A COMPUTER PROGRAM (MODFLOWP) FOR ESTIMATING PARAMETERS OF A TRANSIENT, THREE-DIMENSIONAL, GROUND-WATER FLOW MODEL USING NONLINEAR REGRESSION

by Mary C. Hill

U.S. GEOLOGICAL SURVEY

Open-File Report 91-484

Denver, Colorado
1992

U.S. DEPARTMENT OF THE INTERIOR

MANUEL LUJAN, JR., Secretary

U.S. GEOLOGICAL SURVEY

Dallas L. Peck, Director

For additional information
write to:

Regional Research Hydrologist
U.S. Geological Survey
Water Resources Division
Box 25046, Mail Stop 413
Denver Federal Center
Denver, CO 80225-0046

Copies of this report can
be purchased from:

U.S. Geological Survey
Books and Open-File Reports Section
Box 25425, Mail Stop 517
Federal Center
Denver, CO 80225-0425

PREFACE

This report presents a computer program for estimating parameters of ground-water flow simulations by using nonlinear regression. The program documented in this report is designed for incorporation into a modified version of the modular finite-difference ground-water flow model developed by the U.S. Geological Survey, which is also documented in this report. The performance of this computer program has been tested in models of both hypothetical and actual ground-water flow systems. Future applications, however, might reveal errors that were not detected in the test simulations. Users are requested to notify the originating office of any errors found in the report or in the computer program. Updates might occasionally be made to both the report and computer program. Users who wish to be added to the mailing list to receive notice of updates, if any, can send a request to:

U.S. Geological Survey
P.O. Box 25046, MS 413
Denver, CO 80225-0046

Copies of the computer program and test data sets on tape or diskette are available at cost of processing from:

U.S. Geological Survey
WATSTORE Program Office
437 National Center
Reston, VA 22092
Telephone: (703) 648-5695

CONTENTS

CONTENTS--Continued

CONTENTS--Continued

FIGURES

FIGURES--Continued

TABLES

CONVERSION FACTORS

Multiply	By	To Obtain
cubic foot per second (ft^3/s)	0.3048	cubic meter per second
foot (ft)	0.3048	meter
foot per second (ft/s)	0.3048	meter per second
inch (in.)	25.4	millimeter
mile (mi)	1.609	kilometer
square feet (ft^2)	0.09290	square meters
square mile (mi^2)	2.590	square kilometer

A COMPUTER PROGRAM (MODFLOWP) FOR ESTIMATING PARAMETERS OF A TRANSIENT, THREE-DIMENSIONAL, GROUND-WATER FLOW MODEL USING NONLINEAR REGRESSION

By Mary C. Hill

ABSTRACT

This report documents a new version of the U.S. Geological Survey modular, three-dimensional, finite-difference, ground-water flow model (MODFLOW) which, with the new Parameter-Estimation Package that also is documented in this report, can be used to estimate parameters by nonlinear regression. The new version of MODFLOW is called MODFLOWP (pronounced MOD•FLOW•P), and functions nearly identically to MODFLOW when the Parameter-Estimation Package is not used. Parameters are estimated by minimizing a weighted least-squares objective function by the modified Gauss-Newton method or by a conjugate-direction method. Parameters used to calculate the following MODFLOW model inputs can be estimated: Transmissivity and storage coefficient of confined layers; hydraulic conductivity and specific yield of unconfined layers; vertical leakance; vertical anisotropy (used to calculate vertical leakance); horizontal anisotropy; hydraulic conductance of the River, Streamflow-Routing, General-Head Boundary, and Drain Packages; areal recharge rates; maximum evapotranspiration; pumpage rates; and the hydraulic head at constant-head boundaries. Any spatial variation in parameters can be defined by the user. Data used to estimate parameters can include existing independent estimates of parameter values, observed hydraulic heads or temporal changes in hydraulic heads, and observed gains and losses along head-dependent boundaries (such as streams). Model output includes statistics for analyzing the parameter estimates and the model; these statistics can be used to quantify the reliability of the resulting model, to suggest changes in model construction, and to compare results of models constructed in different ways.

INTRODUCTION

Terminology

Some of the terms used in this report are so fundamental to the discussion that they need to be defined before proceeding. In this report, the term 'parameter' refers to a quantity being estimated. Thus, typical parameters would be, for example, the hydraulic conductivity of defined areas of confined model layers 4 and 5, the recharge rate applied to a defined area of model layer 1 during specified time steps, or a value which, when multiplied by defined constants, produces the hydraulic conductance for selected cells listed in a River Package input file. The values produced by parameter estimation are estimates of these parameters.

In contrast, the term 'model input' refers to the properties required in the input files of MODFLOW, the U.S. Geological Survey (USGS) modular, three-dimensional finite-difference ground-water flow model. The input files are described by McDonald and Harbaugh (1988) and Prudic (1989). Typical model inputs are transmissivity of confined layers, hydraulic conductivity of unconfined layers, and recharge flux. Many model inputs can be calculated using parameters.

Use of the term 'model' in this report also might cause confusion. To coordinate with usage in the regression literature (for example, Fuller, 1987, p. 9), in this report the term model generally refers to the equations and assumptions used to represent a physical system. In this sense, a model includes not only the numerical representation of the physical equations, but also includes assumptions made to represent a physical system. To illustrate, consider linear models, which are used for illustrative purposes in this report. Linear models are of the form presented in equation (6). Assumptions required to create a linear model include how many parameters and independent variables to use. For nonlinear models, such as those constructed using the numerical representation of the ground-water flow equations as described for MODFLOW by McDonald and Harbaugh (1988) and Prudic (1989), typical assumptions concern the location and type of boundary conditions, the definition of parameters to be estimated, the inclusion or exclusion of physical features, and so on. Models of a physical system, therefore, can differ in one or several of the assumptions used in model construction. In general, models also can differ by being based on different physical equations--for example, one model might only include flow through porous media, as in MODFLOW, whereas another also might represent flow through a fracture. No capacity to vary the physical equations, however, is provided in this report.

Confusion results because the numerical representation of the ground-water flow equation presented by McDonald and Harbaugh (1988) and Prudic (1989) (MODFLOW) also is called a model. To resolve any confusion, in this report all references to the numerical representation of the ground-water flow equation include citation of McDonald and Harbaugh (1988) and Prudic (1989) or reference to MODFLOW or MODFLOWP, and an effort has been made to clearly state that in other references to models, the more extensive definition discussed above is implied.

Problem

Most numerical models of ground-water flow systems need to be calibrated--that is, the model needs to be made to match the physical system being modeled. The model and the physical system are compared based on calibration criteria that are defined by the user. For example, typical calibration criteria are that model parameter values are to be consistent with independent estimates of associated field parameters, and that simulated hydraulic-head values are to be similar to observed values.

Numerical models of ground-water flow systems can be calibrated by trial-and-error, in which simulated aspects of the physical system are repeatedly, manually changed until the model satisfactorily matches the physical system as measured using the defined calibration criteria. Although trial-and-error calibration is conceptually simple, it has three limitations. First, there is no way to know if the estimated parameter values satisfy the calibration criteria better than some untested set of parameter values. This lack of knowing makes it difficult to test hypotheses about a ground-water flow system because a model constructed using one hypothesis might produce better results because of the parameter values used, and not because that hypothesis is better than another. (The process of comparing different hypotheses is called model discrimination.) Second, it is difficult to determine if estimated parameters are highly correlated--that is, that coordinated changes in model parameters would produce identical results in terms of the calibration criteria. When high correlations are present, it is impossible to uniquely estimate the parameter values. Third, the reliability of parameter estimates and simulated results can only be assessed by the tedious process of manually perturbing parameter values to perform a sensitivity analysis. The process also is inexact because results depend on how much the parameter values are perturbed, and the appropriate value is generally unknown. The lack of precision makes it difficult to evaluate whether the calibrated model is accurate enough to be used to make conclusions about the aquifer system or to predict aquifer response.

Alternatively, numerical models of ground-water flow systems can be calibrated by nonlinear regression, in which the model itself is used to determine changes in parameter values. Nonlinear regression is accomplished in the following steps:

1. Using the calibration criteria, define an objective function, which is a measure of how closely the model matches the physical system.

2. Determine the parameter values that produce the smallest value of the objective function. This is called minimization or optimization of the objective function, and, using the Parameter-Estimation Package of MODFLOWP, can be accomplished with either the modified Gauss-Newton method or a conjugate-direction method. Because the ground-water flow equation is nonlinear with respect to many of the parameters that are most commonly estimated, the optimization methods are iterative--that is, the same procedure is repeated to update parameter values until the optimal parameter values are reached.

3. Calculate statistics by which model discrimination and assessment of model reliability can be accomplished easily and objectively.

Purpose and Scope

This report documents the changes made to MODFLOW, the USGS modular, transient, three-dimensional, ground-water flow model, to create MODFLOWP (pronounced MOD•FLOW•P). When used in conjunction with the new Parameter-Estimation Package, which also is documented in this report, MODFLOWP is designed to estimate parameters of ground-water flow simulations that are steady-state or transient or both using nonlinear regression. When used without the Parameter-Estimation Package, MODFLOWP performs almost exactly like MODFLOW.

