW=0
       DO 670 KKW=1,NCNTR
  650   KW=KW+1
        IF(KW.GT.NCNTR .AND. ISNGL.EQ.0)GO TO 680
        IF(KW.GT.NCNTR .AND. ISNGL.EQ.1)GO TO 665
        IF(KEYGRD(KW).EQ.1)GO TO 650
        IF(MOD(KKW,2).EQ.1)GO TO 660
        ISNGL=0
        IROW2=10000*KT+KW
        KCON2=KW+(KT-1)*NCNTR
        IF(KDEFHD(KW).EQ.1)X2=HDUS(KCON2)
        IF(KDEFHD(KW).NE.1)X2=MNGDRD(KCON2)
        IF(KT.LE.9)WRITE(18,1880)IROW1,X1,IROW2,X2
        IF(KT.GE.10)WRITE(18,1885)IROW1,X1,IROW2,X2
 1880   FORMAT(4X,'RHS',7X,'DR0',I5,2X,E12.5,3X,'DR0',I5,2X,E12.5)
 1885   FORMAT(4X,'RHS',7X,'DR',I6,2X,E12.5,3X,'DR',I6,2X,E12.5)
        GO TO 670
  660   IROW1=10000*KT+KW
        KCON1=KW+(KT-1)*NCNTR
        ISNGL=1
        IF(KDEFHD(KW).EQ.1)X1=HDUS(KCON1)
        IF(KDEFHD(KW).NE.1)X1=MNGDRD(KCON1)
        IF(KW.NE.NCNTR)GO TO 670
  665   IF(KT.LE.9)WRITE(18,1890)IROW1,X1
        IF(KT.GE.10)WRITE(18,1895)IROW1,X1
 1890   FORMAT(4X,'RHS',7X,'DR0',I5,2X,E12.5)
 1895   FORMAT(4X,'RHS',7X,'DR',I6,2X,E12.5)
        GO TO 680
  670  CONTINUE
```

```
  680 CONTINUE
C---Write end of data
      WRITE(18,1905)
 1905 FORMAT('ENDATA')
      RETURN
      END
C-----------------------------------------------------------------------
C--Subroutine to compute coefficients for linear part of quadratic
C----objective function.
      SUBROUTINE QUAD
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL *8 MNGDRD
      CHARACTER *1 CONTYP,GRATYP
      COMMON /RESP/ HDUS(2500),CONHD(2500),DRDRES(2500),MNGDRD(2500)
      COMMON /LOCATS/ ILOCW(200),JLOCW(200),ILOCC(500),JLOCC(500),
     & KEYQ(200),KEYWL(500),XRAD(200)
      COMMON /PARAMS/ KCALLP,NWLS,NCNTR,NNPER,KPER,CDELT,TIMPER,
     & TIMINC(50)
      COMMON /QUAD1/ SURF(200),COSTC(200,50)
      COMMON /TYP/ CONTYP(2500),GRATYP(1000)
C--Check data to see that quadratic managed  wells are listed as first
C----managed  wells.
      IORD=0
      DO 85 I=1,NWLS
       IF(KEYQ(I).EQ.2)IORD=2
       IF(KEYQ(I).NE.0)GO TO 85
       IF(IORD.EQ.0)GO TO 85
       WRITE(16,84)I
  84   FORMAT(6X,'PROGRAM TERMINATED BY SUBROUTINE QUAD--'/2X,
     1'WELL NUMBER ',I3,' SHOWS THAT ALL QUADRATIC DECISION WELLS ARE'/
     22X,' NOT LISTED BEFORE ALL NON-QUADRATIC DECISION WELLS')
       STOP
  85  CONTINUE
C--Compute number of quadratic managed  wells and check to see that
C----these are listed as first control locations.
      KWLS=0
      IORD=0
      DO 100 I=1,NCNTR
       IF(I.NE.1)GO TO 90
       IF(KEYWL(1).EQ.1)IORD=1
   90  IF(KEYWL(I).NE.1)GO TO 95
       KWLS=KWLS+1
       IF(IORD.EQ.1)GO TO 100
       WRITE(16,91)I
   91  FORMAT(2X,'PROGRAM TERMINATED BY SUBROUTINE QUAD--'/3X,'CONTROL '
     1'LOCATION',I4,' SHOWS THAT ALL QUADRATIC DECISION WELLS ARE NOT '
     2/3X,'LISTED SEQUENTIALLY AS THE FIRST CONTROL LOCATIONS')
       STOP
   95  IORD=2
  100 CONTINUE
C--Read annual discount rate, constant pumping cost, and key to
C----indicate whether costs are constant (0) or variable (1):
      READ(13,10)ANNDIS,COSTF,KEYCOS
   10 FORMAT(G10.3,G10.0,I10)
```

```
      IF(KEYCOS.EQ.0) GO TO 110
C--Read variable pumping costs into cost vector:
      DO 105 N=1,NNPER
        READ(13,15)(COSTC(I,N),I=1,KWLS)
   15   FORMAT(8G10.0)
  105 CONTINUE
      GO TO 130
C--Put constant pumping costs into costs vector:
  110 DO 120 N=1,NNPER
        DO 120 I=1,KWLS
          COSTC(I,N)=COSTF
  120 CONTINUE
C--Discount pumping costs according to management period lengths by
C---months:
  130 DISMON=ANNDIS/12.0
      NDAYS=0
      DO 140 N=1,NNPER
        NDAYS=NDAYS+TIMPER
        NMONTH=NDAYS/30
        DISFAC=1./((1.+DISMON)**NMONTH)
        DO 140 I=1,KWLS
          COSTC(I,N)=DISFAC*COSTC(I,N)
  140 CONTINUE
C--Read elevation of land surface at all quadratic decision wells:
      READ(13,20)(SURF(I),I=1,KWLS)
   20 FORMAT(8G10.0)
C--Compute pumping lifts under unmanaged  conditions:
      DO 170 N=1,NNPER
       KK=0
       LBEG=(N-1)*NCNTR+1
       LEND=N*NCNTR
       DO 170 IW=1,NWLS
        KC=0
        IF(KEYQ(IW).EQ.0)GO TO 145
        KK=KK+1
        GO TO 170
  145   DO 150 L=LBEG,LEND
         KC=KC+1
         IF(ILOCW(IW).EQ.ILOCC(KC).AND.JLOCW(IW).EQ.JLOCC(KC))GO TO 160
         IF(L.LT.LEND) GO TO 150
         WRITE(16,50)IW
   50    FORMAT(/2X,'TERMINATION BY SUBROUTINE QUAD ---',/,' THE LOCATIO
     1N OF WELL',I4,' DOES NOT CORRESPOND TO A CONTROL LOCATION'/)
        STOP
  150   CONTINUE
C--The difference between land surface and unmanaged  head at each
C----well location gives the unmanaged  lift.  The product of this
C----lift and discounted cost per unit lift per unit pumpage gives
C----the cost coefficient for the linear part of the quadratic
C----objective function:
  160 KIW=IW-KK
      COSTC(KIW,N)=COSTC(KIW,N)*(SURF(KIW)-HDUS(L))
  170 CONTINUE
      RETURN
```

```
      END
C-----------------------------------------------------------------------
      SUBROUTINE READ1(NENT,KP,KPM1,NWEL,TMAX,NUMT,DELT,CDLT,I,J,II,
     &RADIUS,XWELL)
      IMPLICIT REAL *8 (A-H,O-Z)
      COMMON /LOCATS/ ILOCW(200),JLOCW(200),ILOCC(500),JLOCC(500),
     & KEYQ(200),KEYWL(500),XRAD(200)
      COMMON /STRESS/ QWELL(2000),XHEAD(2500)
      COMMON /PARAMS/ KCALLP,NWLS,NCNTR,NNPER,KPER,CDELT,TIMPER,
     & TIMINC(50)
      GO TO (100,200) NENT
C--Retrieve pumping period info from "AQMAN" Common:
  100 KP=KPER
      KPM1=KPER-1
      NWEL=NWLS
      TMAX=TIMPER
      DELT=TIMINC(KPER)
      CDLT=CDELT
C--Set limit on time step number very high, so that TIMPER (length of
C----period in days) will be used by TRES.
      NUMT=32000
      RETURN
  200 CONTINUE
C--Retrieve well locations, radii, and pumpages from "AQMAN" Common:
      I=ILOCW(II)
      J=JLOCW(II)
      RADIUS=XRAD(II)
      LPER=(KPER-1)*NWLS
      KWEL=LPER+II
      XWELL=QWELL(KWEL)
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE WRITE1(NENT,IZ,JZ,KW,HW,PHI)
      IMPLICIT REAL *8 (A-H,O-Z)
      DIMENSION PHI(IZ,JZ)
      COMMON /LOCATS/ ILOCW(200),JLOCW(200),ILOCC(500),JLOCC(500),
     & KEYQ(200),KEYWL(500),XRAD(200)
      COMMON /STRESS/ QWELL(2000),XHEAD(2500)
      COMMON /PARAMS/ KCALLP,NWLS,NCNTR,NNPER,KPER,CDELT,TIMPER,
     & TIMINC(50)
      GO TO (101,102) NENT
C--Convert PHI matrix computed by TRESCOTT (at end of pumping period
C----or if steady-state is reached) to XHEAD vector to store in
C----"AQMAN1" COMMON:
  101 KCNTR=NCNTR*(KPER-2)
      LBEG=KCNTR+1
      LEND=NCNTR*(KPER-1)
      LL=0
      DO 100 L=LBEG,LEND
      LL=LL+1
      DO 110 I=1,IZ
      IF(I.NE.ILOCC(LL))GO TO 110
      II=I
```

```
  110 CONTINUE                                                          WRI 220
      DO 120 J=1,JZ                                                     WRI 230
      IF(J.NE.JLOCC(LL))GO TO 120                                       WRI 240
      JJ=J                                                              WRI 250
  120 CONTINUE                                                          WRI 260
      XHEAD(L)=PHI(II,JJ)                                               WRI 270
  100 CONTINUE                                                          WRI 280
      RETURN                                                            WRI 290
  102 CONTINUE                                                          WRI 300
C--If KEYWL is not zero, write head at well radius into XHEAD vector:   WRI 310
      KKW=KW                                                            WRI 320
      LBEG=NCNTR*(KPER-2)+1                                             WRI 330
      LEND=NCNTR*(KPER-1)                                               WRI 340
      LL=0                                                              WRI 350
      DO 10 L=LBEG,LEND                                                 WRI 360
      LL=LL+1                                                           WRI 370
      IF(KEYWL(LL).EQ.0)GO TO 10                                        WRI 380
      KKW=KKW-1                                                         WRI 390
      IF(KKW.NE.0)GO TO 10                                              WRI 400
      XHEAD(L)=HW                                                       WRI 410
      RETURN                                                            WRI 420
   10 CONTINUE                                                          WRI 430
      RETURN                                                            WRI 440
      END                                                               WRI 450
C-----------------------------------------------------------------------
      SUBROUTINE TRES(KCALL)
C                          FINITE-DIFFERENCE MODEL                      MAN  20
C                                    FOR                                MAN  30
C                     SIMULATION OF GROUND-WATER FLOW                   MAN  40
C                            IN TWO DIMENSIONS                          MAN  50
C                                                                       MAN  60
C             BY P. C. TRESCOTT, G. F. PINDER AND S. P. LARSON          MAN  70
C                          U. S. GEOLOGICAL SURVEY                      MAN  80
C                              SEPTEMBER, 1975                          MAN  90
C     *****************************************************************MAN 100
C     MAIN PROGRAM TO DIMENSION DIGITAL MODEL AND CONTROL SEQUENCE      MAN 110
C     OF COMPUTATIONS                                                   MAN 120
C     -----------------------------------------------------------------MAN 130
C---All lines inserted due to adaptation for optimization linkage are   $1-10
C----marked in columns 73-80 with a $. If $$ appears, the line is from  $1-20
C----the original TRESCOTT code, but has been altered. Lines containing $1-30
C----WRITEs which are commented out are not marked. The first digit     $1-40
C----after $ or $$ denotes the TRESCOTT subroutine: 1=Main,2=DATAI,etc. $1-50
C----The remaining digits run sequentially within a subroutine.         $1-60
C----If no line number appears, the line was inserted or altered by the $1-61
C----the New Mexico district when converting the original TRESCOTT code $1-62
C----into FORTRAN-77.                                                   $1-63
C     SPECIFICATIONS:                                                   MAN 140
      IMPLICIT REAL *8 (A-H,O-Z)                                        $1-70
      REAL *8 KEEP,M                                                    $$1-71
      REAL *4 HEADNG(32)
      INTEGER *4 DIML,DIMW,CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,
     1CONTR,LEAK,RECH,SIP,ADI
      INTEGER *4 IH,IZ,JZ,NW,ITMAX,L,ISUM,ISIZ,IP,JP,IR,JR,IC,JC,
```

```
     &IL,JL,IS,JS,IMAX,IMX1
      INTEGER R,P,PU
C
      DIMENSION Y(70000),L(37),IFMT1(9),IFMT2(9),IFMT3(9),NAME(99),
     1YY(1),IFMT(9),IN(9)
      EQUIVALENCE (YY(1),Y(1))
      COMMON/ARR/Y
      INTEGER *4 NUMS,IDK1,IDK2,IRN,IN
      COMMON/NAMES/IFMT,IN,IRN
C                                                                       MAN 190
C                                                                       MAN 230
      COMMON /SARRAY/ VF4(11),CHK(15)                                   MAN 240
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1
      COMMON /SPARAMI/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,LEAK,
     1RECH,SIP,NUMS,LSOR,ADI,IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,
     2ITMAX,LENGTH,NWEL,NW,DIML,DIMW,I,J,IDK1,IDK2,JNO1,INO1,R,P,PU
      COMMON /SPARAMR/ U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,
     1WIDTH,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,EROR
      COMMON /LOCATS/ ILOCW(200),JLOCW(200),ILOCC(500),JLOCC(500),      $1-80
     & KEYQ(200),KEYWL(500),XRAD(200)                                   $1-90
      COMMON /PARAMS/ KCALLP,NWLS,NCNTR,NNPER,KPER,CDELT,TIMPER,        $1-95
     & TIMINC(50)                                                       $1-100
C                                                                       MAN 310
      DATA IFMT1/4H(1H0,4H,I5,,4H10E1,4H1.3/,4H(1H ,4H,5X,,4H10E1,4H1.3)MAN 320
     1,4H)   /                                                          MAN 330
      DATA IFMT2/4H('0',4H,I2,,4H2X,2,4H0F6.,4H1/(5,4HX,20,4HF6.1,4H))  MAN 340
     1,4H    /                                                          MAN 350
      DATA IFMT3/4H(1H0,4H,I5,,4H14F9,4H.5/(,4H1H ,,4H5X,1,4H4F9.,4H5)) MAN 360
     1,4H    /                                                          MAN 370
      DATA NAME/2*4H    ,4H STO,4HRAGE,4H COE,4HFFIC,4HIENT,4*4H    ,4H MAN 380
     1  T,4HRANS,4HMISS,4HIVIT,4HY   ,2*4H    ,4H   A,4HQUIF,4HER H,4HYDMAN 390
     2RA,4HULIC,4H CON,4HDUCT,4HIVIT,4HY   ,4H    ,4H   A,4HQUIF,4HER B,MAN 400
     34HASE ,4HELEV,4HATIO,4HN   ,3*4H    ,4H   S,4HPECI,4HFIC ,4HYIEL,4MAN 410
     4HD   ,4*4H    ,4HAQUI,4HFER ,4HTOP ,4HELEV,4HATIO,4HN   ,4H    ,4HMAN 420
     5CONF,4HININ,4HG BE,4HD HY,4HDRAU,4HLIC ,4HCOND,4HUCTI,4HVITY,3*4H MAN 430
     6    ,4H RIV,4HER H,4HEAD ,4*4H    ,4H   C,4HONFI,4HNING,4H BED,4H TMAN 440
     7HI,4HCKNE,4HSS  ,2*4H    ,4H   L,4HAND ,4HSURF,4HACE ,4HELEV,4HATIMAN 450
     8O,4HN   ,3*4H    ,4H ARE,4HAL R,4HECHA,4HRGE ,4HRATE,2*4H    /    MAN 460
C                                                                       MAN 470
C     OPEN(UNIT=5,STATUS='OLD',ACCESS='DIRECT',RECL=2624)
C     OPEN(UNIT=6,STATUS='OLD',ACCESS='DIRECT',RECL=2624)
C     ..................................................................MAN 490
C--If not first call of TRES, skip to 900:                              $1-110
      KCALLP=KCALL                                                      $1-120
      IF (KCALL.NE.1)GO TO 900                                          $1-130
C                                                                       MAN 500
C     ---READ TITLE,PROGRAM OPTIONS AND PROGRAM SIZE---                 MAN 510
   10 READ (R,370) HEADNG                                               MAN 520
C     WRITE (P,360) HEADNG                                              MAN 530
      READ (R,380) WATER,LEAK,CONVRT,EVAP,RECH,NUMS,CHCK,PNCH,IDK1,IDK2,MAN 540
     1NUM,HEAD                                                          MAN 550
C     WRITE (P,390) WATER,LEAK,CONVRT,EVAP,RECH,NUMS,CHCK,PNCH,IDK1,IDK2MAN 560
C    1,NUM,HEAD                                                         MAN 570
      IF (NUMS.EQ.CHK(11).OR.NUMS.EQ.CHK(12).OR.NUMS.EQ.CHK(13)) GO TO 2MAN 580
```

```
   10                                                                  MAN 590
      WRITE (P,350)                                                    MAN 600
      STOP                                                             MAN 610
   20 READ (R,320) DIML,DIMW,NW,ITMAX                                  MAN 620
C     WRITE (P,340) DIML,DIMW,NW,ITMAX                                 MAN 630
C--Check input data for consistency and linearity                      $1-131
      CALL CHKDAT(NW,WATER,CONVRT,EVAP,LEAK)                           $1-132
C                                                                      MAN 640
C     ---COMPUTE DIMENSIONS FOR ARRAYS---                              MAN 650
      IZ=DIML                                                          MAN 660
      JZ=DIMW                                                          MAN 670
      IH=MAX0(1,NW)                                                    MAN 680
      IMAX=MAX0(DIML,DIMW)                                             MAN 690
      ISIZ=DIML*DIMW                                                   MAN 700
      ISUM=2*ISIZ+1                                                    MAN 710
      IMX1=ITMAX+1                                                     MAN 720
      L(1)=1                                                           MAN 730
      DO 30 I=2,4                                                      MAN 740
      L(I)=ISUM                                                        MAN 750
   30 ISUM=ISUM+2*IMAX                                                 MAN 760
      DO 40 I=5,16                                                     MAN 770
      L(I)=ISUM                                                        MAN 780
   40 ISUM=ISUM+ISIZ                                                   MAN 790
      IF (WATER.NE.CHK(2)) GO TO 60                                    MAN 800
      DO 50 I=17,19                                                    MAN 810
      L(I)=ISUM                                                        MAN 820
   50 ISUM=ISUM+ISIZ                                                   MAN 830
      IP=DIML                                                          MAN 840
      JP=DIMW                                                          MAN 850
      GO TO 80                                                         MAN 860
   60 DO 70 I=17,19                                                    MAN 870
      L(I)=ISUM                                                        MAN 880
   70 ISUM=ISUM+1                                                      MAN 890
      IP=1                                                             MAN 900
      JP=1                                                             MAN 910
   80 IF (LEAK.NE.CHK(9)) GO TO 100                                    MAN 920
      DO 90 I=20,22                                                    MAN 930
      L(I)=ISUM                                                        MAN 940
   90 ISUM=ISUM+ISIZ                                                   MAN 950
      IR=DIML                                                          MAN 960
      JR=DIMW                                                          MAN 970
      GO TO 120                                                        MAN 980
  100 DO 110 I=20,22                                                   MAN 990
      L(I)=ISUM                                                        MAN1000
  110 ISUM=ISUM+1                                                      MAN1010
      IR=1                                                             MAN1020
      JR=1                                                             MAN1030
  120 IF (CONVRT.NE.CHK(7)) GO TO 130                                  MAN1040
      L(23)=ISUM                                                       MAN1050
      ISUM=ISUM+ISIZ                                                   MAN1060
      IC=DIML                                                          MAN1070
      JC=DIMW                                                          MAN1080
      GO TO 140                                                        MAN1090
  130 L(23)=ISUM                                                       MAN1100
```

```
      ISUM=ISUM+1                                                       MAN1110
      IC=1                                                              MAN1120
      JC=1                                                              MAN1130
  140 IF (EVAP.NE.CHK(6)) GO TO 150                                     MAN1140
      L(24)=ISUM                                                        MAN1150
      ISUM=ISUM+ISIZ                                                    MAN1160
      IL=DIML                                                           MAN1170
      JL=DIMW                                                           MAN1180
      GO TO 160                                                         MAN1190
  150 L(24)=ISUM                                                        MAN1200
      ISUM=ISUM+1                                                       MAN1210
      IL=1                                                              MAN1220
      JL=1                                                              MAN1230
  160 IF (NUMS.NE.CHK(11)) GO TO 180                                    MAN1240
      DO 170 I=25,28                                                    MAN1250
      L(I)=ISUM                                                         MAN1260
  170 ISUM=ISUM+ISIZ                                                    MAN1270
      IS=DIML                                                           MAN1280
      JS=DIMW                                                           MAN1290
      GO TO 200                                                         MAN1300
  180 DO 190 I=25,28                                                    MAN1310
      L(I)=ISUM                                                         MAN1320
  190 ISUM=ISUM+1                                                       MAN1330
      IS=1                                                              MAN1340
      JS=1                                                              MAN1350
  200 DO 210 I=29,31                                                    MAN1360
      L(I)=ISUM                                                         MAN1370
  210 ISUM=ISUM+DIMW                                                    MAN1380
      DO 220 I=32,33                                                    MAN1390
      L(I)=ISUM                                                         MAN1400
  220 ISUM=ISUM+DIML                                                    MAN1410
      L(34)=ISUM                                                        MAN1420
      ISUM=ISUM+IH                                                      MAN1430
      L(35)=ISUM                                                        MAN1440
      ISUM=ISUM+2*IH                                                    MAN1450
      IF (MOD(ISUM,2).EQ.0) ISUM=ISUM+1                                 MAN1460
  230 L(36)=ISUM                                                        MAN1480
      ISUM=ISUM+2*IMAX                                                  MAN1490
      L(37)=ISUM                                                        MAN1500
      ISUM=ISUM+IMX1                                                    MAN1510
C     WRITE (P,330) ISUM                                                MAN1520
C                                                                       MAN1530
C     ---PASS INTIIAL ADDRESSES OF ARRAYS TO SUBROUTINES---             MAN1540
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1MAN1550
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))MAN1560
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(MAN1570
     3L(34)),Y(L(35)),Y(L(13)),10)                                      MAN1580
      CALL STEP(Y(L(1)),Y(L(5)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(14)),Y(L(17)MAN1590
     1),Y(L(18)),Y(L(23)),Y(L(29)),Y(L(30)),Y(L(32)),Y(L(34)),Y(L(35)),YMAN1600
     2(L(37)),Y(L(10)),Y(L(11)),Y(L(12)),Y(L(13)),Y(L(15)),Y(L(19)),Y(L(MAN1602
     320)),Y(L(21)),Y(L(22)),Y(L(24)),10)                               MAN1604
      IF (NUMS.EQ.CHK(11)) CALL SOLVE1(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),YMAN1620
     1(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15))MAN1630
     2,Y(L(16)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(31)),Y(MAN1640
```

```
     3L(32)),Y(L(33)),Y(L(37)),Y(L(10)),Y(L(11)),Y(L(24)),Y(L(19)),Y(L(2MAN1650
     43)),Y(L(20)),Y(L(22)),Y(L(21)),Y(L(8)),Y(L(17)),Y(L(18)),Y(L(30)),MAN1652
     5Y(L(34)),Y(L(35)),10)                                             MAN1654
      IF (NUMS.EQ.CHK(12)) CALL SOLVE2(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),YMAN1670
     1(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15))MAN1680
     2,Y(L(16)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(31)),Y(MAN1690
     3L(32)),Y(L(33)),Y(L(37)),Y(L(10)),Y(L(11)),Y(L(24)),Y(L(19)),Y(L(2MAN1700
     43)),Y(L(20)),Y(L(22)),Y(L(21)),Y(L(8)),Y(L(17)),Y(L(18)),Y(L(30)),MAN1702
     5Y(L(34)),Y(L(35)),10)                                             MAN1704
      IF (NUMS.EQ.CHK(13)) CALL SOLVE3(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),YMAN1720
     1(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15))MAN1730
     2,Y(L(16)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(31)),Y(MAN1740
     3L(32)),Y(L(33)),Y(L(36)),Y(L(37)),Y(L(10)),Y(L(11)),Y(L(24)),Y(L(1MAN1750
     49)),Y(L(23)),Y(L(20)),Y(L(22)),Y(L(21)),Y(L(8)),Y(L(17)),Y(L(18)),MAN1752
     5Y(L(30)),Y(L(34)),Y(L(35)),10)                                    MAN1754
      CALL COEF(Y(L(1)),Y(L(5)),Y(L(6)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10))MAN1770
     1,Y(L(11)),Y(L(12)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(MAN1780
     2L(19)),Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(3MAN1790
     32)),Y(L(30)),Y(L(34)),Y(L(35)),10)                                MAN1800
      CALL CHECKI(Y(L(1)),Y(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(10)),Y(L(1MAN1810
     11)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(17)),Y(L(18)),Y(L(19))MAN1820
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),10MAN1830
     3)
      CALL PRNTAI(Y(L(1)),Y(L(8)),Y(L(9)),Y(L(12)),Y(L(14)),Y(L(29)),Y(LMAN1840
     1(32)),1,10)                                                       MAN1850
C     ..................................................................MAN1860
C                                                                       MAN1870
C     ---START COMPUTATIONS---                                          MAN1880
C     ************************                                          MAN1890
C     ---READ AND WRITE DATA FOR GROUPS II AND III---                   MAN1900
  900 CONTINUE                                                          $1-140
C--If not first call of TRES, enter DATAI only to set PHI=SURI:         $1-150
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
     3L(34)),Y(L(35)),Y(L(13)),1)
C--If not first call of TRES, do not read data:                         $1-160
      IF(KCALL.NE.1)GO TO 910                                           $1-170
      DO 1 I=1,9
      IFMT(I)=IFMT3(I)
1     IN(I)=NAME(I)
      IRN=2
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
     3L(34)),Y(L(35)),Y(L(12)),2)
      IF (WATER.EQ.CHK(2)) GO TO 240                                    MAN1930
      DO 2 I=1,9
      IFMT(I)=IFMT3(I)
2     IN(I)=NAME(9+I)
      IRN=3
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
```

```
     3L(34)),Y(L(35)),Y(L(9)),2)
      GO TO 250                                                         MAN1950
240   DO 3 I=1,9
      IFMT(I)=IFMT1(I)
3     IN(I)=NAME(18+I)
      INR=4
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
     3L(34)),Y(L(35)),Y(L(17)),2)
      DO 4 I=1,9
      IFMT(I)=IFMT2(I)
4     IN(I)=NAME(27+I)
      INR=5
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
     3L(34)),Y(L(35)),Y(L(18)),2)
      DO 5 I=1,9
      IFMT(I)=IFMT3(I)
5     IN(I)=NAME(36+I)
      INR=6
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
     3L(34)),Y(L(35)),Y(L(19)),2)
  250 IF(CONVRT.EQ.CHK(7)) THEN
      DO 6 I=1,9
      IFMT(I)=IFMT2(I)
6     IN(I)=NAME(45+I)
      INR=7
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
     3L(34)),Y(L(35)),Y(L(23)),2)
      ENDIF
      IF (LEAK.NE.CHK(9)) GO TO 260                                     MAN2000
      DO 7 I=1,9
      IFMT(I)=IFMT1(I)
7     IN(I)=NAME(54+I)
      INR=8
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
     3L(34)),Y(L(35)),Y(L(20)),2)
      DO 8 I=1,9
      IFMT(I)=IFMT2(I)
8     IN(I)=NAME(63+I)
      INR=9
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
     3L(34)),Y(L(35)),Y(L(21)),2)
      DO 9 I=1,9
```

```
      IFMT(I)=IFMT2(I)
9     IN(I)=NAME(72+I)
      INR=10
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
     3L(34)),Y(L(35)),Y(L(22)),2)
260   IF (EVAP.EQ.CHK(6)) THEN
      DO 11 I=1,9
      IFMT(I)=IFMT2(I)
11    IN(I)=NAME(81+I)
      INR=11
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
     3L(34)),Y(L(35)),Y(L(24)),2)
      ENDIF
      IF(RECH.EQ.CHK(10)) THEN
      DO 12 I=1,9
      IFMT(I)=IFMT1(I)
12    IN(I)=NAME(90+I)
      INR=12
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
     3L(34)),Y(L(35)),Y(L(13)),2)
      ENDIF
      CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
     3L(34)),Y(L(35)),Y(L(13)),3)
C                                                                       MAN2070
C     ---INITIALIZE TRANSMISSIVITY VALUES IN WATER TABLE PROBLEM---     MAN2080
C--If not first call of TRES, enter COEF to recompute                   $1-180
C------transmissivities, based on heads (PHI) for unconfined aquifer:   $1-190
  910 CONTINUE                                                          $1-200
      KT=0                                                              MAN2090
      IF (WATER.EQ.CHK(2)) THEN
      CALL COEF(Y(L(1)),Y(L(5)),Y(L(6)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10))
     1,Y(L(11)),Y(L(12)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(
     2L(19)),Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(3
     32)),Y(L(30)),Y(L(34)),Y(L(35)),2)
      ENDIF
C                                                                       MAN2110
C--If not first call of TRES, begin new pumping period.                 $1-210
      IF(KCALL.NE.1) GO TO 270                                          $1-220
C     ---COMPUTE ITERATION PARAMETERS---                                MAN2120
      IF(NUMS.EQ.CHK(11)) THEN
      CALL SOLVE1(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y
     1(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15))
     2,Y(L(16)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(31)),Y(
     3L(32)),Y(L(33)),Y(L(37)),Y(L(10)),Y(L(11)),Y(L(24)),Y(L(19)),Y(L(2
     43)),Y(L(20)),Y(L(22)),Y(L(21)),Y(L(8)),Y(L(17)),Y(L(18)),Y(L(30)),
     5Y(L(34)),Y(L(35)),1)
```

```
      ENDIF
      IF(NUMS.EQ.CHK(12)) THEN
      CALL SOLVE2(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y
     1(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15))
     2,Y(L(16)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(31)),Y(
     3L(32)),Y(L(33)),Y(L(37)),Y(L(10)),Y(L(11)),Y(L(24)),Y(L(19)),Y(L(2
     43)),Y(L(20)),Y(L(22)),Y(L(21)),Y(L(8)),Y(L(17)),Y(L(18)),Y(L(30)),
     5Y(L(34)),Y(L(35)),1)
      ENDIF
      IF(NUMS.EQ.CHK(13)) THEN
      CALL SOLVE3(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y
     1(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15))
     2,Y(L(16)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(31)),Y(
     3L(32)),Y(L(33)),Y(L(36)),Y(L(37)),Y(L(10)),Y(L(11)),Y(L(24)),Y(L(1
     49)),Y(L(23)),Y(L(20)),Y(L(22)),Y(L(21)),Y(L(8)),Y(L(17)),Y(L(18)),
     5Y(L(30)),Y(L(34)),Y(L(35)),1)
      ENDIF
C                                                                       MAN2160
C     ---INITIALIZE PARAMETERS FOR ALPHAMERIC MAP---                    MAN2170
      IF (CONTR.EQ.CHK(3)) THEN
      CALL PRNTAI(Y(L(1)),Y(L(8)),Y(L(9)),Y(L(12)),Y(L(14)),Y(L(29)),Y(L
     1(32)),1,1)
      ENDIF
C                                                                       MAN2190
C     ---COMPUTE T COEFFICIENTS FOR ARTESIAN PROBLEM---                 MAN2200
      IF(WATER.NE.CHK(2)) THEN
      CALL COEF(Y(L(1)),Y(L(5)),Y(L(6)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10))
     1,Y(L(11)),Y(L(12)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(
     2L(19)),Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(3
     32)),Y(L(30)),Y(L(34)),Y(L(35)),3)
      ENDIF
C                                                                       MAN2220
C     ---READ TIME PARAMETERS AND PUMPING DATA FOR A NEW PUMPING PERIOD-MAN2230
270   CALL DATAI(Y(L(1)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10)),Y(L(11)),Y(L(1
     12)),Y(L(13)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(L(19))
     2,Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(32)),Y(
     3L(34)),Y(L(35)),Y(L(13)),4)
C                                                                       MAN2250
      KT=0                                                              MAN2260
      IFINAL=0                                                          MAN2270
      IERR=0                                                            MAN2280
C                                                                       MAN2290
C     ---START NEW TIME STEP COMPUTATIONS---                            MAN2300
280   CALL STEP(Y(L(1)),Y(L(5)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(14)),Y(L(17)
     1),Y(L(18)),Y(L(23)),Y(L(29)),Y(L(30)),Y(L(32)),Y(L(34)),Y(L(35)),Y
     2(L(37)),Y(L(10)),Y(L(11)),Y(L(12)),Y(L(13)),Y(L(15)),Y(L(19)),Y(L(
     320)),Y(L(21)),Y(L(22)),Y(L(24)),1)
C                                                                       MAN2320
C     ---COMPUTE TRANSIENT PART OF LEAKAGE TERM---                      MAN2330
      IF (LEAK.EQ.CHK(9).AND.SS.NE.0) THEN
      CALL COEF(Y(L(1)),Y(L(5)),Y(L(6)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(10))
     1,Y(L(11)),Y(L(12)),Y(L(14)),Y(L(15)),Y(L(16)),Y(L(17)),Y(L(18)),Y(
     2L(19)),Y(L(20)),Y(L(21)),Y(L(22)),Y(L(23)),Y(L(24)),Y(L(29)),Y(L(3
     32)),Y(L(30)),Y(L(34)),Y(L(35)),1)
```

```
      ENDIF
C                                                                       MAN2350
C     ---ENTER APPROPIATE SOLUTION ROUTINE AND COMPUTE SOLUTION---      MAN2360
      IF (NUMS.EQ.CHK(11)) THEN
      CALL SOLVE1(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y
     1(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15))
     2,Y(L(16)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(31)),Y(
     3L(32)),Y(L(33)),Y(L(37)),Y(L(10)),Y(L(11)),Y(L(24)),Y(L(19)),Y(L(2
     43)),Y(L(20)),Y(L(22)),Y(L(21)),Y(L(8)),Y(L(17)),Y(L(18)),Y(L(30)),
     5Y(L(34)),Y(L(35)),2)
      ENDIF
      IF(NUMS.EQ.CHK(12)) THEN
      CALL SOLVE2(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y
     1(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15))
     2,Y(L(16)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(31)),Y(
     3L(32)),Y(L(33)),Y(L(37)),Y(L(10)),Y(L(11)),Y(L(24)),Y(L(19)),Y(L(2
     43)),Y(L(20)),Y(L(22)),Y(L(21)),Y(L(8)),Y(L(17)),Y(L(18)),Y(L(30)),
     5Y(L(34)),Y(L(35)),2)
      ENDIF
      IF (NUMS.EQ.CHK(13)) THEN
      CALL SOLVE3(Y(L(1)),Y(L(2)),Y(L(3)),Y(L(4)),Y
     1(L(5)),Y(L(6)),Y(L(7)),Y(L(9)),Y(L(12)),Y(L(13)),Y(L(14)),Y(L(15))
     2,Y(L(16)),Y(L(25)),Y(L(26)),Y(L(27)),Y(L(28)),Y(L(29)),Y(L(31)),Y(
     3L(32)),Y(L(33)),Y(L(36)),Y(L(37)),Y(L(10)),Y(L(11)),Y(L(24)),Y(L(1
     49)),Y(L(23)),Y(L(20)),Y(L(22)),Y(L(21)),Y(L(8)),Y(L(17)),Y(L(18)),
     5Y(L(30)),Y(L(34)),Y(L(35)),2)
      ENDIF
C                                                                       MAN2400
C     ---CHECK FOR STEADY STATE AND PRINT OUTPUT AT DESIGNATED          MAN2410
C     TIME STEPS---                                                     MAN2420
      CALL STEP(Y(L(1)),Y(L(5)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(14)),Y(L(17)
     1),Y(L(18)),Y(L(23)),Y(L(29)),Y(L(30)),Y(L(32)),Y(L(34)),Y(L(35)),Y
     2(L(37)),Y(L(10)),Y(L(11)),Y(L(12)),Y(L(13)),Y(L(15)),Y(L(19)),Y(L(
     320)),Y(L(21)),Y(L(22)),Y(L(24)),2)
C                                                                       MAN2440
C     ---LAST TIME STEP IN PUMPING PERIOD ?---                          MAN2450
      IF (IFINAL.NE.1) GO TO 280                                        MAN2460
C                                                                       MAN2470
C     ---CHECK FOR NEW PUMPING PERIOD---                                MAN2480
      IF (KP.LT.NPER) GO TO 270                                         MAN2490
C                                                                       MAN2500
C     ---DISK OUTPUT IF DESIRED---                                      MAN2510
      IF (IDK2.NE.CHK(15)) GO TO 290                                    MAN2520
      CALL STEP(Y(L(1)),Y(L(5)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(14)),Y(L(17)
     1),Y(L(18)),Y(L(23)),Y(L(29)),Y(L(30)),Y(L(32)),Y(L(34)),Y(L(35)),Y
     2(L(37)),Y(L(10)),Y(L(11)),Y(L(12)),Y(L(13)),Y(L(15)),Y(L(19)),Y(L(
     320)),Y(L(21)),Y(L(22)),Y(L(24)),5)
C                                                                       MAN2540
C     ---PUNCHED OUTPUT IF DESIRED---                                   MAN2550
  290 IF (PNCH.NE.CHK(1)) GO TO 300                                     MAN2560
      CALL STEP(Y(L(1)),Y(L(5)),Y(L(7)),Y(L(8)),Y(L(9)),Y(L(14)),Y(L(17)
     1),Y(L(18)),Y(L(23)),Y(L(29)),Y(L(30)),Y(L(32)),Y(L(34)),Y(L(35)),Y
     2(L(37)),Y(L(10)),Y(L(11)),Y(L(12)),Y(L(13)),Y(L(15)),Y(L(19)),Y(L(
     320)),Y(L(21)),Y(L(22)),Y(L(24)),6)
```

```
C                                                                       MAN2580
C--Return to PRE, rather than stopping:                                 $1-250
  300 RETURN                                                            $1-251
C     ---CHECK FOR NEW PROBLEM---                                       MAN2590
C 300 READ (R,320,END=310) NEXT                                         $$1-252
C     IF (NEXT.EQ.0) GO TO 10                                           $$1-253
C 310 STOP                                                              $$1-254
C     ..................................................................MAN2630
C                                                                       MAN2640
C     ---FORMATS---                                                     MAN2650
C     ------------------------------------------------------------------MAN2660
C                                                                       MAN2670
C                                                                       MAN2680
  320 FORMAT (4I10)                                                     MAN2690
  330 FORMAT ('0',54X,'WORDS OF Y VECTOR USED =',I7)                    MAN2700
  340 FORMAT ('0',62X,'NUMBER OF ROWS =',I5/60X,'NUMBER OF COLUMNS =',I5MAN2710
     1/9X,'NUMBER OF WELLS FOR WHICH DRAWDOWN IS COMPUTED AT A SPECIFIEDMAN2720
     2 RADIUS =',I5,/,39X,'MAXIMUM PERMITTED NUMBER OF ITERATIONS =',I5)MAN2730
  350 FORMAT ('-',36X,'NO EQUATION SOLVING SCHEME SPECIFIED, EXECUTION TMAN2740
     1ERMINATED'/37X,58('*'))                                           MAN2750
  360 FORMAT ('1',60X,'U. S. G. S.'//55X,'FINITE-DIFFERENCE MODEL'/65X,'MAN2760
     1FOR'/51X,'SIMULATION OF GROUND-WATER FLOW'//60X,'JANUARY, 1975'//1MAN2770
     233('*')/'0',32A4//133('*'))                                       MAN2780
  370 FORMAT (20A4)                                                     MAN2790
  380 FORMAT (16(A4,1X))                                                MAN2800
  390 FORMAT ('-SIMULATION OPTIONS:  ',13(A4,4X))                       MAN2810
      END                                                               MAN2820-
      SUBROUTINE DATAI(PHI,STRT,SURI,T,TR,TC,S,QRE,WELL,TL,SL,PERM,BOTTODAT  10
     1M,SY,RATE,RIVER,M,TOP,GRND,DELX,DELY,WR,NWR,A,NENT)
C     ------------------------------------------------------------------DAT  30
C     READ AND WRITE INPUT DATA                                         DAT  40
C     ------------------------------------------------------------------DAT  50
C                                                                       DAT  60
C     SPECIFICATIONS:                                                   DAT  70
      IMPLICIT REAL *8 (A-H,O-Z)                                        $2-10
      REAL *8 M,MESUR                                                   $$2-11
      INTEGER R,P,PU,DIML,DIMW
      INTEGER CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD
     1,CONTR,LEAK,RECH,SIP,ADI,NUMS,IDK1,IDK2
C                                                                       DAT 120
      DIMENSION PHI(IZ,JZ), STRT(IZ,JZ), SURI(IZ,JZ), T(IZ,JZ), TR(IZ,JZDAT 130
     1), TC(IZ,JZ), S(IZ,JZ), QRE(IZ,JZ), WELL(IZ,JZ), TL(IZ,JZ), SL(IZ,DAT 140
     2JZ), PERM(IP,JP), BOTTOM(IP,JP), SY(IP,JP), RATE(IR,JR), RIVER(IR,DAT 150
     3JR), M(IR,JR), TOP(IC,JC), GRND(IL,JL), DELX(JZ), DELY(IZ), WR(IH)DAT 160
     4, NWR(IH,2), A(IZ,JZ), IN(9), IFMT(9)                             DAT 170
C                                                                       DAT 180
      COMMON /SARRAY/ VF4(11),CHK(15)                                   DAT 190
      COMMON /SPARAMI/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,LEAK,
     1RECH,SIP,NUMS,LSOR,ADI,IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,
     2ITMAX,LENGTH,NWEL,NW,DIML,DIMW,I,J,IDK1,IDK2,JNO1,INO1,R,P,PU
      COMMON /SPARAMR/ U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,
     1WIDTH,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,EROR
      COMMON /CK/ ETFLXT,STORT,QRET,CHST,CHDT,FLUXT,PUMPT,CFLUXT,FLXNT  DAT 250
      COMMON /PRI/ NA(4),N1,N2,N3
```