This report is intended to be used in conjunction with three other publications. The first two are the documentation of MODFLOW (McDonald and Harbaugh, 1988; Prudic, 1989). When applicable, their package and variable names are repeated in this report, and the reader might need to refer to those two publications for a complete discussion. The third publication is Cooley and Naff's (1990) teaching manual and documentation for a steady-state, two-dimensional, ground-water flow model with nonlinear regression. Many of the ideas presented in this report are discussed in more detail by Cooley and Naff (1990), and that publication is referenced when those ideas are discussed.

Parameters are estimated by using existing, independent estimates of parameter values (called prior estimates of the parameters), measured hydraulic heads and temporal changes in hydraulic head, and measured gains and losses along head-dependent boundaries (called observations of dependent variables). Parameters that are used to calculate the following model inputs can be estimated: properties of confined or unconfined aquifers; horizontal anisotropy; vertical anisotropy (used to calculate the model input vertical leakance); hydraulic conductance of selected cells of the River, Streamflow-Routing, General-Head Boundary, or Drain Packages (McDonald and Harbaugh, 1988; Prudic, 1989); areal recharge rates; maximum evapotranspiration; pumpage rates; and the hydraulic head at constant-head boundaries (head can vary linearly along the boundary).

The report begins with brief descriptions of numerical modeling of ground-water flow, followed by discussions of linear regression, nonlinear regression, and the calculation of sensitivity-equation sensitivities used in modified Gauss-Newton optimization and the gradient of the objective function calculated by the adjoint-state method and used in conjugate-direction optimization. Graphical and statistical methods for analyzing the discrepancies between observed and simulated dependent-variable values, and statistical methods for analyzing estimated parameter values and discriminating between different models are presented. There are detailed instructions for using the computer program, including data-entry formats. Two test cases, data listings and outputs for those cases, a brief description of all new and modified modules, and a listing of the FORTRAN program are included. An example of how regression can be applied to a transient, three-dimensional ground-water flow problem is shown in Yager (1991).

Solving Matrix Equations

Parameter estimation requires the solution of many matrix equations with dimensions equal to the number of active cells in the finite-difference grid used to represent a ground-water flow system. These matrix equations are solved most successfully by using direct (noniterative) or conjugate-gradient solvers. The strongly implicit procedure (SIP) proved to be impractical because one iteration parameter seed (McDonald and Harbaugh, 1988, p. 12-23) did not produce convergence for all matrix equations for some test cases, and other solvers, such as slice-successive overrelaxation (SSOR) and alternating-direction implicit (ADI), tend to be slow (Trescott and Larson, 1977; Aziz and Settari, 1979, p. 281-294). A direct D4 solver module for two-dimensional problems simulated using MODFLOW was used by Hill (1990a), and is very efficient for two-dimensional problems. The D4 solver is limited, however, because round-off error becomes significant for problems with more than 500 to 1,000 active finite-difference cells. The D4 module has not been documented and is not readily available for users. Conjugate-gradient solver modules have been developed for MODFLOW by Kuiper (1987), Meyer and others (1989), Scandrett (1989), and Hill (1990b), and the first and last modules are readily available for users. Conjugate-gradient solvers generally are less efficient than D4 solvers for small problems, but can be used to accurately solve large problems (Hill, 1990a). Problems with nonlinearities such as a water-table layer or nonlinear head-dependent boundary, need an iterative solver.

Expertise Needed to Use this Package

Parameter estimation is complicated, and use of this package with an inadequate background can easily produce fallacious conclusions. It is assumed that readers of this report are familiar with the modeling of ground-water flow and matrix addition and multiplication, and have a statistical background equivalent to one college statistics course. Readers unfamiliar with the modeling of ground-water flow need to read one of the available texts on the subject, such as Wang and Anderson (1982). Readers unfamiliar with matrix addition and multiplication need to read and do exercises from Gere and Weaver (1965). Readers unfamiliar with statistics and linear regression need to read and do exercises from Benjamin and Cornell (1970) or another basic text on statistics, and Draper and Smith (1981). Readers unfamiliar with nonlinear regression need to read and do exercises from Bard (1974), Draper and Smith (1981), and Cooley and Naff (1990).

Notation

Throughout this report, matrices and vectors are presented using the following notation:

Capital or Greek letters underlined twice indicate matrices: $\underline{\underline{A}}$, $\underline{\underline{\omega}}$.

A number underlined twice is a matrix, and all entries are equal to that number: $\underline{\underline{1}}$ is a matrix of ones.

Lower-case letters underlined once indicate column vectors: $\underline{f}$, $\underline{y}$.

A number underlined once is a column vector, and all entries are equal to that number: $\underline{0}$ is a vector of zeros.

The element located in matrix row i and column j is designated as follows: matrix $\underline{\underline{A}}$, elements $a_{i,j}$; matrix $\underline{\underline{V}}$, elements $v_{i,j}$.

The ith element of vector $\underline{f}$ is f_i.

$\underline{\underline{A}}^T$ is the transpose of matrix $\underline{\underline{A}}$.

$\underline{e}^T$ is a row vector with the same elements as $\underline{e}$.

$\underline{\underline{A}}^{-1}$ is the inverse of matrix $\underline{\underline{A}}$.

$\underline{\underline{X}}$ is an N by M matrix if it has N rows and M columns. A column vector might be N by 1, a row vector might be 1 by M

$\underline{\underline{I}}$ is the identity matrix. Elements equal 1 along the diagonal and are zero elsewhere.

Given a square, symmetric matrix $\underline{\underline{A}}$, $\underline{\underline{A}}^2 = \underline{\underline{A}}^T\underline{\underline{A}}$; $\underline{\underline{A}}^{½} = B$, where B is defined so $\underline{\underline{B}}^T\underline{\underline{B}} = \underline{\underline{A}}$.

If the scalar variable S is a function of vector $\underline{a}$ then:

$\frac{\partial S}{\partial \underline{a}}$ is a column vector with the ith element equal to $\partial S/\partial a_i$.

If the vector $\underline{a}$ is a function of the scalar α, then:

$\partial \underline{a}/\partial \alpha$ is a column vector with the ith element equal to $\partial a_i/\partial \alpha$.

Derivatives of matrices with respect to vectors produce arrays with more than two dimensions. These are discussed in detail when they are mentioned in the text.

Subscripts also are used to designate row, column, and layer number of the finite-difference grid and sequential cell number, and are defined when they are mentioned in the text.

Some module variables are used in the text to expedite relating the text to the modules. These variables are written in uppercase letters, and are defined in the text where they are introduced and in the "List of Variables" in Appendix C.

Acknowledgments

The author would like to thank Richard M. Yager, (U.S. Geological Survey, Ithaca, New York) for using early versions of this report and computer program. His efforts toward identifying and correcting errors, and his ideas for making MODFLOWP more useful were exceptional and are greatly appreciated. The author would also like to thank David S. Morgan and Erin A. Lynch (U.S.G.S., Eugene, Oregon, and Lansing, Michigan, respectively) for helping to identify and correct errors in early versions.

Ground-Water Flow Equation

MODFLOW is documented extensively by McDonald and Harbaugh (1988) and Prudic (1989), and only those parts needed to understand MODFLOWP and the Parameter-Estimation Package are repeated here.

In their development, McDonald and Harbaugh (1988, ch. 2) discretized the ground-water flow equation spatially by using the block-centered, finite-difference method, and they implicitly differenced the equation in time. For one finite-difference cell, the discretized, differenced ground-water flow equation can be written as (McDonald and Harbaugh, 1988, p. 2-19):

$$\begin{aligned}
&CR_{i,j-1/2,k}(h^n_{i,j-1,k}-h^n_{i,j,k}) + CR_{i,j+1/2,k}(h^n_{i,j+1,k}-h^n_{i,j,k}) \\
&+ CC_{i-1/2,j,k}(h^n_{i-1,j,k}-h^n_{i,j,k}) + CC_{i+1/2,j,k}(h^n_{i+1,j,k}-h^n_{i,j,k}) \\
&+ CV_{i,j,k-1/2}(h^n_{i,j,k-1}-h^n_{i,j,k}) + CV_{i,j,k+1/2}(h^n_{i,j,k+1}-h^n_{i,j,k}) \\
&+ P^n_{i,j,k}h^n_{i,j,k} + Q^n_{i,j,k} = S_{s_{i,j,k}} \frac{(\Delta r_j \Delta c_i \Delta v_k)(h^n_{i,j,k}-h^{n-1}_{i,j,k})}{t_n - t_{n-1}}
\end{aligned} \qquad (1)$$

where n is the time step at which hydraulic head is being calculated, and n-1 is the previous time step;

i,j,k are the row, column, and layer of the finite-difference cell, and, thus, identify spatial location;

Δr_j is the length of the cell as measured along a row (L);

Δc_i is the length of the cell as measured along a column (L);

Δv_k is the length of the cell as measured perpendicular to the orientation of the layer (L);

$h^n_{i,j,k}$ is the hydraulic head in the center of cell i,j,k at the end of time step n;

$CR_{i,j-1/2,k}$, $CR_{i,j+1/2,k}$ are the lateral hydraulic conductances between the cell at i,j,k and the two neighboring cells in row i (L^2/T);

$CC_{i-1/2,j,k}$, $CC_{i+1/2,j,k}$ are the lateral hydraulic conductances between the cell at i,j,k and the two neighboring cells in column j (L /T);

$CV_{i,j,k-1/2}$, $CV_{i,j,k+1/2}$ are the vertical hydraulic conductances between the cell at i,j,k and the two neighboring cells in adjoining model layers (L^2/T);

$P^n_{i,j,k}$ is the sum of all conductances of head-dependent boundary conditions applicable at cell i,j,k at time step n (L^2/T);

$Q^n_{i,j,k}$ is the sum of all sources or sinks or both at cell i,j,k at time step n, the conductances of head-dependent boundaries multiplied by the known hydraulic head of the head-dependent boundary condition at cell i,j,k at time step n, and terms related to constant-head boundaries (L^3/T); and

$S_{s_{i,j,k}}$ is the specific storage, or the specific yield divided by Δv_k, at cell i,j,k.