```
      COMMON /PRR/ PRNT(122),DIGIT(122),XN(100),BLANK(60),SYM(17),
     1YN(13),VF1(6),VF2(6),VF3(7),YLABEL(6),TITLE(5),XLABEL(3),XN1,
     2MESUR,XSCALE,YSCALE,DINCH,FACT1,FACT2
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1
      COMMON /NAMES/IFMT,IN,IRN
      COMMON /LOCATS/ ILOCW(200),JLOCW(200),ILOCC(500),JLOCC(500),      $2-20
     & KEYQ(200),KEYWL(500),XRAD(200)                                   $2-30
      COMMON /PARAMS/ KCALLP,NWLS,NCNTR,NNPER,KPER,CDELT,TIMPER,        $2-40
     & TIMINC(50)                                                       $2-50
      GO TO(1,2,3,4) NENT
      RETURN                                                            DAT 300
C     ......................................................................
C     *********************
C     ENTRY DATAIN(PHI,STRT,SURI,T,TR,TC,QRE,WELL,TL,SL
C    1,SY,RATE,RIVER,M,TOP,GRND,DELX,DELY,WR,NWR)
C     *********************
1     CONTINUE
      IF(KCALLP.NE.1)GO TO 970                                          $2-60
C                                                                       DAT 350
C     ---READ AND WRITE SCALAR PARAMETERS---                            DAT 360
      READ (R,500) CONTR,XSCALE,YSCALE,DINCH,FACT1,FACT2,MESUR          DAT 370
C     IF (CONTR.EQ.CHK(3)) WRITE (P,610) XSCALE,YSCALE,MESUR,MESUR,DINCHDAT 380
C    1,FACT1,FACT2                                                      DAT 390
      READ (R,490) NPER,KTH,ERR,EROR,SS,QET,ETDIST,LENGTH               DAT 400
C--Overwrite the value of NNPER from AQMAN common onto NPER:            $2-61
      NPER=NNPER                                                        $2-62
      READ(R,485) HMAX,FACTX,FACTY
      IF (ETDIST.LE.0.) ETDIST=1.                                       DAT 420
C     WRITE (P,520) NPER,KTH,ERR,EROR,SS,QET,ETDIST,FACTX,FACTY         DAT 430
C                                                                       DAT 440
C     ---READ CUMULATIVE MASS BALANCE PARAMETERS---                     DAT 450
      READ (R,600) SUM,SUMP,PUMPT,CFLUXT                                DAT 460
      READ (R,600) QRET,CHST,CHDT,FLUXT
      READ (R,605) STORT,ETFLXT,FLXNT
      IF (IDK1.EQ.CHK(14)) GO TO 20                                     DAT 480
      IF (SUM.EQ.0.0) GO TO 40
C     WRITE (P,480) SUM                                                 DAT 500
C     ..................................................................DAT 510
C                                                                       DAT 520
C     ---HEAD DATA TO CONTINUE PREVIOUS COMPUTATIONS READ HERE---       DAT 530
C     ------FROM CARDS:                                                 DAT 540
      DO 10 I=1,DIML                                                    DAT 550
      READ (R,540) (PHI(I,J),J=1,DIMW)                                  DAT 560
C  10 WRITE (P,530) I,(PHI(I,J),J=1,DIMW)                               DAT 570
   10 CONTINUE
      GO TO 40                                                          DAT 580
C     ------READ AND WRITE DATA FROM UNIT 6 ON DISK RATHER THAN CARDS:  DAT 590
   20 READ (6) PHI,SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFLDAT 600
     1XT,FLXNT                                                          DAT 610
C     WRITE (P,480) SUM                                                 DAT 620
      DO 30 I=1,DIML                                                    DAT 630
C  30 WRITE (P,530) I,(PHI(I,J),J=1,DIMW)                               DAT 640
   30 CONTINUE
      REWIND 6                                                          DAT 650
```

```
C       ............................... STRT (STARTING HEAD) ...........DAT 660
   40 READ (R,495) FACT,IVAR,IPRN,IRECS,IRECD                            DAT 670
      IF (IRECS.EQ.1) READ (5,REC=1) STRT                                DAT 680
C     IF ((IVAR.EQ.1.OR.IRECS.EQ.1).AND.IPRN.NE.1) WRITE (P,470)         DAT 690
      DO 80 I=1,DIML                                                     DAT 700
      IF (IVAR.EQ.1) READ (R,540) (STRT(I,J),J=1,DIMW)                   DAT 710
      DO 70 J=1,DIMW                                                     DAT 720
      IF (IRECS.EQ.1) GO TO 60                                           DAT 730
      IF (IVAR.NE.1) GO TO 50                                            DAT 740
      STRT(I,J)=STRT(I,J)*FACT                                           DAT 750
      GO TO 60                                                           DAT 760
   50 STRT(I,J)=FACT                                                     DAT 770
   60 SURI(I,J)=STRT(I,J)                                                DAT 780
      T(I,J)=0.                                                          DAT 785
      TL(I,J)=0.                                                         DAT 790
      SL(I,J)=0.                                                         DAT 800
      TR(I,J)=0.                                                         DAT 810
      TC(I,J)=0.                                                         DAT 820
      WELL(I,J)=0.0                                                      DAT 830
      QRE(I,J)=0.                                                        DAT 840
   70 IF (SUM.EQ.0.0.AND.IDK1.NE.CHK(14)) PHI(I,J)=STRT(I,J)             DAT 850
      IF (IVAR.EQ.0.AND.IRECS.EQ.0.OR.IPRN.EQ.1) GO TO 80                DAT 860
C     WRITE (P,530) I,(STRT(I,J),J=1,DIMW)                               DAT 870
   80 CONTINUE                                                           DAT 880
C     IF (IVAR.NE.1.AND.IRECS.NE.1) WRITE (P,420) FACT                   DAT 890
C     IF (IRECD.EQ.1) WRITE (5,REC=1) STRT                               DAT 900
      IF(KCALLP.EQ.1)RETURN                                              $2-100
  970 DO 990 I=1,DIML                                                    $2-110
      DO 990 J=1,DIMW                                                    $2-120
  990 PHI(I,J)=SURI(I,J)                                                 $2-200
      RETURN                                                             DAT 910
C                                                                        DAT 920
C     ---READ REMAINING ARRAYS FROM CARDS OR DISK (AS SPECIFIED IN THE   DAT 930
C        OPTIONS) AND WRITE THEM ON DISK IF SPECIFIED IN THE OPTIONS---  DAT 940
C     **************                                                     DAT 950
C     ENTRY ARRAY(A,IFMT,IN,IRN)
C     **************                                                     DAT 970
2     CONTINUE
      READ (R,495) FACT,IVAR,IPRN,IRECS,IRECD                            DAT 980
      IK=4*IRECS+2*IVAR+IPRN+1                                           DAT 990
      GO TO (90,90,110,110,140,140), IK                                  DAT1000
   90 DO 100 I=1,DIML                                                    DAT1010
      DO 100 J=1,DIMW
  100 A(I,J)=FACT                                                        DAT1030
C     WRITE (P,430) IN,FACT                                              DAT1040
      GO TO 160                                                          DAT1050
C 110 IF (IK.EQ.3) WRITE (P,440) IN                                      DAT1060
  110 CONTINUE
      DO 130 I=1,DIML                                                    DAT1070
      READ (R,510) (A(I,J),J=1,DIMW)                                     DAT1080
      DO 120 J=1,DIMW                                                    DAT1090
  120 A(I,J)=A(I,J)*FACT                                                 DAT1100
C 130 IF (IK.EQ.3) WRITE (P,IFMT) I,(A(I,J),J=1,DIMW)                    DAT1110
  130 CONTINUE
```

```
      GO TO 160                                                         DAT1120
  140 READ (5,REC=IRN) A                                                DAT1130
      IF (IK.EQ.6) GO TO 160                                            DAT1140
C     WRITE (P,440) IN                                                  DAT1150
      DO 150 I=1,DIML                                                   DAT1160
C 150 WRITE (P,IFMT) I,(A(I,J),J=1,DIMW)                                DAT1170
  150 CONTINUE
C 160 IF (IRECD.EQ.1) WRITE (5,REC=IRN) A                               DAT1180
  160 CONTINUE
      RETURN                                                            DAT1190
C                                                                       DAT1200
C     ---INSERT ZERO VALUES IN THE T OR PERM MATRIX AROUND THE          DAT1210
C        BORDER OF THE MODEL---                                         DAT1220
C     **************                                                    DAT1230
C     ENTRY MDAT(T,TL,PERM,RATE,M,DELX,DELY)
C     **************                                                    DAT1250
3     CONTINUE
      DO 180 I=1,DIML                                                   DAT1260
      DO 180 J=1,DIMW                                                   DAT1270
      IF (WATER.EQ.CHK(2)) GO TO 170                                    DAT1280
      IF (I.EQ.1.OR.I.EQ.DIML.OR.J.EQ.1.OR.J.EQ.DIMW) T(I,J)=0.         DAT1290
      GO TO 180                                                         DAT1300
  170 IF (I.EQ.1.OR.I.EQ.DIML.OR.J.EQ.1.OR.J.EQ.DIMW) PERM(I,J)=0.      DAT1310
  180 CONTINUE                                                          DAT1320
C     ............................... DELX,DELY .......................DAT1330
      READ (R,495) FACT,IVAR,IPRN,IRECS,IRECD                           DAT1340
      IF (IRECS.EQ.1) GO TO 210                                         DAT1350
      IF (IVAR.EQ.1) READ (R,498) DELX                                  DAT1360
      DO 200 J=1,DIMW                                                   DAT1370
      IF (IVAR.NE.1) GO TO 190                                          DAT1380
      DELX(J)=DELX(J)*FACT                                              DAT1390
      GO TO 200                                                         DAT1400
  190 DELX(J)=FACT                                                      DAT1410
  200 CONTINUE                                                          DAT1420
      GO TO 220                                                         DAT1430
  210 READ (5,REC=13) DELX                                              DAT1440
C 220 IF (IRECD.EQ.1) WRITE (5,REC=13) DELX                             DAT1450
  220 CONTINUE
C     IF (IVAR.EQ.1.OR.IRECS.EQ.1.AND.IPRN.NE.1) WRITE (P,550) DELX     DAT1460
C     IF (IVAR.NE.1.AND.IRECS.NE.1) WRITE (P,450) FACT                  DAT1470
      READ (R,495) FACT,IVAR,IPRN,IRECS,IRECD                           DAT1480
      IF (IRECS.EQ.1) GO TO 250                                         DAT1490
      IF (IVAR.EQ.1) READ (R,498) DELY                                  DAT1500
      DO 240 I=1,DIML                                                   DAT1510
      IF (IVAR.NE.1) GO TO 230                                          DAT1520
      DELY(I)=DELY(I)*FACT                                              DAT1530
      GO TO 240                                                         DAT1540
  230 DELY(I)=FACT                                                      DAT1550
  240 CONTINUE                                                          DAT1560
      GO TO 260                                                         DAT1570
  250 READ (5,REC=14) DELY                                              DAT1580
C 260 IF (IRECD.EQ.1) WRITE (5,REC=14) DELY                             DAT1590
  260 CONTINUE
C     IF (IVAR.EQ.1.OR.IRECS.EQ.1.AND.IPRN.NE.1) WRITE (P,560) DELY     DAT1600
```

```
C     IF (IVAR.NE.1.AND.IRECS.NE.1) WRITE (P,460) FACT                   DAT1610
C                                                                        DAT1620
C     ---INITIALIZE VARIABLES---                                         DAT1630
      JNO1=DIMW-1                                                        DAT1640
      INO1=DIML-1                                                        DAT1650
      IF (LEAK.NE.CHK(9).OR.SS.NE.0.) GO TO 280                          DAT1660
      DO 270 I=2,INO1                                                    DAT1670
      DO 270 J=2,JNO1                                                    DAT1680
      IF (M(I,J).EQ.0.) GO TO 270                                        DAT1690
      TL(I,J)=RATE(I,J)/M(I,J)                                           DAT1700
  270 CONTINUE                                                           DAT1710
  280 ETQB=0.0                                                           DAT1720
      ETQD=0.0                                                           DAT1730
      SUBS=0.0                                                           DAT1740
      U=1.0                                                              DAT1750
      TT=0.0                                                             DAT1760
      IM=MIN0(6*DIMW+4,124)                                              DAT1770
      IM=(132-IM)/2                                                      DAT1780
      VF4(3)=DIGIT(IM)                                                   DAT1790
      VF4(8)=DIGIT(IM+5)                                                 DAT1800
      WIDTH=0.                                                           DAT1810
      DO 290 J=2,JNO1                                                    DAT1820
  290 WIDTH=WIDTH+DELX(J)                                                DAT1830
      YDIM=0.                                                            DAT1840
      DO 300 I=2,INO1                                                    DAT1850
  300 YDIM=YDIM+DELY(I)                                                  DAT1860
      RETURN                                                             DAT1870
C     ...................................................................DAT1880
C                                                                        DAT1890
C     ---READ TIME PARAMETERS AND PUMPING DATA FOR A NEW PUMPING PERIOD-DAT1900
C     ************************                                           DAT1910
C     ENTRY NEWPER(PHI,STRT,WELL,SL,RATE,RIVER,M,DELX,DELY,WR,NWR)
C     ************************                                           DAT1930
4     CONTINUE
C                                                                        DAT1940
      IF(KPER.GT.1)GO TO 905                                             $2-210
C--Re-initialize cumulative time and mass balance parameters:            $2-220
      SUM=0.0                                                            $2-230
      SUMP=0.0                                                           $2-240
      PUMPT=0.0                                                          $2-250
      CFLUXT=0.0                                                         $2-260
      QRET=0.0                                                           $2-270
      CHST=0.0                                                           $2-280
      CHDT=0.0                                                           $2-290
      FLUXT=0.0                                                          $2-300
      STORT=0.0                                                          $2-310
      ETFLXT=0.0                                                         $2-320
      FLXNT=0.0                                                          $2-330
      ETQB=0.0                                                           $2-340
      ETQD=0.0                                                           $2-350
      SUBS=0.0                                                           $2-360
      U=1.0                                                              $2-370
      TT=0.0                                                             $2-380
  905 CONTINUE                                                           $2-390
```

```
C--Comment out the TRESCOTT read statement for pumping period info:     $2-391
C     READ (R,505) KP,KPM1,NWEL,TMAX,NUMT,CDLT,DELT                    $$2-392
C--Get pumping period data from "AQMAN" COMMON:                         $2-480
  910 CALL READ1(1,KP,KPM1,NWEL,TMAX,NUMT,DELT,CDLT,M1,M2,M3,M4,M5)     $2-490
C                                                                       DAT1960
C     ---COMPUTE ACTUAL DELT AND NUMT---                                DAT1970
      DT=DELT/24.                                                       DAT1980
      TM=0.0                                                            DAT1990
      DO 310 I=1,NUMT                                                   DAT2000
      DT=CDLT*DT                                                        DAT2010
      TM=TM+DT                                                          DAT2020
      IF (TM.GE.TMAX) GO TO 320                                         DAT2030
  310 CONTINUE                                                          DAT2040
      GO TO 330                                                         DAT2050
  320 DELT=TMAX/TM*DELT                                                 DAT2060
      NUMT=I                                                            DAT2070
C 330 WRITE (P,570) KP,TMAX,NUMT,DELT,CDLT                              DAT2080
  330 CONTINUE
      DELT=DELT*3600.                                                   DAT2090
      TMAX=TMAX*86400.                                                  DAT2100
C                                                                       DAT2110
C     ---INITIALIZE SUMP, STRT, SL, WELL AND WR---                      DAT2120
C     WRITE (P,580) NWEL                                                DAT2130
      IF (KP.GT.KPM1) SUMP=0.                                           DAT2140
      DO 350 I=1,DIML                                                   DAT2150
      DO 350 J=1,DIMW                                                   DAT2160
      IF (KP.EQ.KPM1) GO TO 340                                         DAT2170
      STRT(I,J)=PHI(I,J)                                                DAT2180
  340 IF (LEAK.NE.CHK(9)) GO TO 350                                     DAT2190
      IF (M(I,J).EQ.0.) GO TO 350                                       DAT2200
      SL(I,J)=RATE(I,J)/M(I,J)*(RIVER(I,J)-STRT(I,J))                   DAT2210
  350 WELL(I,J)=0.                                                      DAT2220
C--If start of a new call of TRES, set starting heads to original       $2-550
C---initial heads:                                                      $2-560
      IF(KPER.NE.1)GO TO 930                                            $2-570
      DO 920 I=1,DIML                                                   $2-580
      DO 920 J=1,DIMW                                                   $2-590
      STRT(I,J)=SURI(I,J)                                               $2-600
  920 CONTINUE                                                          $2-610
  930 CONTINUE                                                          $2-620
      IF (NW.EQ.0) GO TO 370                                            DAT2230
      DO 360 I=1,NW                                                     DAT2240
  360 WR(I)=0.                                                          DAT2250
  370 CONTINUE
      IF (NWEL.EQ.0) GO TO 410                                          DAT2260
C                                                                       DAT2270
C     ---READ AND WRITE WELL PUMPING RATES AND WELL RADII---            DAT2280
      KW=0                                                              DAT2290
      DO 400 II=1,NWEL                                                  DAT2300
C--Comment out the TRESCOTT read statement for well data:               $2-670
C     READ (R,490) I,J,WELL(I,J),RADIUS                                $$2-680
C--Retrieve well locations, radii, and  pumpages, from "AQMAN" COMMON:  $2-690
  940 CALL READ1(2,M1,M2,M3,M4,M5,M6,M7,I,J,II,RADIUS,XWELL)            $2-700
      WELL(I,J)=XWELL                                                   $2-710
```

```
      IF (RADIUS.EQ.0.) GO TO 380                                       DAT2320
      KW=KW+1                                                           DAT2330
      IF (KW.GT.NW) GO TO 380                                           DAT2340
      NWR(KW,1)=I                                                       DAT2350
      NWR(KW,2)=J                                                       DAT2360
      WR(KW)=RADIUS                                                     DAT2370
C     WRITE (P,590) I,J,WELL(I,J),WR(KW)                                DAT2380
      GO TO 390                                                         DAT2390
C 380 WRITE (P,590) I,J,WELL(I,J)                                       DAT2400
  380 CONTINUE
  390 WELL(I,J)=WELL(I,J)/(DELX(J)*DELY(I))                             DAT2410
  400 CONTINUE                                                          DAT2420
C--Increase pumping period counter:                                     $2-720
  410 KPER=KPER+1                                                       $2-730
      RETURN                                                            DAT2430
C     ..................................................................DAT2440
C                                                                       DAT2450
C     FORMATS:                                                          DAT2460
C                                                                       DAT2470
C     ------------------------------------------------------------------DAT2480
C                                                                       DAT2490
C                                                                       DAT2500
  420 FORMAT ('0',63X,'STARTING HEAD =',G15.7)                          DAT2510
  430 FORMAT ('0',41X,9A4,'=',G15.7)                                    DAT2520
  440 FORMAT ('1',49X,9A4,/,65X,'MATRIX',/,50X,36('-'))                 DAT2530
  450 FORMAT ('0',72X,'DELX =',G15.7)                                   DAT2540
  460 FORMAT ('0',72X,'DELY =',G15.7)                                   DAT2550
  470 FORMAT ('1',60X,'STARTING HEAD MATRIX'/61X,20('-'))               DAT2560
  480 FORMAT ('1',40X,' CONTINUATION - HEAD AFTER ',G20.7,' SEC PUMPING DAT2570
     1'/42X,58('-'))                                                    DAT2580
  485 FORMAT (3G10.0)
  490 FORMAT (2I10,5G10.0,I10,3G10.0)
  495 FORMAT (G10.0,4I10)
  498 FORMAT (8G10.0)
  500 FORMAT (A4,6X,5G10.0,A6)                                          DAT2600
  505 FORMAT (3I10,G10.0,I10,2G10.0)
  510 FORMAT (20F4.0)                                                   DAT2610
  520 FORMAT ('0',51X,'NUMBER OF PUMPING PERIODS =',I5/49X,'TIME STEPS BDAT2620
     1ETWEEN PRINTOUTS =',I5//51X,'ERROR CRITERION FOR CLOSURE =',G15.7/DAT2630
     241X,'          STEADY STATE ERROR CRITERION =',G15.7//44X,'SPECIFICDAT2640
     3 STORAGE OF CONFINING BED =',G15.7/54X,'EVAPOTRANSPIRATION RATE ='DAT2650
     4,G15.7/56X,'EFFECTIVE DEPTH OF ET =',G15.7//22X,'MULTIPLICATION FADAT2660
     5CTOR FOR TRANSMISSIVITY IN X DIRECTION =',G15.7/63X,'IN Y DIRECTIODAT2670
     6N =',G15.7)                                                       DAT2680
  530 FORMAT ('0',I2,2X,20F6.1/(5X,20F6.1))                             DAT2690
  540 FORMAT (8F10.4)                                                   DAT2700
  550 FORMAT (1H1,46X,40HGRID SPACING IN PROTOTYPE IN X DIRECTION/47X,40DAT2710
     1('-')//('0',12F10.0))                                             DAT2720
  560 FORMAT (1H-,46X,40HGRID SPACING IN PROTOTYPE IN Y DIRECTION/47X,40DAT2730
     1('-')//('0',12F10.0))                                             DAT2740
  570 FORMAT ('-',50X,'PUMPING PERIOD NO.',I4,':',F10.2,' DAYS'/51X,38('DAT2750
     1-')//53X,'NUMBER OF TIME STEPS=',I6//59X,'DELT IN HOURS =',F10.3//DAT2760
     253X,'MULTIPLIER FOR DELT =',F10.3)                                DAT2770
  580 FORMAT ('-',63X,I4,' WELLS'/65X,9('-')//50X,'I',9X,'J    PUMPING RDAT2780
```