The calculation of the hydraulic conductances is discussed by McDonald and Harbaugh (1988, ch. 5), and can be summarized as:

$$CR_{i,j+1/2,k} = 2\Delta c_i \frac{TR_{i,j,k}\, TR_{i,j+1,k}}{TR_{i,j,k}\Delta r_{j+1} + TR_{i,j+1,k}\Delta r_j}, \qquad (2a)$$

$$CC_{i+1/2,j,k} = 2\Delta r_j \frac{TC_{i,j,k}\, TC_{i+1,j,k}}{TC_{i,j,k}\Delta c_{i+1} + TC_{i+1,j,k}\Delta c_i}, \qquad (2b)$$

$$CV_{i,j,k+1/2} = \left(\sum_{g=1}^{m} \frac{\Delta z_g}{K_g}\right)^{-1} \Delta r_j \Delta c_i, \qquad (2c)$$

where $TR_{i,j,k}$ is the transmissivity along the row at cell i,j,k, and equals the hydraulic conductivity along the row multiplied by saturated thickness (L^2/T);

$TC_{i,j,k}$ is the transmissivity along the column at cell i,j,k, and equals the hydraulic conductivity along the column multiplied by the saturated thickness (L^2/T);

m	is the number of strata between cell centers in model layers *k* and *k*+1 that have different values of vertical hydraulic conductivity;
Δz_g	is the thickness of each of the m strata (L);
K_g	is the vertical hydraulic conductivity of each of the m strata (L/T).
$\left(\sum_{g=1}^{m} \frac{\Delta z_g}{K_g}\right)^{-1}$	is the vertical leakance (1/T).

The volume of aquifer material that is accounted for by each of the conductances is shown in figure 1.

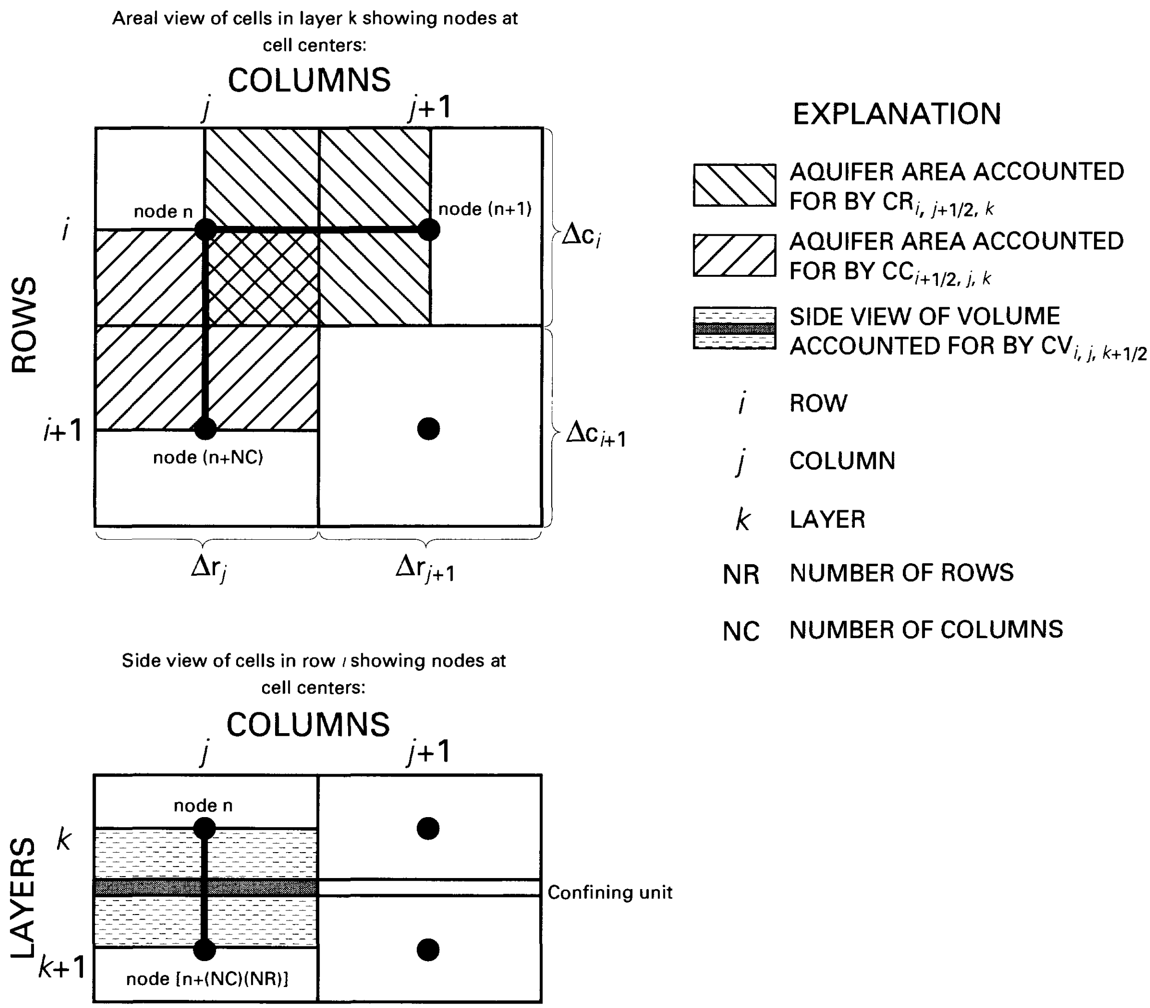


Figure 1.--Aquifer-system volumes accounted for by conductances $CR_{i,j+½,k}$, $CC_{i+½,j,k}$, and $CV_{i,j,k+½}$ in the block-centered, finite-difference method.

If equation (1) is written for each cell of a finite-difference grid, the resulting equations can be expressed in matrix form as:

$$\underline{A}(n)\underline{h}(n) = \underline{B}(n)\underline{h}(n-1) - \underline{f}(n) \qquad (3)$$

where $\underline{h}(n)$ is a vector of hydraulic heads at all grid points at the end of time step n [L];

$\underline{A}(n)$ equals $\frac{-\underline{S}}{\Delta t(n)} + \underline{K} + \underline{P}(n)$ $[L^2/T]$;

$\underline{S}$ is a diagonal matrix of specific storage multiplied by cell volume, or specific yield multiplied by cell area $[L^2]$;

$\Delta t(n)$ is the length of time step n [T];

$\underline{K}$ is a matrix of horizontal and vertical conductances $[L^2/T]$;

$\underline{P}(n)$ is a diagonal matrix of conductances at head-dependent boundaries $[L^2/t]$;

$\underline{B}(n)$ equals $\frac{-\underline{S}}{\Delta t(n)}$ $[L^2/T]$; and

$\underline{f}(n)$ is a vector of the $Q^n_{i,j,k}$, and is sometimes referred to as the forcing function $[L^3/T]$.

Of the matrices from which $\underline{A}$ and $\underline{B}$ are calculated, only $\underline{K}$ has nonzero components off the main diagonal. The structure of $\underline{K}$ for a problem with two rows, three columns, and two layers is shown in figure 2, using subscripts to denote row, layer, and column numbers. $\underline{K}$ is symmetric, and all nonzero components off the main diagonal occur on six off diagonals. The components, $u_{i,j,k}$, on the main diagonal of $\underline{K}$ are calculated as the negative sum of the off-diagonal components for that row. For example:

$$u_{1,1,1} = CR_{1,1+½,1} + CC_{1+½,1,1} + CV_{1,1,1}$$

$$u_{1,2,1} = CR_{1,1+½,1} + CR_{1,2+½,1} + CC_{1+½,2,1} + CV_{1,2,1}.$$

The Parameter-Estimation Package can accommodate confined or unconfined model layers (LAYCON of the Block-Centered Flow Package=0 or 1), but is not designed to accommodate convertible layers (LAYCON=2 or 3). Finite-difference cells in water-table layers become inactive ('go dry') when the simulated hydraulic head declines below the bottom of a water-table layer. MODFLOW does not allow these cells to become active again within a single

$$
\begin{bmatrix}
u_{1,1,1} & -CR_{1,1+1/2,1} & 0 & -CC_{1+1/2,1,1} & 0 & 0 & -CV_{1,1,1+1/2} & 0 & 0 & 0 & 0 & 0 \\
 & u_{1,2,1} & -CR_{1,2+1/2,1} & 0 & -CC_{1+1/2,2,1} & 0 & 0 & -CV_{1,2,1+1/2} & 0 & 0 & 0 & 0 \\
 & & u_{1,3,1} & 0 & 0 & -CC_{1+1/2,3,1} & 0 & 0 & -CV_{1,3,1+1/2} & 0 & 0 & 0 \\
 & & & u_{2,1,1} & -CR_{2,1+1/2,1} & 0 & 0 & 0 & 0 & -CV_{2,1,1+1/2} & 0 & 0 \\
 & & & & u_{2,2,1} & -CR_{2,2+1/2,1} & 0 & 0 & 0 & 0 & -CV_{2,2,1+1/2} & 0 \\
 & \text{symmetric} & & & & u_{2,3,1} & 0 & 0 & 0 & 0 & 0 & -CR_{2,3,1+1/2} \\
 & & & & & & u_{1,1,2} & -CR_{1,1+1/2,2} & 0 & -CC_{1+1/2,1,2} & 0 & 0 \\
 & & & & & & & u_{1,2,1} & -CR_{1,2+1/2,2} & 0 & -CC_{1+1/2,2,1} & 0 \\
 & & & & & & & & u_{1,3,2} & 0 & 0 & -CC_{1+1/2,3,2} \\
 & & & & & & & & & u_{2,1,2} & -CR_{2,1+1/2,2} & 0 \\
 & & & & & & & & & & u_{2,2,2} & -CR_{2,2+1/2,2} \\
 & & & & & & & & & & & u_{2,3;2}
\end{bmatrix}
$$

Figure 2.--Matrix $\underline{K}$ of the finite-difference method for a problem with two rows, three columns, and two layers.

simulation (McDonald and Harbaugh, 1988, p. 5-9). The Parameter-Estimation Package, however, reactivates these cells at the beginning of each parameter-estimation iteration (see chapter in this report on nonlinear regression for a discussion of parameter-estimation iterations).

Estimated parameters can be used to calculate the model inputs listed in table 1. These model inputs are the quantities listed in the MODFLOW input instructions (McDonald and Harbaugh, 1988; Prudic, 1989). The component of equation (3) in which each model input is included also is noted.

Table 1.--*Model inputs listed in the MODFLOW input instructions (McDonald and Harbaugh, 1988; Prudic, 1989) that can be calculated using parameters estimated by using the Parameter-Estimation Package, and the terms of equation (3) in which they occur*

Model Inputs	Term in Equation (3)
Properties of model layers	
Confined layer	
Transmissivity[1,2]	$\underline{K}$
Storage coefficient[1]	$\underline{S}$
Unconfined layer	
Hydraulic conductivity[1,2]	$\underline{K}$
Specific yield[1]	$\underline{S}$
All layers	
Horizontal anisotropy by layer[1]	$\underline{K}$
Vertical leakance between layers[1,2]	$\underline{K}$
Vertical anisotropy between layers (used to calculate vertical leakance)[1]	$\underline{K}$
Head-dependent boundary conductances	$\underline{P}(n);\underline{f}(n)$
River Package[1]	
Streamflow-Routing Package[1]	
General Head-Dependent Boundary Package[1]	
Drain Package[1]	
Maximum evapotranspiration	$\underline{P}(n);\underline{f}(n)$
Constant-head boundaries	$\underline{f}(n)$
Stresses	$\underline{f}(n)$
Pumpage	
Recharge	

[1] Parameters can be defined by a natural-log transformation.
[2] One parameter can be used to calculate any combination of transmissivity of confined layers, hydraulic conductivity of unconfined layers, and vertical leakance between layers.

Parameterization

Although the ground-water flow model permits different model-input values and, therefore, parameter values to be assigned to each model cell, and many model inputs can equal different values at different time steps, estimating this many values is impractical given the data available for most problems. Decreasing the number of parameter values permits them to be reliably estimated with the available data (the reliability of estimates is discussed later in this report). In general, the number of parameter values estimated needs to be a fraction of the number of hydraulic-head and streamflow gain-or-loss observations used to estimate them.

The number of unique parameter values to be estimated can be decreased by assuming that some of the values are known and by parameterization (Shah and others, 1978). In parameterization, a few parameter values are used to define the model-input values for many cells and time steps. Yager (1991) provides an example for a transient, three-dimensional ground-water flow system.

In the Parameter-Estimation Package, spatially varying values of model inputs can be calculated from parameters using cell-by-cell multiplicative factors, multiplication arrays, and zone arrays. Multiplication and zone arrays are used only for model inputs that are read as arrays by MODFLOW (McDonald and Harbaugh, 1988), which includes all 'Properties of model layers' listed in table 1 except horizontal and vertical anisotropy, and maximum evapotranspiration and recharge.

As an example for a model input that is not read as an array by MODFLOW, consider the conductance of the River Package. The conductance for a cell equals the hydraulic conductivity of the riverbed times the area the river occupies within the cell, divided by the riverbed thickness (McDonald and Harbaugh, 1988, p. 6-5). The Parameter-Estimation Package allows the user to define a different multiplicative factor for each river cell included in defining a parameter. Spatial variations in riverbed hydraulic conductivity, area, and thickness, therefore, can be included in multiplicative factors, and a spatially constant parameter value can be estimated. If the riverbed hydraulic conductivity can be considered to be constant for the cells, the parameter can be defined as being equivalent to that riverbed hydraulic conductivity, and the multiplicative factors would equal the area of the river in each cell divided by the riverbed thickness. Alternatively, the initial estimates of riverbed conductance can be used as the multiplicative factors, and the parameter value initially set to 1.0. Then, any change in the parameter value would indicate the change in riverbed conductance produced by nonlinear regression. For example, a value of 0.50 would indicate a 50-percent decrease in the riverbed conductance for the included cells.

As an example for a model input read as an array by MODFLOW, consider the transmissivity of a confined model layer. If it is reasonable to assume that the hydraulic conductivity is constant, the parameter could be defined as the hydraulic conductivity. The multiplicative factor times the multiplication array would equal layer thickness; a zone array could be used if the parameter only represented the hydraulic conductivity of part of the layer. If it is known that the hydraulic conductivity is smaller in one part of the layer, this could be represented by decreasing the elements of the multiplication array in that area, or by introducing an additional parameter equal to the smaller hydraulic conductivity value. If the geohydrology of the ground-water flow system indicates that the hydraulic conductivity is applicable to parts of other model layers, they can be included in the definition.

For model inputs read as arrays by MODFLOW, the model input can be calculated as the sum of contributions from more than one parameter. As a simple example, consider the situation illustrated in figure 3. Here, it is assumed that the model input varies linearly along columns of the finite-difference grid. This variation can be represented by assigning one parameter to be equal to the model input value at row 1, and another parameter to be equal to the model input value at row 10. If the product of all multipliers for the first parameter equals 1.0 at row 1 and linearly decreases to 0.0 at row 10, and the product of all multipliers for the second parameter equals 0.0 at row 1 and linearly increases to 1.0 at row 10, the desired variation and parameter definition is obtained. Note that the values of parameters defined along the two rows of the grid are interpolated to all cells. For two-dimensional variation, two-dimensional finite-element basis functions (Segerlind, 1976) or kriging (Keidser and others, 1990) can be used to calculate multiplication arrays that perform the desired interpolation. For kriging, a fitted or assumed variogram is used, and parameters can be defined as discussed in Appendix A. For three-dimensional variation, three-dimensional finite-element basis functions (Segerlind, 1979) can be used. Other interpolation techniques, such as polynomial interpolation and cubic splines, can also be used.

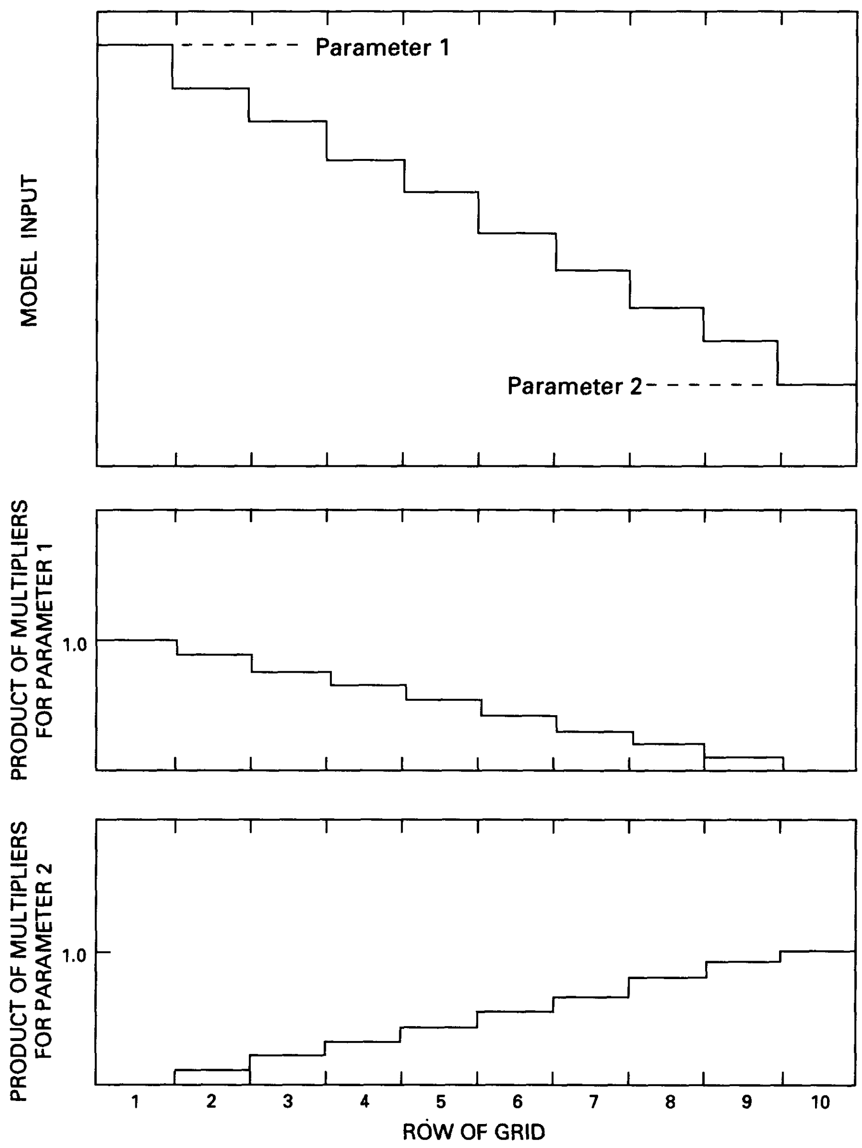


Figure 3.--Multipliers needed to define a discretized linear variation in a model input using two parameters.