```
     1ATE  WELL RADIUS'/)
  590 FORMAT (41X,2I10,2F13.4)
  600 FORMAT (4G20.10)
  605 FORMAT (3G20.10)
  610 FORMAT ('0',30X,'ON ALPHAMERIC MAP:'/40X,'MULTIPLICATION FACTOR FO
     1R X DIMENSION =',G15.7/40X,'MULTIPLICATION FACTOR FOR Y DIMENSION
     2=',G15.7/55X,'MAP SCALE IN UNITS OF  ',A11/50X,'NUMBER OF ',A8,' P
     3ER INCH =',G15.7/43X,'MULTIPLICATION FACTOR FOR DRAWDOWN =',G15.7/
     447X,'MULTIPLICATION FACTOR FOR HEAD =',G15.7)
      END
      SUBROUTINE STEP(PHI,KEEP,STRT,SURI,T,WELL,PERM,BOTTOM,TOP,DELX,DDN
     1,DELY,WR,NWR,TEST3,TR,TC,S,QRE,TL,SY,RATE,RIVER,M,GRND,NENT)
C     ------------------------------------------------------------------
C     INITIALIZE DATA FOR TIME STEP, CHECK FOR STEADY STATE,
C     PRINT AND  PUNCH RESULTS
C     ------------------------------------------------------------------
C
C     SPECIFICATIONS:
      IMPLICIT REAL *8 (A-H,O-Z)
      REAL *8 MESUR,KEEP,M
      REAL *4 MINS,CASE,TYPE,NNAME
      INTEGER R,P,PU,DIML,DIMW,YYY
      INTEGER CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM
     1,HEAD,CONTR,LEAK,RECH,SIP,ADI,NUMS,IDK1,IDK2
C
      DIMENSION PHI(IZ,JZ), KEEP(IZ,JZ), STRT(IZ,JZ), SURI(IZ,JZ), T(IZ,
     1JZ), BOTTOM(IP,JP), WELL(IZ,JZ), PERM(IP,JP), TOP(IC,JC), DELX(JZ)
     2,DDN(JZ), DELY(IZ), WR(IH), NWR(IH,2), ITTO(200), TEST3(IMX1),
     3TR(IZ,JZ),TC(IZ,JZ),S(IZ,JZ),QRE(IZ,JZ),TL(IZ,JZ),SY(IP,JP),
     4RATE(IR,JR),RIVER(IR,JR),M(IR,JR),GRND(IL,JL)
      DIMENSION TYPE(4)
C
      COMMON /SARRAY/ VF4(11),CHK(15)
      COMMON /SPARAMI/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,LEAK,
     1RECH,SIP,NUMS,LSOR,ADI,IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,
     2ITMAX,LENGTH,NWEL,NW,DIML,DIMW,I,J,IDK1,IDK2,JNO1,INO1,R,P,PU
      COMMON /SPARAMR/ U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,
     1WIDTH,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,EROR
      COMMON /CK/ ETFLXT,STORT,QRET,CHST,CHDT,FLUXT,PUMPT,CFLUXT,FLXNT
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1
      COMMON /PRI/ NA(4),N1,N2,N3
      COMMON /PRR/ PRNT(122),DIGIT(122),XN(100),BLANK(60),SYM(17),
     1YN(13),VF1(6),VF2(6),VF3(7),YLABEL(6),TITLE(5),XLABEL(3),XN1,
     2MESUR,XSCALE,YSCALE,DINCH,FACT1,FACT2
C
      COMMON /ALPHAS/ NNAME(16),CASE
      COMMON /PARAMS/ KCALLP,NWLS,NCNTR,NNPER,KPER,CDELT,TIMPER,
     & TIMINC(50)
      DATA TYPE/'LINE','NONL','LINE','QUAD'/
      DATA PIE/3.141593/,YYY/:00000000/
      GO TO (1,2,3,4,5,6) NENT
      RETURN
C     ..................................................................
C
```

```
C     ---START A NEW TIME STEP---
C     ********************
C     ENTRY NEWSTP(PHI,KEEP)
C     ********************
1     CONTINUE
      KT=KT+1
      KOUNT=0
      DO 10 I=1,DIML
      DO 10 J=1,DIMW
   10 KEEP(I,J)=PHI(I,J)
      DELT=CDLT*DELT
      SUM=SUM+DELT
      SUMP=SUMP+DELT
      DAYSP=SUMP/86400.
      YRSP=DAYSP/365.
      HRS=SUM/3600.
      MINS=HRS*60.
      DAYS=HRS/24.
      YRS=DAYS/365.
      RETURN
C     ..................................................................
C
C     ---CHECK FOR STEADY STATE---
C     *********************
C     ENTRY STEADY(PHI,KEEP,STRT,SURI,T,WELL,PERM,BOTTOM,TOP,DELX,
C    1DDN,DELY,WR,NWR,TEST3,TR,TC,S,QRE,TL,SY,RATE,RIVER,
C    2M,GRND)
C     *********************
2     CONTINUE
      TEST2=0.
      DO 20 I=2,INO1
      DO 20 J=2,JNO1
      DTT=DABS(DBLE(KEEP(I,J)-PHI(I,J)))
   20 TEST2=DMAX1(TEST2,DTT)
      IF (TEST2.GE.EROR) GO TO 30
C     WRITE (P,330) KT
      IFINAL=1
      GO TO 40
   30 IF (KT.EQ.NUMT) IFINAL=1
C
C     ---ENTRY FOR TERMINATING COMPUTATIONS IF MAXIMUM ITERATIONS
C     EXCEEDED---
C     *********************
C     ENTRY TERM1(PHI,KEEP,STRT,SURI,T,WELL,PERM,BOTTOM,TOP,DELX,
C    1DDN,DELY,WR,NWR,TEST3,TR,TC,S,QRE,TL,SY,RATE,RIVER,M,GRND)
C     *********************
3     CONTINUE
   40 IF (KT.GT.200) WRITE (P,400)
      ITTO(KT)=KOUNT
      IF (KOUNT.LE.ITMAX) GO TO 80
      IERR=2
      KOUNT=KOUNT-1
      ITTO(KT)=KOUNT
      IF (KT.EQ.1) GO TO 60
```

```
C                                                                       STP 810
C      ---WRITE ON DISK OR PUNCH CARDS AS SPECIFIED IN THE OPTIONS---   STP 820
       XXX=SUM-DELT                                                     STP 830
C      IF (IDK2.EQ.CHK(15)) WRITE (6) ((KEEP(I,J),YYY,I=1,DIML),J=1,DIMW)STP 840
C     1,XXX,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFLXT,FLXNT   STP 850
       IF (PNCH.NE.CHK(1)) GO TO 80                                     STP 860
C      WRITE (PU,360) XXX,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETSTP 870
C     1FLXT,FLXNT                                                       STP 880
       DO 50 I=1,DIML                                                   STP 890
C   50 WRITE (PU,350) (KEEP(I,J),J=1,DIMW)                              STP 900
   50 CONTINUE
       GO TO 80                                                         STP 910
C   60 IF (IDK2.EQ.CHK(15)) WRITE (6) PHI,SUM,SUMP,PUMPT,CFLUXT,QRET,CHSTSTP 920
C     1,CHDT,FLUXT,STORT,ETFLXT,FLXNT                                   STP 930
   60 CONTINUE
       IF (PNCH.NE.CHK(1)) GO TO 80                                     STP 940
C      WRITE (PU,360) SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETSTP 950
C     1FLXT,FLXNT                                                       STP 960
       DO 70 I=1,DIML                                                   STP 970
C   70 WRITE (PU,350) (PHI(I,J),J=1,DIMW)                               STP 980
   70 CONTINUE
C                                                                       STP 990
  80   IF (CHCK.EQ.CHK(5)) THEN
       CALL CHECKI(PHI,KEEP,PHE,STRT,T,TR,TC,S,QRE,WELL,TL,PERM,BOTTOM
      1,SY,RATE,RIVER,M,TOP,GRND,DELX,DELY,1)
       ENDIF
       IF (IERR.EQ.2) GO TO 90                                          STP1010
C                                                                       STP1020
C      ---PRINT OUTPUT AT DESIGNATED TIME STEPS---                      STP1030
C--Only "print" results (ie, Call WRITE1, etc) at end of a pumping      $3-201
C----period. The TRESCOTT variable KTH is thus ignored.                 $3-202
       IF (IFINAL.NE.1) RETURN                                          $$-203
C   90 WRITE (P,340) KT,DELT,SUM,MINS,HRS,DAYS,YRS,DAYSP,YRSP           STP1050
   90 CONTINUE
       IF (CHCK.EQ.CHK(5)) THEN
       CALL CHECKI(PHI,KEEP,PHE,STRT,T,TR,TC,S,QRE,WELL,TL,PERM,BOTTOM
      1,SY,RATE,RIVER,M,TOP,GRND,DELX,DELY,2)
       ENDIF
C      IF (TT.NE.0.) WRITE (P,320) TMIN,TT                              STP1070
       KOUNT=KOUNT+1                                                    STP1080
C      WRITE(P,300) (TEST3(J),J=1,KOUNT)
C      WRITE (P,290) TEST2                                              STP1100
       I3=1                                                             STP1110
       I5=0                                                             STP1120
  100 I5=I5+40                                                          STP1130
       I4=MIN0(KT,I5)                                                   STP1140
C      WRITE (P,390) (I,I=I3,I4)                                        STP1150
C      WRITE (P,380)                                                    STP1160
C      WRITE (P,370) (ITTO(I),I=I3,I4)                                  STP1170
C      WRITE (P,380)                                                    STP1180
       IF (KT.LE.I5) GO TO 110                                          STP1190
       I3=I3+40                                                         STP1200
       GO TO 100                                                        STP1210
C                                                                       STP1220
```

```
C       ---PRINT ALPHAMERIC MAPS---                                     STP1230
  110 IF (CONTR.NE.CHK(3)) GO TO 120                                    STP1240
      IF(FACT1.NE.0.) THEN
      CALL PRNTAI(PHI,SURI,T,S,WELL,DELX,DELY,1,2)
      ENDIF
      IF (FACT2.NE.0.) THEN
      CALL PRNTAI(PHI,SURI,T,S,WELL,DELX,DELY,2,2)
      ENDIF
  120 CONTINUE                                                          $3-209
C                                                                       STP1280
C--Convert head matrix PHI to vector XHEAD and store in "AQMAN" COMMON: $3-210
      CALL WRITE1(1,IZ,JZ,KW,HW,PHI)                                    $3-240
C       ---PRINT HEAD MATRIX---                                         STP1290
C     WRITE (P,310)                                                     STP1300
      DO 130 I=1,DIML                                                   STP1310
C 130 WRITE (P,VF4) I,(PHI(I,J),J=1,DIMW)                               STP1320
  130 CONTINUE
  140 IF (NUM.NE.CHK(4)) GO TO 170                                      STP1330
C                                                                       STP1340
C       ---PRINT DRAWDOWN---                                            STP1350
C     WRITE (P,280)                                                     STP1360
C     ********************                                              STP1370
C     ENTRY DRDN(PHI,SURI,T,WELL,PERM,BOTTOM,TOP,DELX,
C    1DDN,DELY,WR,NWR)
C     ********************                                              STP1390
4     CONTINUE
      DO 160 I=1,DIML                                                   STP1400
      DO 150 J=1,DIMW                                                   STP1410
  150 DDN(J)=SURI(I,J)-PHI(I,J)                                         STP1420
C 160 WRITE (P,VF4) I,(DDN(J),J=1,DIMW)                                 STP1430
  160 CONTINUE
  170 IF (NW.EQ.0.OR.IERR.EQ.1) GO TO 230                               STP1440
C     ..................................................................STP1450
C                                                                       STP1460
C       ---COMPUTE APPROXIMATE HEAD FOR PUMPING WELLS---                STP1470
C     WRITE (P,260)                                                     STP1480
      DO 220 KW=1,NW                                                    STP1490
      IF (WR(KW).EQ.0.) GO TO 220                                       STP1500
      I=NWR(KW,1)                                                       STP1510
      J=NWR(KW,2)                                                       STP1520
C                                                                       STP1530
C     COMPUTE EFFECTIVE RADIUS OF WELL IN MODEL---                      STP1540
      RE=(DELX(J)+DELY(I))/9.62                                         STP1550
      IF (WATER.NE.CHK(2)) GO TO 180                                    STP1560
      IF (CONVRT.NE.CHK(7)) GO TO 190                                   STP1570
      IF (PHI(I,J).LT.TOP(I,J)) GO TO 190                               STP1580
C                                                                       STP1590
C       ---COMPUTATION FOR WELL IN ARTESIAN AQUIFER---                  STP1600
  180 HW=PHI(I,J)+WELL(I,J)*DLOG(RE/WR(KW))/(2.*PIE*T(I,J))*DELX(J)*DELY$$3-330
     1(I)                                                               STP1620
      GO TO 210                                                         STP1630
C                                                                       STP1640
C       ---COMPUTATION FOR WELL IN WATER TABLE AQUIFER                  STP1650
  190 HED=PHI(I,J)-BOTTOM(I,J)                                          STP1660
```

```
      ARG=HED*HED+WELL(I,J)*DLOG(RE/WR(KW))/(PIE*PERM(I,J))*DELX(J)*DELY$$3-340
     1(I)                                                              STP1680
      IF (ARG.GT.0.) GO TO 200                                         STP1690
      ARGDRY=-WELL(I,J)*DLOG(RE/WR(KW))/(PIE*PERM(I,J))*DELX(J)*DELY(I) $3-350
      HW=PHI(I,J)-DSQRT(ARGDRY)+BOTTOM(I,J)                           . $3-360
      WRITE (P,270) I,J                                                STP1700
      WRITE (P,273)I,J,HW                                              $3-370
C     GO TO 220                                                        STP1710
      GO TO 210                                                        $3-380
  200 HW=DSQRT(ARG)+BOTTOM(I,J)                                        $$3-390
C                                                                      STP1730
C     ---COMPUTE DRAWDOWN AT THE WELL AND PRINT RESULTS---             STP1740
  210 DRAW=SURI(I,J)-HW                                                STP1750
C     WRITE (P,250) I,J,WR(KW),HW,DRAW                                 STP1760
C--Enter in-well heads into XHEAD:                                     $3-400
      CALL WRITE1(2,IZ,JZ,KW,HW,PHI)                                   $3-450
  220 CONTINUE                                                         STP1770
  230 IF (IERR.NE.2) RETURN                                            STP1780
      STOP                                                             STP1790
C                                                                      STP1800
C     ---DISK OUTPUT---                                                STP1810
C     ****************                                                 STP1820
C     ENTRY DISK(PHI)
C     ****************                                                 STP1840
5     CONTINUE
C     WRITE (6) PHI,SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETFSTP1850
C    1LXT,FLXNT                                                        STP1860
      RETURN                                                           STP1870
C     ...............................................................STP1880
C                                                                      STP1890
C     ---PUNCHED OUTPUT---                                             STP1900
C     ********************                                             STP1910
C     ENTRY PUNCH(PHI)
C     ********************                                             STP1930
6     CONTINUE
C     WRITE (PU,360) SUM,SUMP,PUMPT,CFLUXT,QRET,CHST,CHDT,FLUXT,STORT,ETSTP1940
C    1FLXT,FLXNT                                                       STP1950
      DO 240 I=1,DIML                                                  STP1960
C 240 WRITE (PU,350) (PHI(I,J),J=1,DIMW)                               STP1970
  240 CONTINUE
      RETURN                                                           STP1980
C                                                                      STP1990
C     ...............................................................STP2000
C                                                                      STP2010
C     FORMATS:                                                         STP2020
C                                                                      STP2030
C                                                                      STP2040
C     ---------------------------------------------------------------STP2050
C                                                                      STP2060
C                                                                      STP2070
  250 FORMAT (' ',43X,2I5,3F11.2)                                      STP2080
  260 FORMAT ('-',50X,'HEAD AND DRAWDOWN IN PUMPING WELLS'/51X,34('-')//STP2090
     148X,'I    J   WELL RADIUS    HEAD    DRAWDOWN'//)                 STP2100
  270 FORMAT (' ',43X,2I5,'  WELL IS DRY')                             STP2110
```

```
  273 FORMAT (' ',10X,'IN-WELL HEAD AT NODE ',2I5,'IS ',F12.4)          $3-460
  280 FORMAT (1H1,60X,'DRAWDOWN'/61X,8('-'))                            STP2120
  290 FORMAT ('OMAXIMUM CHANGE IN HEAD FOR THIS TIME STEP =',F10.3/' ',5STP2130
     13('-'))                                                           STP2140
 300  FORMAT('OMAXIMUM HEAD CHANGE FOR EACH ITERATION',
     1/' ',39('-')/('0',10F12.4))
  310 FORMAT ('1',60X,'HEAD MATRIX'/61X,11('-'))                        STP2170
  320 FORMAT ('ODIMENSIONLESS TIME FOR THIS STEP RANGES FROM',G15.7,'  TSTP2180
     1O',G15.7)                                                         STP2190
  330 FORMAT ('-*****STEADY STATE AT TIME STEP',I4,'*****')             STP2200
  340 FORMAT (1H1,44X,57('-')/45X,'[',14X,'TIME STEP NUMBER =',I9,14X,'[STP2210
     1'/45X,57('-')//50X,29HSIZE OF TIME STEP IN SECONDS=,F14.2//55X,'TOSTP2220
     2TAL SIMULATION TIME IN SECONDS=',F14.2/80X,8HMINUTES=,F14.2/82X,6HSTP2230
     3HOURS=,F14.2/83X,5HDAYS=,F14.2/82X,'YEARS=',F14.2///45X,'DURATION STP2240
     4OF CURRENT PUMPING PERIOD IN DAYS=',F14.2/82X,'YEARS=',F14.2//)   STP2250
  350 FORMAT (8F10.4)                                                   STP2260
  360 FORMAT (4G20.10)                                                  STP2270
  370 FORMAT ('OITERATIONS:',40I4)                                      STP2280
  380 FORMAT (' ',10('-'))                                              STP2290
  390 FORMAT ('OTIME STEP :',40I4)                                      STP2300
  400 FORMAT ('0',10('*'),'THE NUMBER OF TIME STEPS EXCEEDS THE DIMENSIOSTP2310
     1N OF THE VECTOR ITTO AND MAY CAUSE UNEXPECTED RESULTS IN ADDITIONASTP2320
     2L'/'OCOMPUTATION.  AVOID PROBLEMS BY INCREASING THE DIMENSION OF TSTP2330
     3HE VECTOR ITTO IN STEP',10('*'))                                  STP2340
      END                                                               STP2350-
      SUBROUTINE SOLVE1(PHI,BE,G,TEMP,KEEP,PHE,STRT,T,S,QRE,WELL,TL,SL,DSIP  10
     1EL,ETA,V,XI,DELX,BET,DELY,ALF,TEST3,TR,TC,GRND,SY,TOP,RATE,M,
     2RIVER,SURI,PERM,BOTTOM,DDN,WR,NWR,NENT)
C     ------------------------------------------------------------------SIP  40
C      SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE                      SIP  50
C     ------------------------------------------------------------------SIP  60
C                                                                       SIP  70
C     SPECIFICATIONS:                                                   SIP  80
      IMPLICIT REAL *8 (A-H,O-Z)                                        $4-10
      REAL *8 KEEP,M                                                    $$4-11
      INTEGER R,P,PU,DIML,DIMW,IORDER(21)
      INTEGER  CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD
     1,CONTR,LEAK,RECH,SIP,ADI,NUMS,IDK1,IDK2
C                                                                       SIP 140
      DIMENSION PHI(1),BE(1),G(1),TEMP(1),KEEP(1),PHE(1),STRT(1)
     1,T(1),S(1),QRE(1),WELL(1),TL(1),SL(1),DEL(1),ETA(1),V(1)
     2,XI(1),DELX(1),BET(1),DELY(1),ALF(1),TEST3(1),TR(1),TC(1)
     3,GRND(1),SY(1),TOP(1),RATE(1),M(1),RIVER(1),SURI(1),
     4PERM(1),BOTTOM(1),DDN(1),WR(1),NWR(1)
      DIMENSION RHOP(20)                                                $4-20
C                                                                       SIP 190
      COMMON /SARRAY/ VF4(11),CHK(15)                                   SIP 200
      COMMON /SPARAMI/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,LEAK,
     1RECH,SIP,NUMS,LSOR,ADI,IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,
     2ITMAX,LENGTH,NWEL,NW,DIML,DIMW,I,J,IDK1,IDK2,JNO1,INO1,R,P,PU
      COMMON /SPARAMR/ U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,
     1WIDTH,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,EROR
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1
      GO TO (1,2) NENT
```

```
      RETURN                                                            SIP 260
C     ..................................................................SIP 270
C                                                                       SIP 280
C     ---COMPUTE AND PRINT ITERATION PARAMETERS---                      SIP 290
C     *******************                                               SIP 300
C     ENTRY ITER1(PHI,BE,G,TEMP,KEEP,PHE,STRT,T,S,QRE,WELL,TL,SL,
C    1DEL,ETA,V,XI,DELX,BET,DELY,ALF,TEST3,TR,TC,GRND,SY,TOP,RATE,M,
C    2RIVER)
C     *******************                                               SIP 320
1     CONTINUE
C     ---INITIALIZE ORDER OF ITERATION PARAMETERS (OR REPLACE WITH A    SIP 330
C     READ STATEMENT)---                                                SIP 340
      DATA IORDER/1,2,3,4,5,1,2,3,4,5,11*1/
      I2=INO1-1                                                         SIP 360
      J2=JNO1-1                                                         SIP 370
      L2=LENGTH/2                                                       SIP 380
      PL2=L2-1.                                                         SIP 390
      W=0.                                                              SIP 400
      PI=0.                                                             SIP 410
C                                                                       SIP 420
C     ---COMPUTE AVERAGE MAXIMUM PARAMETER FOR PROBLEM---               SIP 430
      DO 10 I=2,INO1                                                    SIP 440
      DO 10 J=2,JNO1                                                    SIP 450
      N=I+DIML*(J-1)                                                    SIP 460
      IF (T(N).EQ.0.) GO TO 10                                          SIP 470
      PI=PI+1.                                                          SIP 480
      DX=DELX(J)/WIDTH                                                  SIP 490
      DY=DELY(I)/YDIM                                                   SIP 500
      W=W+1.-DMIN1(2.*DX*DX/(1.+FACTY*DX*DX/(FACTX*DY*DY)),2.*DY*DY/(1.+$$4-30
     1FACTX*DY*DY/(FACTY*DX*DX)))                                       SIP 520
   10 CONTINUE                                                          SIP 530
      W=W/PI                                                            SIP 540
C                                                                       SIP 550
C     ---COMPUTE PARAMETERS IN GEOMETRIC SEQUENCE---                    SIP 560
      PJ=-1.                                                            SIP 570
      DO 20 I=1,L2                                                      SIP 580
      PJ=PJ+1.                                                          SIP 590
   20 TEMP(I)=1.-(1.-W)**(PJ/PL2)                                       SIP 600
C                                                                       SIP 610
C     ---ORDER SEQUENCE OF PARAMETERS---                                SIP 620
      DO 30 J=1,LENGTH                                                  SIP 630
   30 RHOP(J)=TEMP(IORDER(J))                                           SIP 640
C     WRITE (P,370) HMAX                                                SIP 650
C     WRITE (P,380) LENGTH,(RHOP(J),J=1,LENGTH)                         SIP 660
      RETURN                                                            SIP 670
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     ---INIITIALIZE DATA FOR A NEW ITERATION----
C
   40 KOUNT=KOUNT+1                                                     SIP 710
      IF (KOUNT.LE.ITMAX) GO TO 50                                      SIP 720
      WRITE (P,360)                                                     SIP 730
      CALL STEP(PHI,KEEP,STRT,SURI,T,WELL,PERM,BOTTOM,TOP,DELX,DDN
     1,DELY,WR,NWR,TEST3,TR,TC,S,QRE,TL,SY,RATE,RIVER,M,GRND,3)
   50 IF (MOD(KOUNT,LENGTH)) 60,60,70                                   SIP 750
```

```
C      ************************                                       SIP 760
C      ENTRY NEWITA(PHI,BE,G,TEMP,KEEP,PHE,STRT,T,S,QRE,WELL,TL,SL,
C     1DEL,ETA,V,XI,DELX,BET,DELY,ALF,TEST3,TR,TC,GRND,SY,TOP,RATE,M,
C     2RIVER,SURI,PERM,BOTTOM,DDN,WR,NWR)
C      ************************                                       SIP 780
2      CONTINUE
C      ...............................................................SIP 680
C                                                                     SIP 690
C      ---INITIALIZE DATA FOR A NEW ITERATION---                      SIP 700
   60 NTH=0                                                           SIP 790
   70 NTH=NTH+1                                                       SIP 800
      W=RHOP(NTH)                                                     SIP 810
      TEST3(KOUNT+1)=0.                                               SIP 820
      TEST=0.                                                         SIP 830
      N=DIML*DIMW                                                     SIP 840
      DO 80 I=1,N                                                     SIP 850
      PHE(I)=PHI(I)                                                   SIP 860
      DEL(I)=0.                                                       SIP 870
      ETA(I)=0.                                                       SIP 880
      V(I)=0.                                                         SIP 890
   80 XI(I)=0.                                                        SIP 900
      BIGI=0.0                                                        SIP 910
C                                                                     SIP 920
C      ---COMPUTE TRANSMISSIVITY AND T COEFFICIENTS IN WATER TABLE    SIP 930
C      OR WATER TABLE-ARTESIAN SIMUATION---                           SIP 940
      IF (WATER.NE.CHK(2)) GO TO 90                                   SIP 950
      CALL COEF(PHI,KEEP,PHE,STRT,SURI,T,TR,TC,S,WELL,TL,SL,PERM,BOTTOM
     1,SY,RATE,RIVER,M,TOP,GRND,DELX,DELY,DDN,WR,NWR,2)
C                                                                     SIP 970
C      ---CHOOSE SIP NORMAL OR REVERSE ALGORITHM---                   SIP 980
   90 IF (MOD(KOUNT,2)) 100,230,100                                   SIP 990
C      ...............................................................SIP1000
C      ---ORDER EQUATIONS WITH ROW 1 FIRST - 3X3 EXAMPLE:             SIP1010
C               1 2 3                                                 SIP1020
C               4 5 6                                                 SIP1030
C               7 8 9                                                 SIP1040
C      ...............................................................SIP1050
  100 DO 210 I=2,INO1                                                 SIP1060
      DO 210 J=2,JNO1                                                 SIP1070
      N=I+DIML*(J-1)                                                  SIP1080
      NL=N-DIML                                                       SIP1090
      NR=N+DIML                                                       SIP1100
      NA=N-1                                                          SIP1110
      NB=N+1                                                          SIP1120
C                                                                     SIP1130
C      ---SKIP COMPUTATIONS IF NODE IS OUTSIDE AQUIFER BOUNDARY---    SIP1140
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 210                         SIP1150
C                                                                     SIP1160
C      ---COMPUTE COEFFICIENTS---                                     SIP1170
      D=TR(NL)/DELX(J)                                                SIP1180
      F=TR(N)/DELX(J)                                                 SIP1190
      B=TC(NA)/DELY(I)                                                SIP1200
      H=TC(N)/DELY(I)                                                 SIP1210
      IF (EVAP.NE.CHK(6)) GO TO 120                                   SIP1220
```

```
C                                                                       SIP1230
C     ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---              SIP1240
      ETQB=0.                                                           SIP1250
      ETQD=0.0                                                          SIP1260
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 120                           SIP1270
      IF (PHE(N).GT.GRND(N)) GO TO 110                                  SIP1280
      ETQB=QET/ETDIST                                                   SIP1290
      ETQD=ETQB*(ETDIST-GRND(N))                                        SIP1300
      GO TO 120                                                         SIP1310
  110 ETQD=QET                                                          SIP1320
C                                                                       SIP1330
C     ---COMPUTE STORAGE TERM---                                        SIP1340
  120 IF (CONVRT.EQ.CHK(7)) GO TO 130                                   SIP1350
      RHO=S(N)/DELT                                                     SIP1360
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT                               SIP1370
      GO TO 200                                                         SIP1380
C                                                                       SIP1390
C     ---COMPUTE STORAGE COEFFICIENT FOR CONVERSION PROBLEM---          SIP1400
  130 SUBS=0.0                                                          SIP1410
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 170             SIP1420
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 160             SIP1430
      IF (KEEP(N)-PHE(N)) 140,150,150                                   SIP1440
  140 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))                           SIP1450
      GO TO 170                                                         SIP1460
  150 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))                           SIP1470
  160 RHO=SY(N)/DELT                                                    SIP1480
      GO TO 180                                                         SIP1490
  170 RHO=S(N)/DELT                                                     SIP1500
  180 IF (LEAK.NE.CHK(9)) GO TO 200                                     SIP1510
C                                                                       SIP1520
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---          SIP1530
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 200                        SIP1540
      HED1=DMAX1(STRT(N),TOP(N))                                        $$4-40
      U=1.                                                              SIP1560
      HED2=0.                                                           SIP1570
      IF (PHE(N).GE.TOP(N)) GO TO 190                                   SIP1580
      HED2=TOP(N)                                                       SIP1590
      U=0.                                                              SIP1600
  190 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))      SIP1610
  200 CONTINUE                                                          SIP1620
C                                                                       SIP1630
C     ---SIP 'NORMAL' ALGORITHM---                                      SIP1640
C     ---FORWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR V---         SIP1650
      E=-B-D-F-H-RHO-TL(N)*U-ETQB                                       SIP1660
      CH=DEL(NA)*B/(1.+W*DEL(NA))                                       SIP1670
      GH=ETA(NL)*D/(1.+W*ETA(NL))                                       SIP1680
      BH=B-W*CH                                                         SIP1690
      DH=D-W*GH                                                         SIP1700
      EH=E+W*CH+W*GH                                                    SIP1710
      FH=F-W*CH                                                         SIP1720
      HH=H-W*GH                                                         SIP1730
      ALFA=BH                                                           SIP1740
      BETA=DH                                                           SIP1750
      GAMA=EH-ALFA*ETA(NA)-BETA*DEL(NL)                                 SIP1760
```

```
      DEL(N)=FH/GAMA
      ETA(N)=HH/GAMA
      RES=-D*PHI(NL)-F*PHI(NR)-H*PHI(NB)-B*PHI(NA)-E*PHI(N)-RHO*KEEP(N)-
     1SL(N)-QRE(N)-WELL(N)+ETQD-SUBS-TL(N)*STRT(N)
      V(N)=(HMAX*RES-ALFA*V(NA)-BETA*V(NL))/GAMA
  210 CONTINUE
C
C     ---BACK SUBSTITUTE FOR VECTOR XI---
      DO 220 I=1,I2
      I3=DIML-I
      DO 220 J=1,J2
      J3=DIMW-J
      N=I3+DIML*(J3-1)
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 220
      XI(N)=V(N)-DEL(N)*XI(N+DIML)-ETA(N)*XI(N+1)
C
C     ---COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERION---
      TCHK=DABS(XI(N))                                                  $$4-50
      IF(TCHK.GT.BIGI) BIGI=TCHK
      PHI(N)=PHI(N)+XI(N)
  220 CONTINUE
      IF (BIGI.GT.ERR) TEST=1.
      TEST3(KOUNT+1)=BIGI
      IF (TEST.EQ.1.) GO TO 40
      RETURN
C
C     ..................................................................
C     ---ORDER EQUATIONS WITH THE LAST ROW FIRST - 3X3  EXAMPLE:
C               7 8 9
C               4 5 6
C               1 2 3
C     ..................................................................
  230 DO 340 II=1,I2
      I=DIML-II
      DO 340 J=2,JNO1
      N=I+DIML*(J-1)
      NL=N-DIML
      NR=N+DIML
      NA=N-1
      NB=N+1
C
C     ---SKIP COMPUTATIONS IF NODE IS OUTSIDE AQUIFER BOUNDARY---
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 340
C
C     ---COMPUTE COEFFICIENTS---
      D=TR(NL)/DELX(J)
      F=TR(N)/DELX(J)
      B=TC(NA)/DELY(I)
      H=TC(N)/DELY(I)
      IF (EVAP.NE.CHK(6)) GO TO 250
C
C     ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---
      ETQB=0.
      ETQD=0.0
```

```
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 250
      IF (PHE(N).GT.GRND(N)) GO TO 240
      ETQB=QET/ETDIST
      ETQD=ETQB*(ETDIST-GRND(N))
      GO TO 250
  240 ETQD=QET
C
C     ---COMPUTE STORAGE TERM---
  250 IF (CONVRT.EQ.CHK(7)) GO TO 260
      RHO=S(N)/DELT
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT
      GO TO 330
C
C     ---COMPUTE STORAGE COEFFICIENT FOR CONVERSION PROBLEM---
  260 SUBS=0.0
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 300
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 290
      IF (KEEP(N)-PHE(N)) 270,280,280
  270 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))
      GO TO 300
  280 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))
  290 RHO=SY(N)/DELT
      GO TO 310
  300 RHO=S(N)/DELT
  310 IF (LEAK.NE.CHK(9)) GO TO 330
C
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 330
      HED1=DMAX1(STRT(N),TOP(N))
      U=1.
      HED2=0.
      IF (PHE(N).GE.TOP(N)) GO TO 320
      HED2=TOP(N)
      U=0.
  320 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))
  330 CONTINUE
C
C     ---SIP 'REVERSE' ALGORITHM---
C     ---FORWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR V---
      E=-B-D-F-H-RHO-TL(N)*U-ETQB
      CH=DEL(NB)*H/(1.+W*DEL(NB))
      GH=ETA(NL)*D/(1.+W*ETA(NL))
      BH=H-W*CH
      DH=D-W*GH
      EH=E+W*CH+W*GH
      FH=F-W*CH
      HH=B-W*GH
      ALFA=BH
      BETA=DH
      GAMA=EH-ALFA*ETA(NB)-BETA*DEL(NL)
      DEL(N)=FH/GAMA
      ETA(N)=HH/GAMA
      RES=-D*PHI(NL)-F*PHI(NR)-H*PHI(NB)-B*PHI(NA)-E*PHI(N)-RHO*KEEP(N)-
     1SL(N)-QRE(N)-WELL(N)+ETQD-SUBS-TL(N)*STRT(N)
```

```
      V(N)=(HMAX*RES-ALFA*V(NB)-BETA*V(NL))/GAMA                        SIP2850
  340 CONTINUE                                                          SIP2860
C                                                                       SIP2870
C     ---BACK SUBSTITUTE FOR VECTOR XI---                               SIP2880
      DO 350 I3=2,INO1                                                  SIP2890
      DO 350 J=1,J2                                                     SIP2900
      J3=DIMW-J                                                         SIP2910
      N=I3+DIML*(J3-1)                                                  SIP2920
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 350                           SIP2930
      XI(N)=V(N)-DEL(N)*XI(N+DIML)-ETA(N)*XI(N-1)                       SIP2940
C                                                                       SIP2950
C     ---COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERION---          SIP2960
      TCHK=DABS(XI(N))                                                  $$4-70
      IF(TCHK.GT.BIGI) BIGI=TCHK
      PHI(N)=PHI(N)+XI(N)
  350 CONTINUE                                                          SIP3000
      IF (BIGI.GT.ERR) TEST=1.                                          SIP3010
      TEST3(KOUNT+1)=BIGI                                               SIP3020
      IF (TEST.EQ.1.) GO TO 40                                          SIP3030
      RETURN                                                            SIP3040
C                                                                       SIP3050
C     ..................................................................SIP3060
C                                                                       SIP3070
C     ---FORMATS---                                                     SIP3080
C                                                                       SIP3090
C     ------------------------------------------------------------------SIP3100
C                                                                       SIP3110
C                                                                       SIP3120
  360 FORMAT ('0EXCEEDED PERMITTED NUMBER OF ITERATIONS'/' ',39('*'))   SIP3130
  370 FORMAT ('-',44X,'SOLUTION BY THE STRONGLY IMPLICIT PROCEDURE'/45X,SIP3140
     143('_'),//,61X,'BETA=',F5.2)                                      SIP3150
  380 FORMAT (1H0,I5,22H ITERATION PARAMETERS:,6D15.7/(/28X,6D15.7/))   SIP3160
      END                                                               SIP3170-
      SUBROUTINE SOLVE2(PHI,BE,G,TEMP,KEEP,PHE,STRT,T,S,QRE,WELL,TL,SL,DSOR   10
     1EL,ETA,V,XI,DELX,BETA,DELY,ALFA,TEST3,TR,TC,GRND,SY,TOP,RATE,M,RIVSOR   20
     2ER,SURI,PERM,BOTTOM,DDN,WR,NWR,NENT)
C     ------------------------------------------------------------------SOR   40
C     SOLUTION BY LINE SUCCESSIVE OVERRELAXATION                        SOR   50
C     ------------------------------------------------------------------SOR   60
C                                                                       SOR   70
C     SPECIFICATIONS:                                                   SOR   80
      IMPLICIT REAL *8 (A-H,O-Z)                                        $5-10
      REAL *8 IMK,KEEP,M                                                $$5-20
      INTEGER R,P,PU,DIML,DIMW
      INTEGER CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD
     1,CONTR,LEAK,RECH,SIP,ADI,NUMS,IDK1,IDK2
C                                                                       SOR  140
      DIMENSION PHI(1),BE(1),G(1),TEMP(1),KEEP(1),PHE(1),STRT(1)
     1,T(1),S(1),QRE(1),WELL(1),TL(1),SL(1),DEL(1),ETA(1),
     2V(1),XI(1),DELX(1),BETA(1),DELY(1),ALFA(1),TEST3(1),
     3TR(1),TC(1),GRND(1),SY(1),TOP(1),RATE(1),M(1),RIVER(1)
      DIMENSION RHOP(20)                                                $5-30
C                                                                       SOR  190
      COMMON /SARRAY/ VF4(11),CHK(15)                                   SOR  200
```

```
      COMMON /SPARAMI/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,LEAK,
     1RECH,SIP,NUMS,LSOR,ADI,IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,
     2ITMAX,LENGTH,NWEL,NW,DIML,DIMW,I,J,IDK1,IDK2,JNO1,INO1,R,P,PU
      COMMON /SPARAMR/ U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,
     1WIDTH,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,EROR
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1
      GO TO (1,2) NENT
      RETURN
C     ..................................................................
C
C     ---WRITE ACCELERATION PARAMETER---
C     *********************
C     ENTRY ITER2(PHI,BE,G,TEMP,KEEP,PHE,STRT,T,S,QRE,WELL,TL,SL,
C    1DEL,ETA,V,XI,DELX,BETA,DELY,ALFA,TEST3,TR,TC,GRND,SY,TOP,RATE,M,
C    2RIVER)
C     *********************
1     CONTINUE
C     WRITE (P,490)
C     WRITE (P,500) HMAX,LENGTH
      RETURN
C     ..................................................................
C
C     ---INITIALIZE DATA FOR A NEW ITERATION---
   10 KOUNT=KOUNT+1
      IF (KOUNT.LE.ITMAX) GO TO 20
      WRITE (P,510)
      CALL STEP(PHI,KEEP,STRT,SURI,T,WELL,PERM,BOTTOM,TOP,DELX,DDN
     1,DELY,WR,NWR,TEST3,TR,TC,S,QRE,TL,SY,RATE,RIVER,M,GRND,3)
C     ************************
C     ENTRY NEWITB(PHI,BE,G,TEMP,KEEP,PHE,STRT,T,S,QRE,WELL,TL,SL,
C    1DEL,ETA,V,XI,DELX,BETA,DELY,ALFA,TEST3,TR,TC,GRND,SY,TOP,RATE,M,
C    2RIVER,SURI,PERM,BOTTOM,DDN,WR,NWR)
C     ************************
2     CONTINUE
   20 TEST3(KOUNT+1)=0.
      TEST=0.
      N=DIML*DIMW
      DO 30 I=1,N
   30 PHE(I)=PHI(I)
      BIGI=0.0
C
C     ---COMPUTE TRANSMISSIVITY AND T COEFFICIENTS IN WATER TABLE
C     OR WATER TABLE-ARTESIAN SIMUATION---
      IF (WATER.NE.CHK(2)) GO TO 40
      CALL COEF(PHI,KEEP,PHE,STRT,SURI,T,TR,TC,S,WELL,TL,SL,PERM,BOTTOM
     1,SY,RATE,RIVER,M,TOP,GRND,DELX,DELY,DDN,WR,NWR,2)
C     ..................................................................
C
C     ---SOLUTION BY LSOR---
C     --------------------
   40 NO3=DIMW-2
      TEMP(DIMW)=0.0
      DO 170 I=2,INO1
      DO 150 J=2,JNO1
```

```
      N=I+DIML*(J-1)
      NA=N-1
      NB=N+1
      NL=N-DIML
      NR=N+DIML
      BE(J)=0.0
      G(J)=0.0
C
C     ---SKIP COMPUTATIONS IF NODE IS OUTSIDE AQUIFER BOUNDARY---
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 150
C
C     ---COMPUTE COEFFICIENTS---
      D=TR(N-DIML)/DELX(J)
      F=TR(N)/DELX(J)
      B=TC(N-1)/DELY(I)
      H=TC(N)/DELY(I)
      IF (EVAP.NE.CHK(6)) GO TO 60
C
C     ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---
      ETQB=0.
      ETQD=0.0
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 60
      IF (PHE(N).GT.GRND(N)) GO TO 50
      ETQB=QET/ETDIST
      ETQD=ETQB*(ETDIST-GRND(N))
      GO TO 60
   50 ETQD=QET
C
C     ---COMPUTE STORAGE TERM---
   60 IF (CONVRT.EQ.CHK(7)) GO TO 70
      RHO=S(N)/DELT
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT
      GO TO 140
C
C     ---COMPUTE STORAGE COEFFICIENT FOR CONVERSION PROBLEM---
   70 SUBS=0.0
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 110
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 100
      IF (KEEP(N)-PHE(N)) 80,90,90
   80 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))
      GO TO 110
   90 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))
  100 RHO=SY(N)/DELT
      GO TO 120
  110 RHO=S(N)/DELT
  120 IF (LEAK.NE.CHK(9)) GO TO 140
C
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 140
      HED1=DMAX1(STRT(N),TOP(N))
      U=1.
      HED2=0.
      IF (PHE(N).GE.TOP(N)) GO TO 130
      HED2=TOP(N)
```

```
      U=0.
  130 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))
  140 CONTINUE
C
C     ---FORWARD SUBSTITUTE, COMPUTING INTERMEDIATE VECTOR G---
      E=-D-F-B-H-RHO-TL(N)*U-ETQB
      W=E-D*BE(J-1)
      BE(J)=F/W
      Q=-B*PHI(NA)-H*PHI(NB)-RHO*KEEP(N)-SL(N)-QRE(N)-WELL(N)+ETQD-SUBS-
     1TL(N)*STRT(N)-D*PHI(NL)-F*PHI(NR)-E*PHI(N)
      G(J)=(Q-D*G(J-1))/W
  150 CONTINUE
C
C     ---BACK SUBSTITUTE FOR TEMP---
      DO 160 KNO4=1,NO3
      NO4=DIMW-KNO4
      TEMP(NO4)=G(NO4)-BE(NO4)*TEMP(NO4+1)
  160 CONTINUE
C
C     ---EXTRAPOLATED VALUE OF PHI---
      DO 170 J=2,JNO1
      N=I+DIML*(J-1)
      PHI(N)=PHI(N)+HMAX*TEMP(J)
C
C     ---COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERION---
      TCHK=DABS(TEMP(J))
      IF(TCHK.GT.BIGI) BIGI=TCHK
  170 CONTINUE
      IF (BIGI.GT.ERR) TEST=1.
      TEST3(KOUNT+1)=BIGI
      IF (KOUNT.EQ.0) GO TO 10
      IF (TEST.EQ.0.) RETURN
C
C     ---TEST FOR TWO DIMENSIONAL CORRECTION---
      IF (MOD(KOUNT,LENGTH).NE.0) GO TO 10
      GO TO 200
  180 DO 190 I=2,INO1
      DO 190 J=2,JNO1
      N=I+DIML*(J-1)
      IF (T(N).EQ.0.) GO TO 190
      PHI(N)=PHI(N)+ALFA(I)+BETA(J)
  190 CONTINUE
      GO TO 10
C     ..................................................................
C
C     ---TWO DIMENSIONAL CORRECTION TO LSOR---
C     ----------------------------------------
C
C     ---COMPUTE ALFA CORRECTION FOR ROWS---
  200 DO 210 I=1,DIML
      ALFA(I)=0.
      BE(I)=0.0
  210 G(I)=0.0
      DO 330 I=2,INO1
```