The formulation provided for spatial parameterization allows a great deal of flexibility in defining parameters. It is hoped that this will allow users to define parameters that make the most physical sense in a given situation, and to realistically represent spatial variability while minimizing the number of parameters estimated. Note that any defined spatial variability is assumed to be known, and all uncertainty is related to the estimated parameters. If the spatial variability is not truly known, which is common, the results of the nonlinear regression might indicate unrealistically reliable model results.

The temporal parameterization permitted in the Parameter-Estimation Package is limited to making parameter values equal for specified time steps; the time steps can be from different stress periods and might or might not be continuous. In a sense, this is zonation applied temporally.

Some estimated parameters can be redefined by a natural-log transformation. Redefining the parameter in this way sometimes produces a better-conditioned regression problem (Carrera, 1984), always ensures that the parameter value remains positive, and, if the parameter is assumed to be lognormally distributed, allows more convenient statistics to be used to characterize the reliability of prior and final estimates. Hydraulic-conductivity estimates commonly are considered to be lognormally distributed because point hydraulic-conductivity measurements have been determined to be lognormally distributed in some geohydrologic situations (Davis, 1969, p. 76; Nielsen and others, 1973; Freeze, 1975; Neuman, 1982; Sudicky, 1986).

Some example lognormal distributions are shown in figure 4. Discussions of the lognormal distribution are available in Benjamin and Cornell (1970, p. 262-270), and Schmittroth (1979); the reader is referred to these sources for its characteristics. Here, note that if K is a lognormally distributed parameter, $\exp[E(\ell n\ K)]$, equals the modal value of K, and $\exp[E(\ell n\ K)+\sigma^2/2]$ equals E(K), the mean of K, where σ^2 is the variance of ℓn K. The exponential of confidence intervals on $E(\ell n\ K)$, therefore, are confidence intervals on the modal value of K, and might not be symmetric about $\exp[E(\ell n\ K)]$. The model inputs associated with parameters that can be redefined by a natural-log transformation are noted in table 1.

Users who prefer $\log_{10}$ transformation can convert to $\log_{10}$ by multiplying the log-transformed parameter value estimated by the Parameter-Estimation Package by $\log_{10} e = 0.4342945$. This is derived by noting that a parameter, b, can be expressed as $b = e^{\ell n\, b}$. Then, $\log_{10} b = \ell n\, b \log_{10} e$.

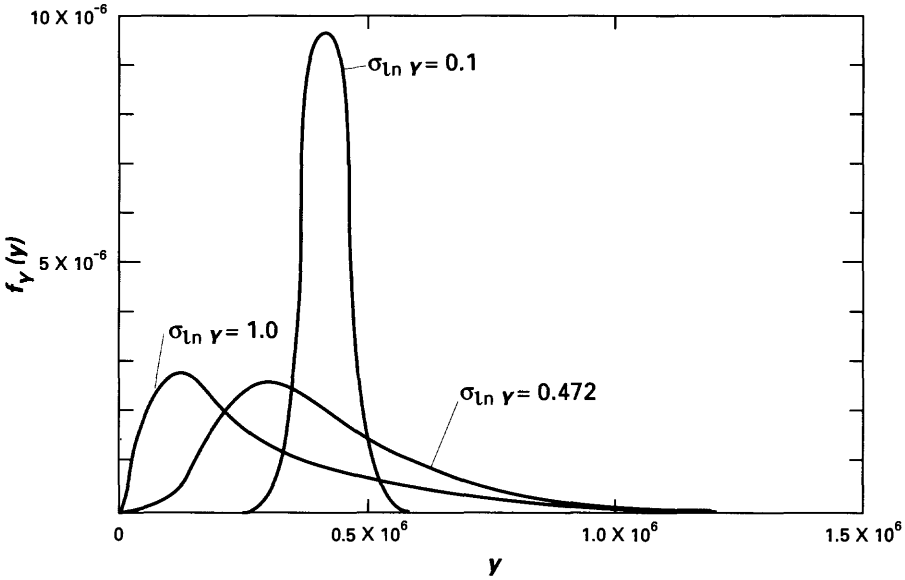


Figure 4.--Lognormal distributions with the same mean showing the effect of $\sigma_{\ell nY}$, where ℓnY is a normally distributed random variable. [Modified by Benjamin and Cornell (1970); used with permission of the publisher.

OBSERVATIONS

Several types of commonly available observations of dependent variables can be used to estimate parameters, including observations of hydraulic head or temporal changes in hydraulic head at arbitrary times and locations, observations of hydraulic head or temporal changes in hydraulic head averaged over several model layers, and observations of gains and losses along head-dependent boundaries, such as streamflow gains and losses. To use these observations to estimate parameters by nonlinear regression, corresponding simulated values, $\hat{y}$, need to be calculated so that the difference between observed and simulated values can be evaluated and minimized. Calculation of $\hat{y}$ for the different types of observations is described in the following sections; use of $\hat{y}$ in nonlinear regression is described later in the text.

Calculation of $\hat{y}$ needs to be modified if water-table cells involved in the calculation 'go dry'. This problem also is discussed in this chapter.

Hydraulic Head at One Location Observed Over Time

Hydraulic heads can be observed for any time represented by the simulation. If the observation occurs within a time step, $\hat{y}$ is calculated by linearly interpolating values calculated at the beginning and end of the time step using TOFF, a coefficient specified by the user (Appendix A). TOFF equals the time of the observation minus the time at the beginning of the time step divided by the length of the time step. Use of TOFF is illustrated in figure 5.

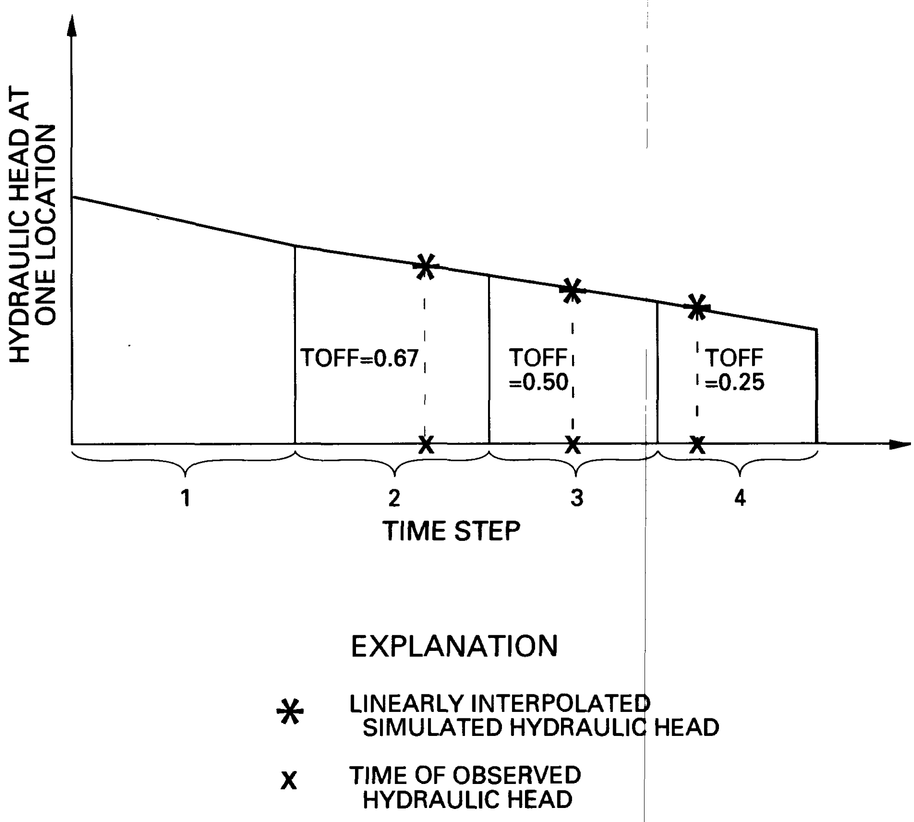


Figure 5.--Use of TOFF to calculate simulated values of hydraulic heads observed within time steps by linear interpolation.

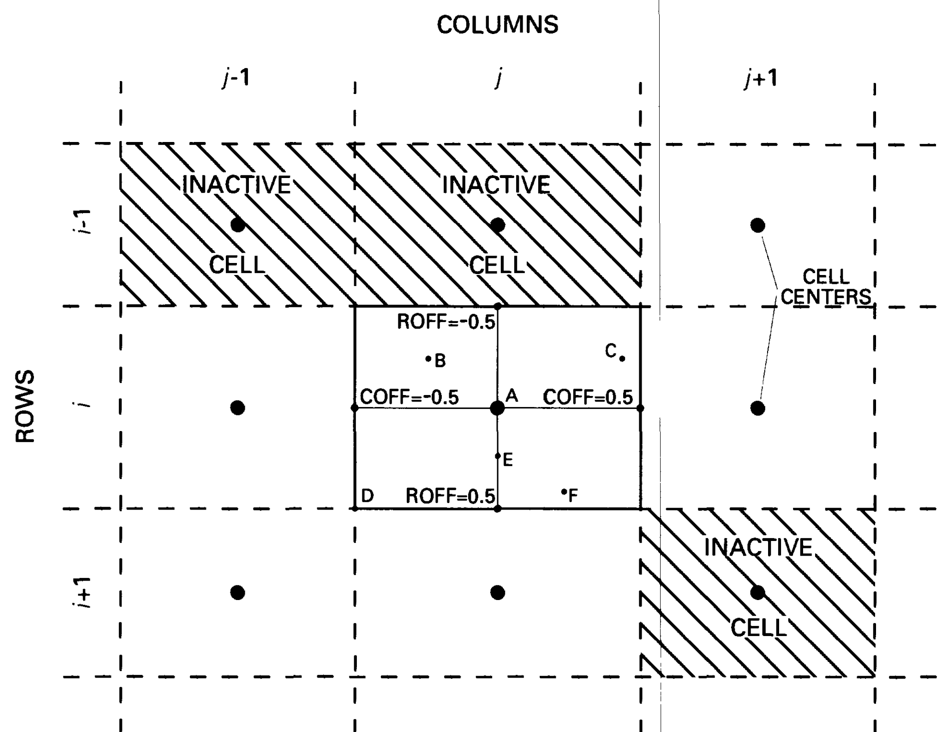
Hydraulic heads observed at one location over time can be represented in the Parameter-Estimation Package as hydraulic heads at each time of observation, or as an initial hydraulic head followed by changes in hydraulic head. The second representation permits some types of correlation in the true errors to be accurately and easily represented, as described later in this report in the section "Weighting Observations".

Hydraulic Head at Arbitrary Locations

The finite-difference method calculates hydraulic heads at the center of each active finite-difference cell within a layer. Observation wells, however, rarely are located at cell centers and might not be screened throughout the entire thickness represented by the model layer. In this report, hydraulic heads are assumed to be equal through the thickness of each model layer, so variations caused by limited screening of the observation well within a layer are ignored. Simulated hydraulic heads at observation locations are calculated by interpolating within the two-dimensional plane of a single layer. Six locations (A-F) for which hydraulic heads might need to be interpolated are shown in figure 6. Location A is exactly in the center of a cell, so no interpolation would be needed. Location E is exactly between two cell centers, so interpolation using two hydraulic heads would be needed. Hydraulic heads at all other locations would require interpolation using three or four hydraulic heads.

Exact interpolation of hydraulic heads is not always possible for the finite-difference method described by equations (1) and (2) because hydraulic properties that are defined for cells do not extend between locations where hydraulic head is calculated. For example, interpolation for locations C, D, or F in figure 6 could require as many as four different hydraulic-conductivity values, and, for this complicated case, no exact interpolation method is available.

Approximate geometric interpolation methods that exclude the variations in hydraulic conductivity are available. In this report, geometric interpolation based on linear, finite-element basis functions is used. Linear one-dimensional basis functions (equivalent to linear interpolation) are used for locations such as B and E in figure 6, which are adjacent to two inactive cells or are exactly between adjoining cell centers; triangular basis functions are used for locations such as C and F in figure 6, which are within a triangle formed by the centers of three neighboring cells because the fourth neighboring cell is inactive; and quadrilateral basis functions are used for locations such as D in figure 6, which are within a rectangle formed by the centers of four active cells. All basis functions are calculated using local coordinates that are specified by the user and define where the arbitrary location is within a cell relative to the location of the cell center. These local coordinates are a row offset, ROFF, and a column offset, COFF, and their use is illustrated in figure 6. Note that ROFF is negative in the direction of decreasing row numbers, and COFF is negative in the direction of decreasing column numbers.



POINT	ROFF	COFF
A	0.0	0.0
B	-0.25	-0.25
C	-0.25	0.4
D	0.5	-0.5
E	0.25	0.0
F	0.4	0.25

Figure 6.--Locating points within a finite-difference cell using ROFF and COFF.

The basis functions used are described in numerous texts and are not discussed in this report. They are equivalent to the one-dimensional simplex, two-dimensional simplex, and quadratic-element basis functions of Segerlind (1976, p. 24, 28, and 258), and the triangular "archetypal" and rectangular-element basis functions of Wang and Anderson (1982, p. 119 and 153). Wang and Anderson (1982) do not discuss a linear, one-dimensional basis function.

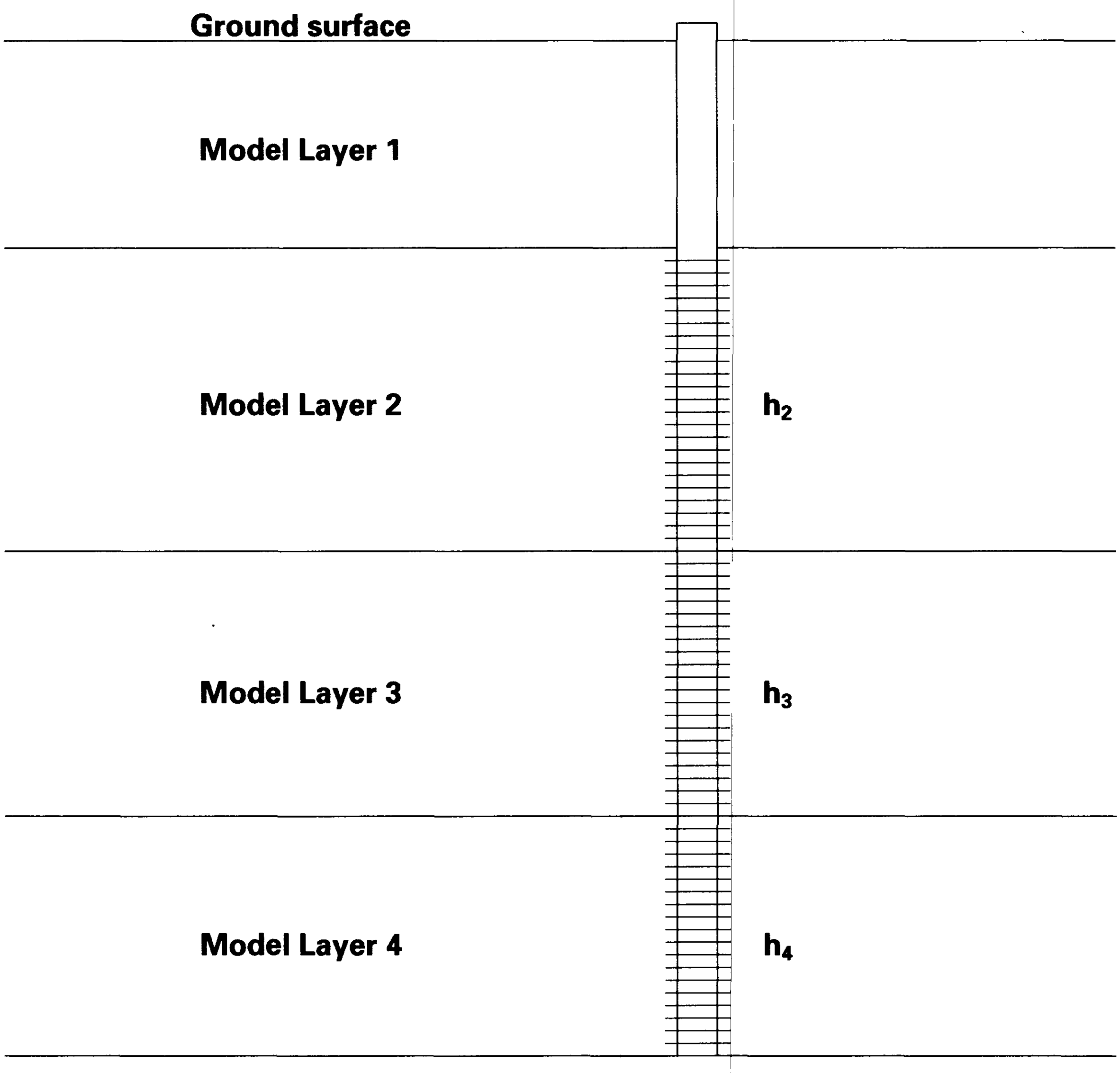
Errors introduced by using geometric interpolation might become substantial when the hydraulic properties of neighboring cells are different and cell dimensions are large. At such locations, the differences between observed and simulated hydraulic heads might be inaccurate and could produce inaccurate parameter estimates. This problem would be characterized by larger than expected differences between observed and simulated hydraulic heads.