```
      A=0.                                                              SOR1730
      B2=0.                                                             SOR1740
      C=0.                                                              SOR1750
      Q=0.                                                              SOR1760
C                                                                       SOR1770
C     ---SUMMATION OF COEFFICIENTS FOR EACH ROW---                      SOR1780
      DO 320 J=2,JNO1                                                   SOR1790
      N=I+DIML*(J-1)                                                    SOR1800
      NA=N-1                                                            SOR1810
      NB=N+1                                                            SOR1820
      NL=N-DIML                                                         SOR1830
      NR=N+DIML                                                         SOR1840
      IF (S(N).LT.0.) GO TO 330                                         SOR1850
      IF (T(N).EQ.0.) GO TO 320                                         SOR1860
C                                                                       SOR1870
C     ---COMPUTE COEFFICIENTS---                                        SOR1880
      D=TR(N-DIML)/DELX(J)                                              SOR1890
      F=TR(N)/DELX(J)                                                   SOR1900
      B=TC(N-1)/DELY(I)                                                 SOR1910
      H=TC(N)/DELY(I)                                                   SOR1920
      IF (EVAP.NE.CHK(6)) GO TO 230                                     SOR1930
C                                                                       SOR1940
C     ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---              SOR1950
      ETQB=0.                                                           SOR1960
      ETQD=0.0                                                          SOR1970
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 230                           SOR1980
      IF (PHE(N).GT.GRND(N)) GO TO 220                                  SOR1990
      ETQB=QET/ETDIST                                                   SOR2000
      ETQD=ETQB*(ETDIST-GRND(N))                                        SOR2010
      GO TO 230                                                         SOR2020
  220 ETQD=QET                                                          SOR2030
C                                                                       SOR2040
C     ---COMPUTE STORAGE TERM---                                        SOR2050
  230 IF (CONVRT.EQ.CHK(7)) GO TO 240                                   SOR2060
      RHO=S(N)/DELT                                                     SOR2070
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT                               SOR2080
      GO TO 310                                                         SOR2090
  240 SUBS=0.0                                                          SOR2100
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 280             SOR2110
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 270             SOR2120
      IF (KEEP(N)-PHE(N)) 250,260,260                                   SOR2130
  250 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))                           SOR2140
      GO TO 280                                                         SOR2150
  260 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))                           SOR2160
  270 RHO=SY(N)/DELT                                                    SOR2170
      GO TO 290                                                         SOR2180
  280 RHO=S(N)/DELT                                                     SOR2190
  290 IF (LEAK.NE.CHK(9)) GO TO 310                                     SOR2200
C                                                                       SOR2210
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---          SOR2220
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 310                        SOR2230
      HED1=DMAX1(STRT(N),TOP(N))                                        $$5-50
      U=1.                                                              SOR2250
      HED2=0.                                                           SOR2260
```

```
      IF (PHE(N).GE.TOP(N)) GO TO 300
      HED2=TOP(N)
      U=0.
  300 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))
  310 CONTINUE
C
      A=A-B
      B1=B+H+RHO+TL(N)*U+ETQB
      B2=B2+B1
      C=C-H
      Q=Q+(D*PHI(NL)+F*PHI(NR)+B*PHI(NA)+H*PHI(NB)+RHO*KEEP(N)+SL(N)+QRES
     1(N)+WELL(N)-ETQD+SUBS+TL(N)*STRT(N)-(D+F+B1)*PHI(N))
  320 CONTINUE
C
C     ---COMPUTATION OF INTERMEDIATE VECTOR G---
      W=B2-A*BE(I-1)
      BE(I)=C/W
      G(I)=(Q-A*G(I-1))/W
  330 CONTINUE
C
C     ---BACK SUBSTITUTE FOR ALFA---
      NO3=DIML-2
      DO 340 KNO4=1,NO3
      NO4=DIML-KNO4
  340 ALFA(NO4)=G(NO4)-BE(NO4)*ALFA(NO4+1)
C     *****************************************************************
C
C     ---COMPUTE BETA CORRECTION FOR COLUMNS---
      DO 350 J=1,DIMW
      BETA(J)=0.
      BE(J)=0.0
  350 G(J)=0.0
      DO 470 J=2,JNO1
      A=0.
      B2=0.
      C=0.
      Q=0.
C
C     ---SUMMATION OF COEFFICIENTS FOR EACH COLUMN---
      DO 460 I=2,INO1
      N=I+DIML*(J-1)
      NA=N-1
      NB=N+1
      NL=N-DIML
      NR=N+DIML
      IF (S(N).LT.0.) GO TO 470
      IF (T(N).EQ.0.) GO TO 460
      D=TR(N-DIML)/DELX(J)
      F=TR(N)/DELX(J)
      B=TC(N-1)/DELY(I)
      H=TC(N)/DELY(I)
      IF (EVAP.NE.CHK(6)) GO TO 370
C
C     ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---
```

```
      ETQB=0.
      ETQD=0.0
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 370
      IF (PHE(N).GT.GRND(N)) GO TO 360
      ETQB=QET/ETDIST
      ETQD=ETQB*(ETDIST-GRND(N))
      GO TO 370
  360 ETQD=QET
C
C     ---COMPUTE STORAGE TERM---
  370 IF (CONVRT.EQ.CHK(7)) GO TO 380
      RHO=S(N)/DELT
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT
      GO TO 450
C
C     ---COMPUTE STORAGE COEFFICIENT FOR CONVERSION PROBLEM---
  380 SUBS=0.0
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 420
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 410
      IF (KEEP(N)-PHE(N)) 390,400,400
  390 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))
      GO TO 420
  400 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))
  410 RHO=SY(N)/DELT
      GO TO 430
  420 RHO=S(N)/DELT
  430 IF (LEAK.NE.CHK(9)) GO TO 450
C
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 450
      HED1=DMAX1(STRT(N),TOP(N))
      U=1.
      HED2=0.
      IF (PHE(N).GE.TOP(N)) GO TO 440
      HED2=TOP(N)
      U=0.
  440 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))
  450 CONTINUE
C
      A=A-D
      B1=D+F+RHO+TL(N)*U+ETQB
      B2=B2+B1
      C=C-F
      Q=Q+(D*PHI(NL)+F*PHI(NR)+B*PHI(NA)+H*PHI(NB)+RHO*KEEP(N)+SL(N)+QRE
     1(N)+WELL(N)-ETQD+SUBS+TL(N)*STRT(N)-(B+H+B1)*PHI(N))
  460 CONTINUE
C
C     ---COMPUTATION OF INTERMEDIATE VECTOR G---
      W=B2-A*BE(J-1)
      BE(J)=C/W
      G(J)=(Q-A*G(J-1))/W
  470 CONTINUE
C
C     ---BACK SUBSTITUTE FOR BETA---
```

```
      NO3=DIMW-2                                                        SOR3350
      DO 480 KNO4=1,NO3                                                 SOR3360
      NO4=DIMW-KNO4                                                     SOR3370
  480 BETA(NO4)=G(NO4)-BE(NO4)*BETA(NO4+1)                              SOR3380
      GO TO 180                                                         SOR3390
C     ..................................................................SOR3400
C                                                                       SOR3410
C     ---FORMATS---                                                     SOR3420
C                                                                       SOR3430
C     ------------------------------------------------------------------SOR3440
C                                                                       SOR3450
C                                                                       SOR3460
  490 FORMAT ('-',45X,'SOLUTION BY LINE SUCCESSIVE OVERRELAXATION'/46X,4SOR3470
     12('_'))                                                           SOR3480
  500 FORMAT ('-',26X,'ACCELERATION PARAMETER =',F6.3,'  TWO DIMENSIONALSOR3490
     1 CORRECTION EVERY',I5,' ITERATIONS')                              SOR3500
  510 FORMAT ('0EXCEEDED PERMITTED NUMBER OF ITERATIONS'/' ',39('*'))   SOR3510
      END                                                               SOR3520-
      SUBROUTINE SOLVE3(PHI,BE,G,TEMP,KEEP,PHE,STRT,T,S,QRE,WELL,TL,SL,DADI  10
     1EL,ETA,V,XI,DELX,BETA,DELY,ALFA,XII,TEST3,TR,TC,GRND,SY,TOP,RATE,MADI  20
     2,RIVER,SURI,PERM,BOTTOM,DDN,WR,NWR,NENT)
C     ------------------------------------------------------------------
C     SOLUTION BY THE ALTERNATING DIRECTION IMPLICIT METHOD
C     ------------------------------------------------------------------ADI  60
C                                                                       ADI  70
C     SPECIFICATIONS:                                                   ADI  80
      IMPLICIT REAL *8 (A-H,O-Z)                                        $6-10
      REAL *8 IMK,KEEP,M                                                $$6-20
      INTEGER R,P,PU,DIML,DIMW
      INTEGER CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD
     1,CONTR,LEAK,RECH,SIP,ADI,NUMS,IDK1,IDK2
C                                                                       ADI 140
      DIMENSION PHI(1),BE(1),G(1),TEMP(1),KEEP(1),PHE(1),STRT(1)
     1,T(1),S(1),QRE(1),WELL(1),TL(1),SL(1),DEL(1),ETA(1),V(1)
     2,XI(1),DELX(1),BETA(1),DELY(1),ALFA(1),XII(1),TEST3(1),
     3TR(1),TC(1),GRND(1),SY(1),TOP(1),RATE(1),M(1),RIVER(1)
      DIMENSION RHOP(20)                                                $6-30
C                                                                       ADI 190
      COMMON /SARRAY/ VF4(11),CHK(15)                                   ADI 200
      COMMON /SPARAMI/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,LEAK,
     1RECH,SIP,NUMS,LSOR,ADI,IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,
     2ITMAX,LENGTH,NWEL,NW,DIML,DIMW,I,J,IDK1,IDK2,JNO1,INO1,R,P,PU
      COMMON /SPARAMR/ U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,
     1WIDTH,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,EROR
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1
      GO TO (1,2) NENT
      RETURN                                                            ADI 260
C     ..................................................................ADI 270
C                                                                       ADI 280
C     ---COMPUTE AND PRINT ITERATION PARAMETERS---                      ADI 290
C     ********************                                              ADI 300
C     ENTRY ITER3(PHI,BE,G,TEMP,KEEP,PHE,STRT,T,S,QRE,WELL,TL,SL,
C    1DEL,ETA,V,XI,DELX,BETA,DELY,ALFA,XII,TEST3,TR,TC,GRND,SY,TOP,
C    2RATE,M,RIVER)
```

```
C     *******************                                           ADI 320
1     CONTINUE
      HMIN=2.                                                       ADI 330
      IN4=DIMW-2                                                    ADI 340
      IN5=DIML-2                                                    ADI 350
      XVAL=3.1415**2/(2.*IN4*IN4)                                   ADI 360
      YVAL=3.1415**2/(2.*IN5*IN5)                                   ADI 370
      DO 10 I=2,INO1                                                ADI 380
      DO 10 J=2,JNO1                                                ADI 390
      N=I+DIML*(J-1)                                                ADI 400
      IF (T(N).EQ.0.) GO TO 10                                      ADI 410
      XPART=XVAL*(1/(1+DELX(J)**2*FACTY/DELY(I)**2*FACTX))          ADI 420
      YPART=YVAL*(1/(1+DELY(I)**2*FACTX/DELX(J)**2*FACTY))          ADI 430
      HMIN=DMIN1(HMIN,XPART,YPART)                                  $$6-40
   10 CONTINUE                                                      ADI 450
      ALPHA=DEXP(DLOG(HMAX/HMIN)/(LENGTH-1))                        $$6-50
      RHOP(1)=HMIN                                                  ADI 470
      DO 20 NTIME=2,LENGTH                                          ADI 480
   20 RHOP(NTIME)=RHOP(NTIME-1)*ALPHA                               ADI 490
C     WRITE (P,400)
C     WRITE (P,410) LENGTH, (RHOP(J),J=1,LENGTH)
      RETURN                                                        ADI 520
C     ..............................................................ADI 530
C                                                                   ADI 540
C     ---INITIALIZE DATA FOR A NEW ITERATION---                     ADI 550
   30 KOUNT=KOUNT+1                                                 ADI 560
      IF (KOUNT.LE.ITMAX) GO TO 40                                  ADI 570
      WRITE (P,390)                                                 ADI 580
      CALL STEP(PHI,KEEP,STRT,SURI,T,WELL,PERM,BOTTOM,TOP,DELX,DDN
     1,DELY,WR,NWR,TEST3,TR,TC,S,QRE,TL,SY,RATE,RIVER,M,GRND,3)
   40 IF (MOD(KOUNT,LENGTH)) 50,50,60                               ADI 600
C     ***********************                                       ADI 610
C     ENTRY NEWITC(PHI,BE,G,TEMP,KEEP,PHE,STRT,T,S,QRE,WELL,TL,SL,
C    1DEL,ETA,V,XI,DELX,BETA,DELY,ALFA,XII,TEST3,TR,TC,GRND,SY,TOP,
C    2RATE,M,RIVER,SURI,PERM,BOTTOM,DDN,WR,NWR)
C     ***********************                                       ADI 630
2     CONTINUE
   50 NTH=0                                                         ADI 640
   60 NTH=NTH+1                                                     ADI 650
      PARAM=RHOP(NTH)                                               ADI 660
      TEST3(KOUNT+1)=0.                                             ADI 670
      TEST=0.                                                       ADI 680
      N=DIML*DIMW                                                   ADI 690
      DO 70 I=1,N                                                   ADI 700
   70 PHE(I)=PHI(I)                                                 ADI 710
      BIGI=0.0                                                      ADI 720
C                                                                   ADI 730
C     ---COMPUTE TRANSMISSIVITY AND T COEFFICIENTS IN WATER TABLE   ADI 740
C     OR WATER TABLE-ARTESIAN SIMUATION---                          ADI 750
      IF (WATER.NE.CHK(2)) GO TO 80                                 ADI 760
      CALL COEF(PHI,KEEP,PHE,STRT,SURI,T,TR,TC,S,WELL,TL,SL,PERM,BOTTOM
     1,SY,RATE,RIVER,M,TOP,GRND,DELX,DELY,DDN,WR,NWR,2)
C     ..............................................................ADI 780
C                                                                   ADI 790
```

```
C      ---SOLUTION BY ADI---
C      --------------------
C      ---COMPUTE IMPLICITLY ALONG ROWS---
   80 NO3=DIMW-2
      DO 90 J=1,DIMW
      N=1+DIML*(J-1)
   90 TEMP(J)=PHI(N)
      DO 230 I=2,DIML
      DO 200 J=2,JNO1
      N=I+DIML*(J-1)
      NA=N-1
      NB=N+1
      NL=N-DIML
      NR=N+DIML
      BE(J)=0.0
      G(J)=0.0
C
C      ---SKIP COMPUTATIONS IF NODE IS OUTSIDE AQUIFER BOUNDARY---
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 200
C
C      ---COMPUTE COEFFICIENTS---
      D=TR(N-DIML)/DELX(J)
      F=TR(N)/DELX(J)
      B=TC(N-1)/DELY(I)
      H=TC(N)/DELY(I)
      IF (EVAP.NE.CHK(6)) GO TO 110
C
C      ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---
      ETQB=0.
      ETQD=0.0
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 110
      IF(PHE(N).GT.GRND(N))GO TO 100
      ETQB=QET/ETDIST
      ETQD=ETQB*(ETDIST-GRND(N))
      GO TO 110
  100 ETQD=QET
C
C      ---COMPUTE STORAGE TERM---
  110 IF (CONVRT.EQ.CHK(7)) GO TO 120
      RHO=S(N)/DELT
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT
      GO TO 190
C
C      ---COMPUTE STORAGE COEFFICIENT FOR CONVERSION PROBLEM---
  120 SUBS=0.0
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 160
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 150
      IF (KEEP(N)-PHE(N)) 130,140,140
  130 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))
      GO TO 160
  140 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))
  150 RHO=SY(N)/DELT
      GO TO 170
  160 RHO=S(N)/DELT
```

```
  170 IF (LEAK.NE.CHK(9)) GO TO 190
C
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 190
      HED1=DMAX1(STRT(N),TOP(N))
      U=1.
      HED2=0.
      IF (PHE(N).GE.TOP(N)) GO TO 180
      HED2=TOP(N)
      U=0.
  180 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))
  190 CONTINUE
C
C     ---CALCULATE VALUES FOR PARAMETERS USED IN THOMAS ALGORITHM
C     AND FORWARD SUBSTITUTE TO COMPUTE INTERMEDIATE VECTOR G---
      IMK=(B+D+F+H)*PARAM
      E=-D-F-RHO-IMK-TL(N)*U-ETQB
      W=E-D*BE(J-1)
      BE(J)=F/W
      Q=-B*PHI(NA)+(B+H-IMK-E)*PHI(N)-H*PHI(NB)-RHO*KEEP(N)-SL(N)-QRE(N)
     1-WELL(N)+ETQD-SUBS-TL(N)*STRT(N)-D*PHI(NL)-F*PHI(NR)
      G(J)=(Q-D*G(J-1))/W
  200 CONTINUE
C
C     ---BACK SUBSTITUTE FOR HEAD VALUES AND PLACE THEM IN TEMP---
      XII(DIMW)=0.D0
      DO 220 KNO4=1,NO3
      NO4=DIMW-KNO4
      N=I+DIML*(NO4-1)
C
C     ---FIRST PLACE TEMP VALUES IN PHI(N-1)---
      PHI(N-1)=TEMP(NO4)
      IF (T(N).NE.0..AND.S(N).GE.0.) GO TO 210
      XII(NO4)=0.D0
      GO TO 220
  210 XII(NO4)=G(NO4)-BE(NO4)*XII(NO4+1)
  220 TEMP(NO4)=PHI(N)+XII(NO4)
  230 CONTINUE
C     ..................................................................
C
C     ---COMPUTE IMPLICITLY ALONG COLUMNS---
      NO3=DIML-2
      DO 240 I=1,DIML
  240 TEMP(I)=PHI(I)
      DO 380 J=2,DIMW
      DO 350 I=2,INO1
      N=I+DIML*(J-1)
      NA=N-1
      NB=N+1
      NL=N-DIML
      NR=N+DIML
      BE(I)=0.0
      G(I)=0.0
C
```

```
C     ---SKIP COMPUTATIONS IF NODE IS OUTSIDE AQUIFER BOUNDARY---       ADI1880
      IF (T(N).EQ.0..OR.S(N).LT.0.) GO TO 350                           ADI1890
C                                                                       ADI1900
C     ---COMPUTE COEFFICIENTS---                                        ADI1910
      D=TR(N-DIML)/DELX(J)                                              ADI1920
      F=TR(N)/DELX(J)                                                   ADI1930
      B=TC(N-1)/DELY(I)                                                 ADI1940
      H=TC(N)/DELY(I)                                                   ADI1950
      IF (EVAP.NE.CHK(6)) GO TO 260                                     ADI1960
C                                                                       ADI1970
C     ---COMPUTE EXPLICIT AND IMPLICIT PARTS OF ET RATE---              ADI1980
      ETQB=0.                                                           ADI1990
      ETQD=0.0                                                          ADI2000
      IF (PHE(N).LE.GRND(N)-ETDIST) GO TO 260                           ADI2010
      IF (PHE(N).GT.GRND(N)) GO TO 250                                  ADI2020
      ETQB=QET/ETDIST                                                   ADI2030
      ETQD=ETQB*(ETDIST-GRND(N))                                        ADI2040
      GO TO 260                                                         ADI2050
  250 ETQD=QET                                                          ADI2060
C                                                                       ADI2070
C     ---COMPUTE STORAGE TERM---                                        ADI2080
  260 IF (CONVRT.EQ.CHK(7)) GO TO 270                                   ADI2090
      RHO=S(N)/DELT                                                     ADI2100
      IF (WATER.EQ.CHK(2)) RHO=SY(N)/DELT                               ADI2110
      GO TO 340                                                         ADI2120
C                                                                       ADI2130
C     ---COMPUTE STORAGE COEFFICIENT FOR CONVERSION PROBLEM---          ADI2140
  270 SUBS=0.0                                                          ADI2150
      IF (KEEP(N).GE.TOP(N).AND.PHE(N).GE.TOP(N)) GO TO 310             ADI2160
      IF (KEEP(N).LT.TOP(N).AND.PHE(N).LT.TOP(N)) GO TO 300             ADI2170
      IF (KEEP(N)-PHE(N)) 280,290,290                                   ADI2180
  280 SUBS=(SY(N)-S(N))/DELT*(KEEP(N)-TOP(N))                           ADI2190
      GO TO 310                                                         ADI2200
  290 SUBS=(S(N)-SY(N))/DELT*(KEEP(N)-TOP(N))                           ADI2210
  300 RHO=SY(N)/DELT                                                    ADI2220
      GO TO 320                                                         ADI2230
  310 RHO=S(N)/DELT                                                     ADI2240
  320 IF (LEAK.NE.CHK(9)) GO TO 340                                     ADI2250
C                                                                       ADI2260
C     ---COMPUTE NET LEAKAGE TERM FOR CONVERSION SIMULATION---          ADI2270
      IF (RATE(N).EQ.0..OR.M(N).EQ.0.) GO TO 340                        ADI2280
      HED1=DMAX1(STRT(N),TOP(N))                                        $$6-70
      U=1.                                                              ADI2300
      HED2=0.                                                           ADI2310
      IF (PHE(N).GE.TOP(N)) GO TO 330                                   ADI2320
      HED2=TOP(N)                                                       ADI2330
      U=0.                                                              ADI2340
  330 SL(N)=RATE(N)/M(N)*(RIVER(N)-HED1)+TL(N)*(HED1-HED2-STRT(N))      ADI2350
  340 CONTINUE                                                          ADI2360
C                                                                       ADI2370
C     ---CALCULATE VALUES FOR PARAMETERS USED IN THOMAS ALGORITHM       ADI2380
C     AND FORWARD SUBSTITUTE TO COMPUTE INTERMEDIATE VECTOR G---        ADI2390
      IMK=(B+D+F+H)*PARAM                                               ADI2400
      E=-B-H-RHO-IMK-TL(N)*U-ETQB                                       ADI2410
```