Although this section only discusses observed hydraulic heads, identical procedures are used when an initial hydraulic head followed by temporal changes in hydraulic head are used.

Multilayer Hydraulic Heads

If an observation well is screened over intervals that represent more than one model layer, and the observed hydraulic head or change in hydraulic head is affected by all screened intervals, then the associated simulated value is a weighted average of the hydraulic heads or changes in hydraulic head calculated for each of the layers involved. The simulated value is calculated by multiplying the hydraulic head or change in hydraulic head in each layer by a user-specified proportion and then summing the results, as shown in figure 7. A more realistic representation of this problem would be produced by calculating the proportions based on the flow-system and aquifer characteristics, but the Parameter-Estimation Package does not do this.

Interpolation for multilayer hydraulic heads can be complicated because neighboring cells can be active or inactive, depending on the layer. In the Parameter-Estimation Package, the interpolation is defined using the IBOUND array (McDonald and Harbaugh, 1988, p. 4-2) of the first layer listed for the multilayer hydraulic-head observation (see DATA SET 6A of the INPUT



$$\hat{y} = p_2 h_2 + p_3 h_3 + p_4 h_4$$

h_2, h_3, and h_4 are calculated hydraulic heads at the observation location in layers 2, 3, and 4.

p_2, p_3, and p_4 are proportions defined by the user.

Figure 7.--Calculating the simulated value of hydraulic head for a multilayer observation well.

FILE, Appendix A). Thus, for each neighboring cell that is inactive in any of the other model layers, the cell in the same row and column in the first layer listed needs to be active. If no one layer contains a complete set of inactive cells, correct interpolation cannot be accomplished. This is illustrated in figure 8.

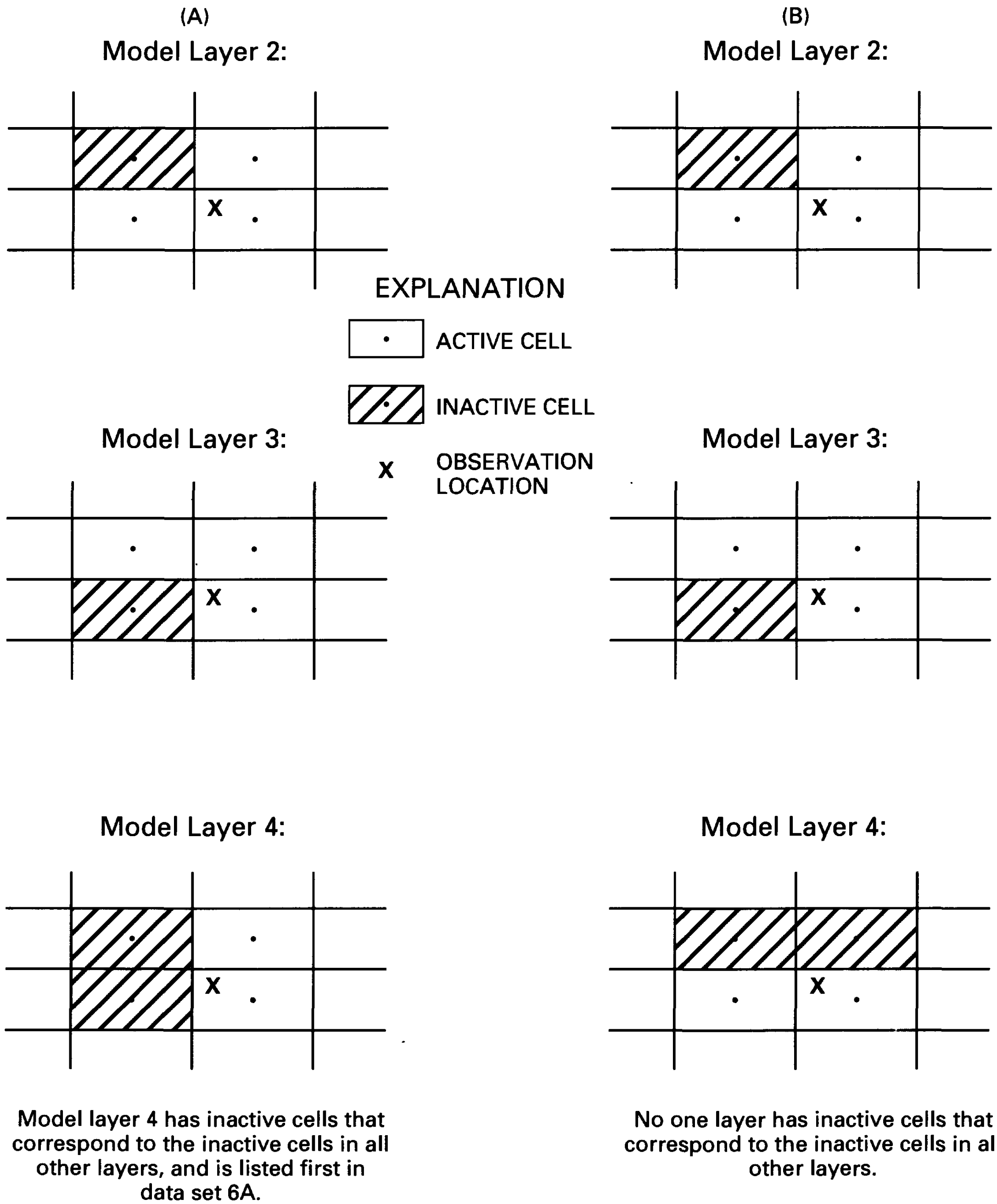


Figure 8.--Situations that do (A) and do not (B) produce correct spatial interpolation for multilayer hydraulic-head observations.

Gains and Losses Along Head-Dependent Boundaries

Head-dependent boundaries correspond to, for example, rivers, drains and lakes, and can be simulated using the River, Drain, or General-Head Boundary Packages (McDonald and Harbaugh, 1988, ch. 6, 9, and 11), or the Streamflow-Routing Package (Prudic, 1989). Flows into or out of any part of a head-dependent boundary for which gains and losses have been observed are calculated as:

$$\hat{y} = \sum_{i=1}^{NQCL} c_i q_i = \sum_{i=1}^{NQCL} c_i \frac{K_i}{D_i} A_i (H_i - h_i), \qquad (4)$$

where NQCL is the number of finite-difference cells (number of reaches for the Streamflow-Routing Package) used to simulate that part of the boundary;

q_i is the simulated flow rate at one cell (L^3/T) (negative for flow out of the aquifer);

c_i is a user-defined multiplicative factor;

K_i is the cell hydraulic conductivity (L/T) of, for example, the riverbed or lakebed;

D_i is the cell thickness (L) of, for example, the riverbed or lakebed;

A_i is the area of the water body within the finite-difference cell (L^2);

h_i is the simulated hydraulic head in the ground-water system adjacent to the head-dependent boundary (L); and

H_i is the water level in the water body or the elevation of the drain.

The observed equivalent of $\hat{y}$, y, would equal Q_2-Q_1, as shown in figure 9A; the related components of equation (4) are shown in figures 9A and 9B. Generally $c_i = 1.0$. However, if Q_1 or Q_2 or both are located within a cell instead of the edges of the cells as in figure 9A, c_i needs to be less than 1.0 so that only part of the flow calculated for the cell is included in $\hat{y}$.