```
      W=E-B*BE(I-1)                                                     ADI2420
      BE(I)=H/W                                                         ADI2430
      Q=-D*PHI(NL)+(D+F-IMK-E)*PHI(N)-F*PHI(NR)-RHO*KEEP(N)-SL(N)-QRE(N)ADI2440
     1-WELL(N)+ETQD-SUBS-TL(N)*STRT(N)-B*PHI(NA)-H*PHI(NB)              ADI2450
      G(I)=(Q-B*G(I-1))/W                                               ADI2460
  350 CONTINUE                                                          ADI2470
C                                                                       ADI2480
C     ---BACK SUBSTITUTE FOR HEAD VALUES AND PLACE THEM IN TEMP---      ADI2490
      XII(DIML)=0.D0                                                    ADI2500
      DO 370 KNO4=1,NO3                                                 ADI2510
      NO4=DIML-KNO4                                                     ADI2520
      N=NO4+DIML*(J-1)                                                  ADI2530
C                                                                       ADI2540
C     ---FIRST PLACE TEMP VALUES IN PHI(N-DIML)----                     ADI2550
      PHI(N-DIML)=TEMP(NO4)                                             ADI2560
      IF (T(N).NE.0..AND.S(N).GE.0.) GO TO 360                          ADI2570
      XII(NO4)=0.D0                                                     ADI2580
      TEMP(NO4)=PHI(N)                                                  ADI2590
      GO TO 370                                                         ADI2600
  360 XII(NO4)=G(NO4)-BE(NO4)*XII(NO4+1)                                ADI2610
      TEMP(NO4)=PHI(N)+XII(NO4)                                         ADI2620
C                                                                       ADI2630
C     ---COMPARE MAGNITUDE OF CHANGE WITH CLOSURE CRITERION---          ADI2640
      TCHK=DABS(TEMP(NO4)-PHE(N))                                       $$6-80
      IF(TCHK.GT.BIGI) BIGI=TCHK
  370 CONTINUE                                                          ADI2670
  380 CONTINUE                                                          ADI2680
      IF (BIGI.GT.ERR) TEST=1.                                          ADI2690
      TEST3(KOUNT+1)=BIGI                                               ADI2700
      IF (TEST.EQ.1.) GO TO 30                                          ADI2710
      RETURN                                                            ADI2720
C     ..................................................................ADI2730
C                                                                       ADI2740
C     ---FORMATS---                                                     ADI2750
C                                                                       ADI2760
C     ------------------------------------------------------------------ADI2770
C                                                                       ADI2780
C                                                                       ADI2790
  390 FORMAT ('0EXCEEDED PERMITTED NUMBER OF ITERATIONS'/' ',39('*'))   ADI2800
  400 FORMAT ('-',38X,'SOLUTION BY THE ALTERNATING DIRECTION IMPLICIT PRADI2810
     1OCEDURE'/39X,56('*'))
  410 FORMAT (///1H0,I5,22H ITERATION PARAMETERS:,6D12.3//28X,10D12.3)  ADI2830
      END                                                               ADI2840-
      SUBROUTINE COEF(PHI,KEEP,PHE,STRT,SURI,T,TR,TC,S,WELL,TL,SL,PERM,BCOF   10
     1OTTOM,SY,RATE,RIVER,M,TOP,GRND,DELX,DELY,DDN,WR,NWR,NENT)
C     ------------------------------------------------------------------COF   30
C     COMPUTE COEFFICIENTS                                              COF   40
C     ------------------------------------------------------------------COF   50
C                                                                       COF   60
C     SPECIFICATIONS:                                                   COF   70
      IMPLICIT REAL *8 (A-H,O-Z)                                        $7-10
      REAL *8 KEEP,M                                                    $7-11
      INTEGER R,P,PU,DIML,DIMW
      INTEGER CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD
```

```
     1,CONTR,LEAK,RECH,SIP,ADI,NUMS,IDK1,IDK2
C                                                                       COF 120
      DIMENSION PHI(1),KEEP(1),PHE(1),STRT(1),SURI(1),T(1),TR(1)
     1,TC(1),S(1),WELL(1),TL(1),SL(1),PERM(1),BOTTOM(1),SY(1),
     2RATE(1),RIVER(1),M(1),TOP(1),GRND(1),DELX(1),DELY(1),
     3DDN(1),WR(1),NWR(1)
C                                                                       COF 160
      COMMON /SARRAY/ VF4(11),CHK(15)                                   COF 170
      COMMON /SPARAMI/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,LEAK,
     1RECH,SIP,NUMS,LSOR,ADI,IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,
     2ITMAX,LENGTH,NWEL,NW,DIML,DIMW,I,J,IDK1,IDK2,JNO1,INO1,R,P,PU
      COMMON /SPARAMR/ U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,
     1WIDTH,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,EROR
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1
C                                                                       COF 230
      GO TO (1,2,3) NENT
      DATA PIE/3.141593/                                                COF 235
      RETURN                                                            COF 240
C     ..................................................................COF 250
C                                                                       COF 260
C     ---COMPUTE COEFFICIENTS FOR TRANSIENT PART OF LEAKAGE TERM---     COF 270
C     *********************                                             COF 280
C     ENTRY CLAY(T,S,TL,RATE,M)                                         COF 290
C     *********************                                             COF 300
1     CONTINUE
      TMIN=1.E30                                                        COF 310
      TT=0.0                                                            COF 320
      PRATE=0.                                                          COF 330
      DO 50 I=1,DIML                                                    COF 340
      DO 50 J=1,DIMW                                                    COF 350
      N=I+DIML*(J-1)                                                    COF 360
C                                                                       COF 370
C     ---SKIP COMPUTATIONS IF T, RATE OR M = 0, OR IF CONSTANT          COF 380
C        HEAD BOUNDARY---                                               COF 390
      IF (RATE(N).LE.0..OR.T(N).EQ.0..OR.M(N).EQ.0..OR.S(N).LT.0.) GO TOCOF 400
     1 50                                                               COF 410
C                                                                       COF 420
C     ---IF VALUE FOR TL(N  ) WILL EQUAL VALUE FOR PREVIOUS NODE,       COF 430
C     SKIP PART OF COMPUTATIONS---                                      COF 440
      IF (RATE(N)*M(N).EQ.PRATE) GO TO 40                               COF 450
      DIMT=RATE(N)*SUMP/(M(N)*M(N)*SS*3)                                COF 460
      IF (DIMT.GT.TT) TT=DIMT                                           COF 470
      IF (DIMT.LT.TMIN) TMIN=DIMT                                       COF 480
      PPT=PIE*PIE*DIMT                                                  COF 490
C                                                                       COF 500
C     ---RECOMPUTE PPT IF DIMT WITHIN RANGE FOR SHORT TIME COMPUTATION--COF 510
      IF (DIMT.LT.1.0E-03) PPT=1.0/DIMT                                 COF 520
      CC=(2.3-PPT)/(2.*PPT)                                             COF 530
C                                                                       COF 540
C     ---COMPUTE SUM OF EXPONENTIALS---                                 COF 550
      SUMN=0.0                                                          COF 560
      DO 20 K=1,200                                                     COF 570
      POWER=K*K*PPT                                                     COF 580
      IF (POWER.LE.150.) GO TO 10                                       COF 590
```

```
      POWER=150                                                         COF 600
   10 PEX=DEXP(-POWER)                                                  $$7-20
      SUMN=SUMN+PEX                                                     COF 620
      IF (PEX.GT.0.00009) GO TO 20                                      COF 630
      IF (K.GT.CC) GO TO 30                                             COF 640
   20 CONTINUE                                                          COF 650
C                                                                       COF 660
C     ---COMPUTE DENOMINATER DEPENDING ON VALUE OF DIMT---              COF 670
   30 DENOM=1.0                                                         COF 680
      IF (DIMT.LT.1.0E-03) DENOM=DSQRT(PIE*DIMT)                        $$7-30
C                                                                       COF 700
C     ---HEAD VALUES ARE NOT INCLUDED IN COMPUTATION OF Q FACTOR SINCE  COF 710
C        LEAKAGE IS CONSIDERED IMPLICITLY---                            COF 720
   40 Q1=RATE(N)/(M(N)*DENOM)                                           COF 730
      TL(N)=Q1+2.*Q1*SUMN                                               COF 740
      PRATE=RATE(N)*M(N)                                                COF 750
   50 CONTINUE                                                          COF 760
      TMIN=TMIN*3.0                                                     COF 770
      TT=TT*3.0                                                         COF 780
      RETURN                                                            COF 790
C     ..................................................................COF 800
C                                                                       COF 810
C     ---COMPUTE TRANSMISSIVITY IN WT OR WT-ARTESIAN CONVERSION PROBLEM-COF 820
C     ********************                                              COF 830
C     ENTRY TRANS(PHI,KEEP,SURI,T,TR,TC,WELL,PERM,BOTTOM,TOP,DDN,
C    1WR,NWR,DELX,DELY)
C     ********************                                              COF 850
2     CONTINUE
      DO 60 I=1,DIML                                                    COF 860
      DO 60 J=1,DIMW                                                    COF 870
      N=I+DIML*(J-1)                                                    COF 880
      IF (PERM(N).EQ.0.) GO TO 60                                       COF 890
      HED=PHI(N)                                                        COF 900
      IF (CONVRT.EQ.CHK(7)) HED=DMIN1(PHI(N),TOP(N))                    $$7-40
      T(N)=PERM(N)*(HED-BOTTOM(N))                                      COF 920
      IF (T(N).GT.0.) GO TO 60                                          COF 930
      IF (WELL(N).LT.0.) GO TO 70                                       COF 940
C                                                                       COF 950
C     ---THE FOLLOWING STATEMENTS APPLY WHEN NODES (EXCEPT WELL NODES)  COF 960
C     GO DRY---                                                         COF 970
      PERM(N)=0.                                                        COF 980
      T(N)=0.0                                                          COF 990
      TR(N-DIML)=0.                                                     COF1000
      TR(N)=0.                                                          COF1010
      TC(N-1)=0.                                                        COF1020
      TC(N)=0.                                                          COF1030
      PHI(N)=SURI(N)                                                    COF1040
C     WRITE (P,150) I,J                                                 COF1050
   60 CONTINUE                                                          COF1060
      IF (KT.EQ.0) RETURN                                               COF1070
      GO TO 90                                                          COF1080
C                                                                       COF1090
C     ---START PROGRAM TERMINATION WHEN A WELL GOES DRY---              COF1100
   70 WRITE (P,120) I,J                                                 COF1110
```

```
C     WRITE (P,130)                                                    COF1120
      IERR=1                                                           COF1130
      CALL STEP(PHI,KEEP,STRT,SURI,T,WELL,PERM,BOTTOM,TOP,DELX,DDN
     1,DELY,WR,NWR,TEST3,TR,TC,S,QRE,TL,SY,RATE,RIVER,M,GRND,4)
      DO 80 I=2,INO1                                                   COF1150
      DO 80 J=2,JNO1                                                   COF1160
      N=I+DIML*(J-1)                                                   COF1170
   80 PHI(N)=KEEP(N)                                                   COF1180
      SUM=SUM-DELT                                                     COF1190
      SUMP=SUMP-DELT                                                   COF1200
      KT=KT-1                                                          COF1210
      IF (KT.EQ.0) STOP                                                COF1220
      IF (IDK2.EQ.CHK(15)) THEN
      CALL STEP(PHI,KEEP,STRT,SURI,T,WELL,PERM,BOTTOM,TOP,DELX,DDN
     1,DELY,WR,NWR,TEST3,TR,TC,S,QRE,TL,SY,RATE,RIVER,M,GRND,5)
      ENDIF
      IF (PNCH.EQ.CHK(1)) THEN
      CALL STEP(PHI,KEEP,STRT,SURI,T,WELL,PERM,BOTTOM,TOP,DELX,DDN
     1,DELY,WR,NWR,TEST3,TR,TC,S,QRE,TL,SY,RATE,RIVER,M,GRND,6)
      ENDIF
      IF (MOD(KT,KTH).EQ.0) STOP                                       COF1250
C     WRITE (P,140) KT,SUM                                             COF1260
      CALL STEP(PHI,KEEP,STRT,SURI,T,WELL,PERM,BOTTOM,TOP,DELX,DDN
     1,DELY,WR,NWR,TEST3,TR,TC,S,QRE,TL,SY,RATE,RIVER,M,GRND,4)
      IF(CHCK.EQ.CHK(5)) THEN
      CALL CHECKI(PHI,KEEP,PHE,STRT,T,TR,TC,S,QRE,WELL,TL,PERM,BOTTOM
     1,SY,RATE,RIVER,M,TOP,GRND,DELX,DELY,2)
      ENDIF
      STOP                                                             COF1290
C                                                                      COF1300
C     ---COMPUTE T COEFFICIENTS---                                     COF1310
C     ********************                                             COF1320
C     ENTRY TCOF(T,TR,DELX,DELY)                                       COF1330
3     CONTINUE
C     ********************                                             COF1340
   90 DO 110 I=1,INO1                                                  COF1350
      DO 110 J=1,JNO1                                                  COF1360
      N=I+DIML*(J-1)                                                   COF1370
      NR=N+DIML                                                        COF1380
      NB=N+1                                                           COF1390
      IF (T(N).EQ.0.) GO TO 110                                        COF1400
      IF (T(NR).EQ.0.) GO TO 100                                       COF1410
      TR(N)=(2.*T(NR)*T(N))/(T(N)*DELX(J+1)+T(NR)*DELX(J))*FACTX       COF1420
  100 IF (T(NB).EQ.0.) GO TO 110                                       COF1430
      TC(N)=(2.*T(NB)*T(N))/(T(N)*DELY(I+1)+T(NB)*DELY(I))*FACTY       COF1440
  110 CONTINUE                                                         COF1450
      RETURN                                                           COF1460
C                                                                      COF1470
C     ---FORMATS---                                                    COF1480
C                                                                      COF1490
C     -----------------------------------------------------------------COF1500
C                                                                      COF1510
C                                                                      COF1520
  120 FORMAT ('-************WELL',I3,',',I3,' GOES DRY*************') COF1530
```

```
  130 FORMAT ('1',50X,'DRAWDOWN WHEN WELL WENT DRY')
  140 FORMAT ('1',32X,'DRAWDOWN FOR TIME STEP',I3,'; SIMULATION TIME =',
     11PE15.7,' SECONDS')
  150 FORMAT ('-',20('*'),' NODE ',I4,',',I4,' GOES DRY ',20('*'))
      END
      SUBROUTINE CHECKI(PHI,KEEP,PHE,STRT,T,TR,TC,S,QRE,WELL,TL,PERM,BOT
     1TOM,SY,RATE,RIVER,M,TOP,GRND,DELX,DELY,NENT)
C     ------------------------------------------------------------------
C     COMPUTE A MASS BALANCE
C     ------------------------------------------------------------------
C
C     SPECIFICATIONS:
      IMPLICIT REAL *8 (A-H,O-Z)
      REAL *8 KEEP,M
      INTEGER R,P,PU,DIML,DIMW
      INTEGER CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD
     1,CONTR,LEAK,RECH,SIP,ADI,NUMS,IDK1,IDK2
C
      DIMENSION PHI(IZ,JZ), KEEP(IZ,JZ), PHE(IZ,JZ), STRT(IZ,JZ), T(IZ,J
     1Z), TR(IZ,JZ), TC(IZ,JZ), S(IZ,JZ), QRE(IZ,JZ), WELL(IZ,JZ), TL(IZ
     2,JZ), PERM(IZ,JZ), BOTTOM(IP,JP), SY(IP,JP), RATE(IR,JR), RIVER(IR
     3,JR),M(IR,JR),TOP(IC,JC),GRND(IL,JL),DELX(JZ),DELY(IZ)
      COMMON /BAL/ QE(100,100),QL(100,100),QCH(150),LROW(150),LCOL(150)
C
      COMMON /SARRAY/ VF4(11),CHK(15)
      COMMON /SPARAMI/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,LEAK,
     1RECH,SIP,NUMS,LSOR,ADI,IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,
     2ITMAX,LENGTH,NWEL,NW,DIML,DIMW,I,J,IDK1,IDK2,JNO1,INO1,R,P,PU
      COMMON /SPARAMR/ U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,
     1WIDTH,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,EROR
      COMMON /CK/ ETFLXT,STORT,QRET,CHST,CHDT,FLUXT,PUMPT,CFLUXT,FLXNT
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1
      GO TO (1,2) NENT
      RETURN
C     ..................................................................
C     ********************
C     ENTRY CHECK(PHI,KEEP,STRT,T,WELL,TOP,DELX,DELY,TR,TC,S,
C    2QRE,TL,SY,RATE,RIVER,M,GRND)
1     CONTINUE
C     ********************
C     ---INITIALIZE VARIABLES---
      PUMP=0.
      STOR=0.
      FLUXS=0.0
      CHD1=0.0
      CHD2=0.0
      QREFLX=0.
      CFLUX=0.
      FLUX=0.
      ETFLUX=0.
      FLXN=0.0
      DO 5 ICH=1,100
      QCH(ICH)=0.
      LROW(ICH)=0.
```

```
      LCOL(ICH)=0.
    5 CONTINUE
      ICH=0
C     ..................................................................CHK 420
C                                                                       CHK 430
C     ---COMPUTE RATES,STORAGE AND PUMPAGE FOR THIS STEP---             CHK 440
      QL(1,1)=0.
      QE(1,1)=0.
      DO 240 I=2,DIML                                                   CHK 450
      DO 240 J=2,DIMW                                                   CHK 460
      QL(I,J)=0.
      QE(I,J)=0.
      IF (T(I,J).EQ.0.) GO TO 240                                       CHK 470
      AREA=DELX(J)*DELY(I)                                              CHK 480
      IF (S(I,J).GE.0.) GO TO 120                                       CHK 490
C                                                                       CHK 500
C     ---COMPUTE FLOW RATES TO AND FROM CONSTANT HEAD BOUNDARIES---     CHK 510
      ICH=ICH+1
      LROW(ICH)=I
      LCOL(ICH)=J
      IF (S(I,J-1).LT.0..OR.T(I,J-1).EQ.0.) GO TO 30                    CHK 520
      X=(STRT(I,J)-PHI(I,J-1))*TR(I,J-1)*DELY(I)                        CHK 530
      QCH(ICH)=QCH(ICH)+X
      IF (X) 10,30,20                                                   CHK 540
   10 CHD1=CHD1+X                                                       CHK 550
      GO TO 30                                                          CHK 560
   20 CHD2=CHD2+X                                                       CHK 570
   30 IF (S(I,J+1).LT.0..OR.T(I,J+1).EQ.0.) GO TO 60                    CHK 580
      X=(STRT(I,J)-PHI(I,J+1))*TR(I,J)*DELY(I)                          CHK 590
      QCH(ICH)=QCH(ICH)+X
      IF (X) 40,60,50                                                   CHK 600
   40 CHD1=CHD1+X                                                       CHK 610
      GO TO 60                                                          CHK 620
   50 CHD2=CHD2+X                                                       CHK 630
   60 IF (S(I-1,J).LT.0..OR.T(I-1,J).EQ.0.) GO TO 90                    CHK 640
      X=(STRT(I,J)-PHI(I-1,J))*TC(I-1,J)*DELX(J)                        CHK 650
      QCH(ICH)=QCH(ICH)+X
      IF (X) 70,90,80                                                   CHK 660
   70 CHD1=CHD1+X                                                       CHK 670
      GO TO 90                                                          CHK 680
   80 CHD2=CHD2+X                                                       CHK 690
   90 IF (S(I+1,J).LT.0..OR.T(I+1,J).EQ.0.) GO TO 240                   CHK 700
      X=(STRT(I,J)-PHI(I+1,J))*TC(I,J)*DELX(J)                          CHK 710
      QCH(ICH)=QCH(ICH)+X
      IF (X) 100,240,110                                                CHK 720
  100 CHD1=CHD1+X                                                       CHK 730
      GO TO 240                                                         CHK 740
  110 CHD2=CHD2+X                                                       CHK 750
      GO TO 240                                                         CHK 760
C                                                                       CHK 770
C     ---RECHARGE AND WELLS---                                          CHK 780
  120 QREFLX=QREFLX+QRE(I,J)*AREA                                       CHK 790
      IF (WELL(I,J)) 130,150,140                                        CHK 800
  130 PUMP=PUMP+WELL(I,J)*AREA                                          CHK 810
```

```
      GO TO 150                                                         CHK 820
  140 CFLUX=CFLUX+WELL(I,J)*AREA                                        CHK 830
  150 IF (EVAP.NE.CHK(6)) GO TO 190                                     CHK 840
C                                                                       CHK 850
C     ---COMPUTE ET RATE---                                             CHK 860
      IF (PHI(I,J).GE.GRND(I,J)-ETDIST) GO TO 160                       CHK 870
      ETQ=0.0                                                           CHK 880
      GO TO 180                                                         CHK 890
  160 IF (PHI(I,J).LE.GRND(I,J)) GO TO 170                              CHK 900
      ETQ=QET                                                           CHK 910
      GO TO 180                                                         CHK 920
  170 ETQ=QET/ETDIST*(PHI(I,J)+ETDIST-GRND(I,J))                        CHK 930
  180 ETFLUX=ETFLUX-ETQ*AREA                                            CHK 940
      QE(I,J)=ETQ*AREA
C                                                                       CHK 950
C     ---COMPUTE VOLUME FROM STORAGE---                                 CHK 960
  190 STORE=S(I,J)                                                      CHK 970
      IF (WATER.EQ.CHK(2)) STORE=SY(I,J)                                CHK 980
      IF (CONVRT.NE.CHK(7)) GO TO 230                                   CHK 990
      X=KEEP(I,J)-PHI(I,J)                                              CHK1000
      IF (X) 200,210,210                                                CHK1010
  200 HED1=PHI(I,J)                                                     CHK1020
      HED2=KEEP(I,J)                                                    CHK1030
      X=DABS(X)                                                         $$8-20
      GO TO 220                                                         CHK1050
  210 HED1=KEEP(I,J)                                                    CHK1060
      HED2=PHI(I,J)                                                     CHK1070
  220 STORE=S(I,J)                                                      CHK1080
      IF (HED1-TOP(I,J).LE.0.) STORE=SY(I,J)                            CHK1090
      IF ((HED1-TOP(I,J))*(HED2-TOP(I,J)).LT.0.0) STORE=(HED1-TOP(I,J))/CHK1100
     1X*S(I,J)+(TOP(I,J)-HED2)/X*SY(I,J)                                CHK1110
  230 STOR=STOR+STORE*(KEEP(I,J)-PHI(I,J))*AREA                         CHK1120
C                                                                       CHK1130
C     ---COMPUTE LEAKAGE RATE---                                        CHK1140
      IF (LEAK.NE.CHK(9)) GO TO 240                                     CHK1150
      IF (M(I,J).EQ.0.) GO TO 240                                       CHK1160
      HED1=STRT(I,J)                                                    CHK1170
      IF (CONVRT.EQ.CHK(7)) HED1=DMAX1(STRT(I,J),TOP(I,J))              $$8-30
      HED2=PHI(I,J)                                                     CHK1190
      IF (CONVRT.EQ.CHK(7)) HED2=DMAX1(PHI(I,J),TOP(I,J))               $$8-40
      XX=RATE(I,J)*(RIVER(I,J)-HED1)*AREA/M(I,J)                        CHK1210
      YY=TL(I,J)*(HED1-HED2)*AREA                                       CHK1220
      FLUX=FLUX+XX                                                      CHK1230
      XNET=XX+YY                                                        CHK1240
      QL(I,J)=XNET
      FLUXS=FLUXS+XNET                                                  CHK1250
      IF (XNET.LT.0.) FLXN=FLXN-XNET                                    CHK1260
  240 CONTINUE                                                          CHK1270
C     ..................................................................CHK1280
C                                                                       CHK1290
C     ---COMPUTE CUMULATIVE VOLUMES, TOTALS, AND DIFFERENCES---         CHK1300
      STORT=STORT+STOR                                                  CHK1310
      STOR=STOR/DELT                                                    CHK1320
      ETFLXT=ETFLXT-ETFLUX*DELT                                         CHK1330
```

```
      FLUXT=FLUXT+FLUXS*DELT                                            CHK1340
      FLXNT=FLXNT+FLXN*DELT                                             CHK1350
      FLXPT=FLUXT+FLXNT                                                 CHK1360
      QRET=QRET+QREFLX*DELT                                             CHK1370
      CHDT=CHDT-CHD1*DELT                                               CHK1380
      CHST=CHST+CHD2*DELT                                               CHK1390
      PUMPT=PUMPT-PUMP*DELT                                             CHK1400
      CFLUXT=CFLUXT+CFLUX*DELT                                          CHK1410
      TOTL1=STORT+QRET+CFLUXT+CHST+FLXPT                                CHK1420
      TOTL2=CHDT+PUMPT+ETFLXT+FLXNT                                     CHK1430
      SUMR=QREFLX+CFLUX+CHD2+CHD1+PUMP+ETFLUX+FLUXS+STOR                CHK1440
      DIFF=TOTL2-TOTL1                                                  CHK1450
      PERCNT=0.0                                                        CHK1460
      IF (TOTL2.EQ.0.) GO TO 250                                        CHK1470
      PERCNT=DIFF/TOTL2*100.                                            CHK1480
  250 RETURN                                                            CHK1490
C     ..................................................................CHK1500
C                                                                       CHK1510
C     ---PRINT RESULTS---                                               CHK1520
C     ************************                                          CHK1530
C     ENTRY CWRITE                                                      CHK1540
C     ************************                                          CHK1550
2     CONTINUE
C                                                                       CHK1560
C     WRITE (P,260) STOR,QREFLX,STORT,CFLUX,QRET,PUMP,CFLUXT,ETFLUX,CHSTCHK1570
C    1,FLXPT,CHD2,TOTL1,CHD1,FLUX,FLUXS,ETFLXT,CHDT,SUMR,PUMPT,FLXNT,TOTCHK1580
C    2L2,DIFF,PERCNT                                                    CHK1590
      IF(LEAK.NE.CHK(9)) GO TO 277
C     WRITE(P,281)
      DO 275 I=1,DIML
C275  WRITE(P,280) I,(QL(I,J),J=1,DIMW)
 275  CONTINUE
277   IF(EVAP.NE.CHK(6)) GO TO 282
C     WRITE(P,279)
      DO 276 I=1,DIML
C276  WRITE(P,280) I,(QE(I,J),J=1,DIMW)
  276 CONTINUE
279   FORMAT('1',49X,' EVAPOTRANSPIRATION-FLUX MATRIX ',/,49X,32('-'))
 280  FORMAT (1X,I4,10(1X,1PE11.4),/,(5X,10(1X,1PE11.4)))
 281  FORMAT('1',49X,' LEAKAGE-FLUX MATRIX ',/,49X,21('-'))
C282  WRITE(P,270)(LROW(ICHX),LCOL(ICHX),QCH(ICHX),ICHX=1,ICH)
  282 CONTINUE
      RETURN                                                            CHK1600
C                                                                       CHK1610
C     ---FORMATS---                                                     CHK1620
C                                                                       CHK1630
C     ------------------------------------------------------------------CHK1640
C                                                                       CHK1650
C                                                                       CHK1660
  260 FORMAT ('0',10X,'CUMULATIVE MASS BALANCE:',16X,'L**3',23X,'RATES FCHK1670
     1OR THIS TIME STEP:',16X,'L**3/T'/11X,24('-'),43X,25('-')//20X,'SOUCHK1680
     2RCES:',69X,'STORAGE =',F20.4/20X,8('-'),68X,'RECHARGE =',F20.4/27XCHK1690
     3,'STORAGE =',F20.2,35X,'CONSTANT FLUX =',F20.4/26X,'RECHARGE =',F2CHK1700
     40.2,41X,'PUMPING =',F20.4/21X,'CONSTANT FLUX =',F20.2,30X,'EVAPOTRCHK1710
```

```
     5ANSPIRATION =',F20.4/21X,'CONSTANT HEAD =',F20.2,34X,'CONSTANT HEACHK1720
     6D:'/27X,'LEAKAGE =',F20.2,46X,'IN =',F20.4/21X,'TOTAL SOURCES =',FCHK1730
     720.2,45X,'OUT =',F20.4/96X,'LEAKAGE:'/20X,'DISCHARGES:',45X,'FROM CHK1740
     8PREVIOUS PUMPING PERIOD =',F20.4/20X,11('-'),68X,'TOTAL =',F20.4/1CHK1750
     96X,'EVAPOTRANSPIRATION =',F20.2/21X,'CONSTANT HEAD =',F20.2,36X,'SCHK1760
     $UM OF RATES=',F20.4/19X,'QUANTITY PUMPED =',F20.2/27X,'LEAKAGE =',CHK1770
     $F20.2/19X,'TOTAL DISCHARGE =',F20.2//17X,'DISCHARGE-SOURCES =',F20CHK1780
     $.2/15X,'PER CENT DIFFERENCE =',F20.2//)                           CHK1790
  270 FORMAT(//4X,'CONSTANT HEAD DISCHARGE:'/
     15X,'ROW    COL    DISCHARGE (CFS)'/
     15X,'---    ---    ---------------'/
     2(5X,I3,4X,I3,4X,E11.4))
      END                                                               CHK1800-
      SUBROUTINE PRNTAI(PHI,SURI,T,S,WELL,DELX,DELY,NG,NENT)
C     ------------------------------------------------------------------PRN  20
C     PRINT MAPS OF DRAWDOWN AND HYDRAULIC HEAD                         PRN  30
C     ------------------------------------------------------------------PRN  40
C                                                                       PRN  50
C     SPECIFICATIONS:                                                   PRN  60
      IMPLICIT REAL *8 (A-H,O-Z)                                        $9-10
      REAL *8 MESUR,K                                                   $$9-20
      INTEGER R,P,PU,DIML,DIMW
      INTEGER    CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD
     1,CONTR,LEAK,RECH,SIP,ADI,NUMS,IDK1,IDK2
C                                                                       PRN 110
      DIMENSION PHI(IZ,JZ), SURI(IZ,JZ), S(IZ,JZ), WELL(IZ,JZ), DELX(JZ)PRN 120
     1, DELY(IZ), T(IZ,JZ)                                              PRN 130
C                                                                       PRN 140
      COMMON /SARRAY/ VF4(11),CHK(15)                                   PRN 150
      COMMON /SPARAMI/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,LEAK,
     1RECH,SIP,NUMS,LSOR,ADI,IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,
     2ITMAX,LENGTH,NWEL,NW,DIML,DIMW,I,J,IDK1,IDK2,JNO1,INO1,R,P,PU
      COMMON /SPARAMR/ U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,
     1WIDTH,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,EROR
      COMMON /PRI/ NA(4),N1,N2,N3
      COMMON /PRR/ PRNT(122),DIGIT(122),XN(100),BLANK(60),SYM(17),
     1YN(13),VF1(6),VF2(6),VF3(7),YLABEL(6),TITLE(5),XLABEL(3),XN1,
     2MESUR,XSCALE,YSCALE,DINCH,FACT1,FACT2
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1
      GO TO (1,2) NENT
      RETURN                                                            PRN 250
C     ..................................................................PRN 260
C                                                                       PRN 270
C     ---INITIALIZE VARIABLES FOR PLOT---                               PRN 280
C     ********************                                              PRN 290
C     ENTRY MAP(WELL,DELX,DELY)                                         PRN 300
C     ********************                                              PRN 310
1     CONTINUE
   10 XSF=DINCH*XSCALE                                                  PRN 320
      YSF=DINCH*YSCALE                                                  PRN 330
      NYD=YDIM/YSF                                                      PRN 340
      IF (NYD*YSF.LE.YDIM-DELY(INO1)/2.) NYD=NYD+1                      PRN 350
      IF (NYD.LE.12) GO TO 20                                           PRN 360
      DINCH=YDIM/(12.*YSCALE)                                           PRN 370
```

```
C     WRITE (P,310) DINCH                                               PRN 380
C     IF (YSCALE.LT.1.0) WRITE (P,320)                                  PRN 390
      GO TO 10                                                          PRN 400
   20 NXD=WIDTH/XSF                                                     PRN 410
      IF (NXD*XSF.LE.WIDTH-DELX(JNO1)/2.) NXD=NXD+1                     PRN 420
      N4=NXD*N1+1                                                       PRN 430
      N5=NXD+1                                                          PRN 440
      N6=NYD+1                                                          PRN 450
      N8=N2*NYD+1                                                       PRN 460
      NA(1)=N4/2-1                                                      PRN 470
      NA(2)=N4/2                                                        PRN 480
      NA(3)=N4/2+3                                                      PRN 490
      NC=(N3-N8-10)/2                                                   PRN 500
      ND=NC+N8                                                          PRN 510
      NE=MAX0(N5,N6)                                                    PRN 520
      VF1(3)=DIGIT(ND)                                                  PRN 530
      VF2(3)=DIGIT(ND)                                                  PRN 540
      VF3(3)=DIGIT(NC)                                                  PRN 550
      XLABEL(3)=MESUR                                                   PRN 560
      YLABEL(6)=MESUR                                                   PRN 570
      DO 40 I=1,NE                                                      PRN 580
      NNX=N5-I                                                          PRN 590
      NNY=I-1                                                           PRN 600
      IF (NNY.GE.N6) GO TO 30                                           PRN 610
      YN(I)=YSF*NNY/YSCALE                                              PRN 620
   30 IF (NNX.LT.0) GO TO 40                                            PRN 630
      XN(I)=XSF*NNX/YSCALE                                              PRN 640
   40 CONTINUE                                                          PRN 650
      RETURN                                                            PRN 660
C     ..................................................................PRN 670
C                                                                       PRN 680
C     ********************                                              PRN 690
C     ENTRY PRNTA(PHI,SURI,T,S,WELL,DELX,DELY,NG)
C     ********************                                              PRN 710
2     CONTINUE
C     ---VARIABLES INITIALIZED EACH TIME A PLOT IS REQUESTED---         PRN 720
      DIST=WIDTH-DELX(JNO1)/2.                                          PRN 730
      JJ=JNO1                                                           PRN 740
      LL=1                                                              PRN 750
      Z=NXD*XSF                                                         PRN 760
C     IF (NG.EQ.1) WRITE (P,280) (TITLE(I),I=1,2)                       PRN 770
C     IF (NG.EQ.2) WRITE (P,280) (TITLE(I),I=3,5)                       PRN 780
      DO 270 I=1,N4                                                     PRN 790
C                                                                       PRN 800
C     ---LOCATE X AXES---                                               PRN 810
      IF (I.EQ.1.OR.I.EQ.N4) GO TO 50                                   PRN 820
      PRNT(1)=SYM(12)                                                   PRN 830
      PRNT(N8)=SYM(12)                                                  PRN 840
      IF ((I-1)/N1*N1.NE.I-1) GO TO 70                                  PRN 850
      PRNT(1)=SYM(14)                                                   PRN 860
      PRNT(N8)=SYM(14)                                                  PRN 870
      GO TO 70                                                          PRN 880
C                                                                       PRN 890
C     ---LOCATE Y AXES---                                               PRN 900
```

```
   50 DO 60 J=1,N8                                                      PRN 910
      IF ((J-1)/N2*N2.EQ.J-1) PRNT(J)=SYM(14)                           PRN 920
   60 IF ((J-1)/N2*N2.NE.J-1) PRNT(J)=SYM(13)                           PRN 930
C                                                                       PRN 940
C     ---COMPUTE LOCATION OF NODES AND DETERMINE APPROPRIATE SYMBOL---  PRN 950
   70 IF (DIST.LT.0..OR.DIST.LT.Z-XN1*XSF) GO TO 220                    PRN 960
      YLEN=DELY(2)/2.                                                   PRN 970
      DO 200 L=2,INO1                                                   PRN 980
      J=YLEN*N2/YSF+1.5                                                 PRN 990
      IF (T(L,JJ).EQ.0.) GO TO 140                                      PRN1000
      IF (S(L,JJ).LT.0.) GO TO 190                                      PRN1010
      INDX3=0                                                           PRN1020
      GO TO (80,90), NG                                                 PRN1030
   80 K=(SURI(L,JJ)-PHI(L,JJ))*FACT1                                    PRN1040
C     -TO CYCLE SYMBOLS FOR DRAWDOWN, REMOVE C FROM COL. 1 OF NEXT CARD-PRN1050
C     K=DMOD(K,10.)                                                     $$9-20
      GO TO 100                                                         PRN1070
   90 K=PHI(L,JJ)*FACT2                                                 PRN1080
  100 IF (K) 110,140,120                                                PRN1090
  110 IF (J-2.GT.0) PRNT(J-2)=SYM(13)                                   PRN1100
      N=-K                                                              PRN1110
      IF (N.LT.100) GO TO 130                                           PRN1120
      GO TO 170                                                         PRN1130
  120 N=K                                                               PRN1140
      IF (N.LT.100) GO TO 130                                           PRN1150
      IF (N.GT.999) GO TO 170                                           PRN1160
      INDX3=N/100                                                       PRN1170
      IF (J-2.GT.0) PRNT(J-2)=SYM(INDX3)                                PRN1180
      N=N-INDX3*100                                                     PRN1190
  130 INDX1=MOD(N,10)                                                   PRN1200
      IF (INDX1.EQ.0) INDX1=10                                          PRN1210
C     -TO CYCLE SYMBOLS FOR DRAWDOWN, REMOVE C FROM COL. 1 OF NEXT CARD-PRN1220
C     IF (NG.EQ.1) GO TO 150                                            PRN1230
      INDX2=N/10                                                        PRN1240
      IF (INDX2.GT.0) GO TO 160                                         PRN1250
      INDX2=10                                                          PRN1260
      IF (INDX3.EQ.0) INDX2=15                                          PRN1270
      GO TO 160                                                         PRN1280
  140 INDX1=15                                                          PRN1290
  150 INDX2=15                                                          PRN1300
  160 IF (J-1.GT.0) PRNT(J-1)=SYM(INDX2)                                PRN1310
      PRNT(J)=SYM(INDX1)                                                PRN1320
      GO TO 200                                                         PRN1330
  170 DO 180 II=1,3                                                     PRN1340
      JI=J-3+II                                                         PRN1350
  180 IF (JI.GT.0) PRNT(JI)=SYM(11)                                     PRN1360
  190 IF (S(L,JJ).LT.0.) PRNT(J)=SYM(16)                                PRN1370
  200 YLEN=YLEN+(DELY(L)+DELY(L+1))/2.                                  PRN1380
  210 DIST=DIST-(DELX(JJ)+DELX(JJ-1))/2.                                PRN1390
      JJ=JJ-1                                                           PRN1400
      IF (JJ.EQ.0) GO TO 220                                            PRN1410
      IF (DIST.GT.Z-XN1*XSF) GO TO 210                                  PRN1420
  220 CONTINUE                                                          PRN1430
C                                                                       PRN1440
```

```
C      ---PRINT AXES,LABELS, AND SYMBOLS---                                  PRN1450
       IF (I-NA(LL).EQ.0) GO TO 240                                          PRN1460
       IF ((I-1)/N1*N1-(I-1)) 250,230,250                                    PRN1470
C 230  WRITE (P,VF1) (BLANK(J),J=1,NC),(PRNT(J),J=1,N8),XN(1+(I-1)/6)        PRN1480
  230  CONTINUE
       GO TO 260                                                             PRN1490
C 240  WRITE (P,VF2) (BLANK(J),J=1,NC),(PRNT(J),J=1,N8),XLABEL(LL)           PRN1500
  240  CONTINUE
       LL=LL+1                                                               PRN1510
       GO TO 260                                                             PRN1520
C 250  WRITE (P,VF2) (BLANK(J),J=1,NC),(PRNT(J),J=1,N8)                      PRN1530
  250  CONTINUE
C                                                                            PRN1540
C      ---COMPUTE NEW VALUE FOR Z AND INITIALIZE PRNT---                     PRN1550
  260  Z=Z-2.*XN1*XSF                                                        PRN1560
       DO 270 J=1,N8                                                         PRN1570
  270  PRNT(J)=SYM(15)                                                       PRN1580
C                                                                            PRN1590
C      ---NUMBER AND LABEL Y AXIS AND PRINT LEGEND---                        PRN1600
C      WRITE (P,VF3) (BLANK(J),J=1,NC),(YN(I),I=1,N6)                        PRN1610
C      WRITE (P,300) (YLABEL(I),I=1,6)                                       PRN1620
C      IF (NG.EQ.1) WRITE (P,290) FACT1                                      PRN1630
C      IF (NG.EQ.2) WRITE (P,290) FACT2                                      PRN1640
       RETURN                                                                PRN1650
C                                                                            PRN1660
C      ---FORMATS---                                                         PRN1670
C                                                                            PRN1680
C      -----------------------------------------------------------------PRN1690
C                                                                            PRN1700
C                                                                            PRN1710
  280  FORMAT ('1',53X,4A8//)                                                PRN1720
  290  FORMAT ('0EXPLANATION'/' ',11('-')//' R = CONSTANT HEAD BOUNDARY'/PRN1730
      1' *** = VALUE EXCEEDED 3 FIGURES'/' MULTIPLICATION FACTOR =',F8.3)PRN1740
  300  FORMAT ('0',39X,6A8)                                                  PRN1750
  310  FORMAT ('0',25X,10('*'),' TO FIT MAP WITHIN 12 INCHES, DINCH REVISPRN1760
      1ED TO',G15.7,1X,10('*'))                                              PRN1770
  320  FORMAT ('0',45X,'NOTE: GENERALLY SCALE SHOULD BE > OR = 1.0')         PRN1780
       END                                                                   PRN1790-
       BLOCK DATA                                                            BLD  10
C      ----------                                                            BLD  20
       IMPLICIT REAL *8 (A-H,O-Z)                                            $10-10
       REAL *8 MESUR                                                         $$10-20
       INTEGER R,P,PU,DIML,DIMW
       INTEGER  CHK,WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,LEAK
      1,RECH,SIP,ADI,NUMS,IDK1,IDK2
C                                                                            BLD  60
       COMMON /DPARAM/ RHO,B,D,F,H                                           BLD  70
       COMMON /SARRAY/ VF4(11),CHK(15)                                       BLD  80
       COMMON /SPARAMI/ WATER,CONVRT,EVAP,CHCK,PNCH,NUM,HEAD,CONTR,LEAK,
      1RECH,SIP,NUMS,LSOR,ADI,IERR,KOUNT,IFINAL,NUMT,KT,KP,NPER,KTH,
      2ITMAX,LENGTH,NWEL,NW,DIML,DIMW,I,J,IDK1,IDK2,JNO1,INO1,R,P,PU
       COMMON /SPARAMR/ U,SS,TT,TMIN,ETDIST,QET,ERR,TMAX,CDLT,HMAX,YDIM,
      1WIDTH,DELT,SUM,SUMP,SUBS,STORE,TEST,ETQB,ETQD,FACTX,FACTY,EROR
       COMMON /PRI/ NA(4),N1,N2,N3
```

```
      COMMON /PRR/ PRNT(122),DIGIT(122),XN(100),BLANK(60),SYM(17),
     1YN(13),VF1(6),VF2(6),VF3(7),YLABEL(6),TITLE(5),XLABEL(3),XN1,
     2MESUR,XSCALE,YSCALE,DINCH,FACT1,FACT2
      COMMON /ARSIZE/ IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IH,IMAX,IMX1
C     ******************************************************************
C
      DATA IZ,JZ,IP,JP,IR,JR,IC,JC,IL,JL,IS,JS,IMAX/13*20/,IH/1/
      DATA CHK/'PUNC','WATE','CONT','NUME','CHEC','EVAP','CONV','HEAD','
     1LEAK','RECH','SIP','LSOR','ADI','DK1 ','DK2 '/,R,P,PU/15,16,17/,
     2B,D,F,H/4*0.D0/
      DATA SYM/'1','2','3','4','5','6','7','8','9','0','*','[','-','+','
     1 ','R','W'/
      DATA PRNT/122*' '/,N1,N2,N3,XN1/6,10,133,.833333333D-1/,BLANK/60*'
     1 '/,NA(4)/1000/
      DATA XLABEL/' X DIS- ','TANCE IN',' MILES '/,YLABEL/'DISTANCE',
     1'FROM OR','IGIN IN ','Y DIRECT','ION, IN ','MILES   '/,TITLE/
     2'PLOT OF ','DRAWDOWN','PLOT OF ','HYDRAULI','C HEAD  '/
      DATA DIGIT/'1','2','3','4','5','6','7','8','9','10','11','12','13'
     1,'14','15','16','17','18','19','20','21','22','23','24','25','26',
     2'27','28','29','30','31','32','33','34','35','36','37','38','39','
     340','41','42','43','44','45','46','47','48','49','50','51','52','5
     43','54','55','56','57','58','59','60','61','62','63','64','65','66
     5','67','68','69','70','71','72','73','74','75','76','77','78',' 79
     6','80','81','82','83','84','85','86','87','88','89','90','91','92'
     7,'93','94','95','96','97','98','99','100','101','102','103',' 104'
     8,'105','106','107','108','109','110','111','112','113','114','115'
     9,'116','117','118','119','120','121','122'/
      DATA VF1/'(1H ',',','    ','A1,F','10.2',')'/
      DATA VF2/'(1H ',',','    ','A1,1','X,A8',')'/
      DATA VF3/'(1H0',',','    ','A1,F','3.1,','12F1','0.2)'/
      DATA VF4/'(1H0',',','    ','X,I2',',2X,','20F6','.1/(','    ','X,2
     10','F6.1','))'/
C     ******************************************************************
      END
```


☆U.S. GOVERNMENT PRINTING OFFICE:1987-181-407:60012