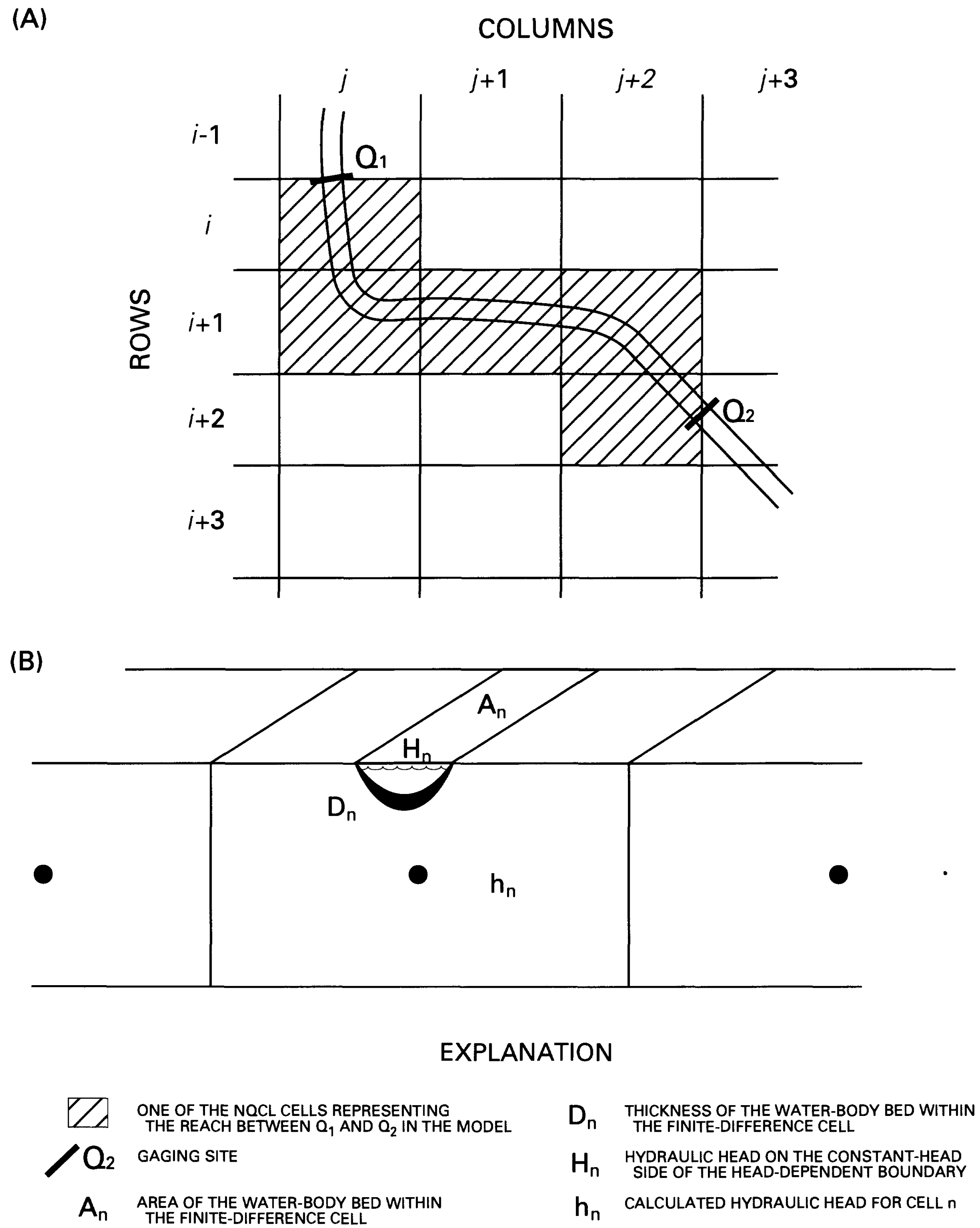


Figure 9.--Representation of head-dependent boundary gain and loss observations: A, gaging sites and cells used to represent the reach between gaging sites in the model; and B, quantities used to calculate the simulated gain or loss.

There are exceptions to equation (4) in some of the model packages. In the River and Streamflow-Routing Packages, if h_i declines below the bottom of the riverbed or streambed, the flow at cell i is calculated as (figure 10A and 10C):

$$q_i = \frac{K_i}{D_i} A_i (H_i - E_i) \qquad (5)$$

where E_i is the elevation of the bottom of the streambed. In the Drain Package, if h_i declines below the bottom of the drain, $q_i=0$ (no flow is simulated between the ground-water system and the drain, as shown in figure 10B). The Streamflow-Routing Package also has the exception that if the calculated loss from a reach exceeds the flow into that reach, the loss is set equal to the flow into the reach.

The exceptions noted above result in the elimination of h_i from the calculated head-dependent boundary gains and losses, and decrease the importance of the observed gain or loss to the estimation of parameters. In the extreme, an observation might have no affect on parameter estimation, in which instance the observation is omitted for the parameter-estimation iteration being executed. Messages are printed in the output from the Parameter-Estimation Package when the exceptions listed above occur and when an observation is omitted.

Dry Cells in Water-Table Layers at Observation Locations

When the hydraulic head at a cell in a water-table layer declines below the bottom of the aquifer, the cell is designated as inactive and remains inactive through the last time step (McDonald and Harbaugh, 1988, p. 5-9). Such cells are said to 'go dry'. At a dry cell, hydraulic head is not calculated, and the cell cannot be used to calculate simulated hydraulic heads or head-dependent boundary gains and losses for the parameter-estimation iteration. For head-dependent boundary reaches this generally poses little problem because cells along the reaches do not tend to go dry as often as other cells. When they do go dry, these cells usually account for only a fraction of a reach. No special provisions have been made in the Parameter-Estimation Package to account for cells going dry along head-dependent boundary reaches with observed gains or losses.

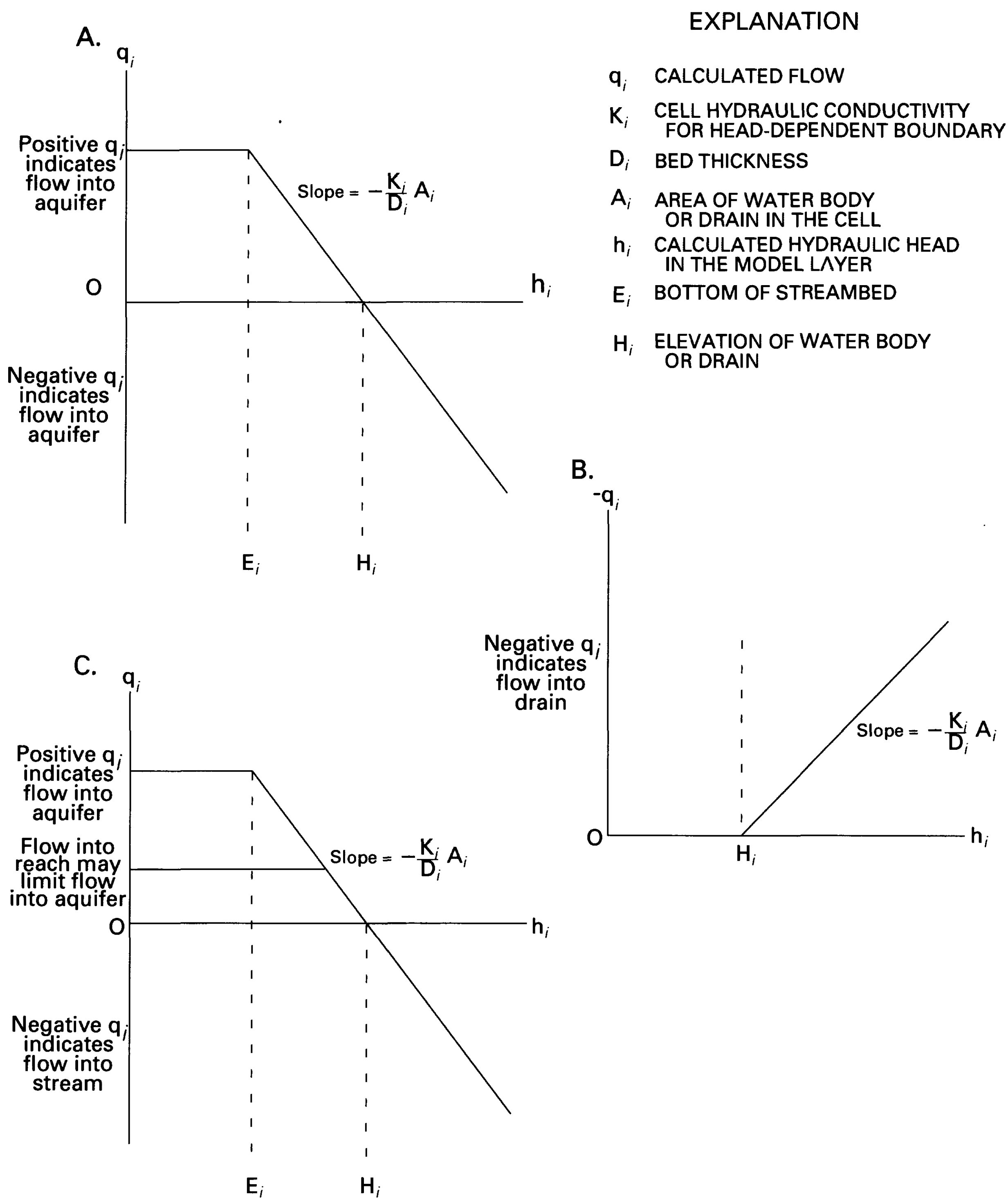


Figure 10.--The dependence of simulated gains and losses on hydraulic head in the model layer (h_i) in: A, the River Package, B, the Drain Package, and C, the Streamflow-Routing Package.

Problems are more severe when cells go dry at or adjacent to hydraulic-head observation locations. There are three types of problems, as shown in figure 11. First, if the observation is single layer and an adjacent cell which is used in the interpolation method discussed earlier goes dry, the dry cell usually can be omitted from the interpolation without introducing too much error into the interpolated value. This procedure was adopted in the Parameter-Estimation Package. Second, if the observation is multilayer and cells used for interpolation in one or more layers go dry, the proportions used to weight the hydraulic heads from those layers probably are no longer valid. Although the cells could be omitted from the interpolation for the layers involved and a simulated hydraulic head analogous to the observed value could be calculated, the problem with the proportions can not easily be resolved. In the Parameter-Estimation Package, multilayer observations for which any cells used in the interpolation go dry are omitted from the parameter-estimation procedure. Third, if the observation is single layer or multilayer and the cell containing the observation location goes dry in any of the layers involved, the observation is omitted from the parameter-estimation procedure. The effect of omitting the observations for the last two situations is that the impetus for changing the parameters to keep the dry areas wet is lost from the parameter-estimation procedure. This loss is unfortunate, but, at this point, no practical alternative exists.

As discussed previously in the section "Ground-water flow equation", the Parameter-Estimation Package reactivates all dry cells at the beginning of each parameter-estimation iteration, so that the original interpolation and number of observations are reinstated.

Omitted Observations

If observations are alternately used and omitted and used again in successive parameter-estimation iterations, which might occur for hydraulic-head observations in water-table layers that go dry and for head-dependent boundary gain and loss observations, parameter estimation might not converge (see the section "The Sum-of-Squares Objective Function in Nonlinear Problems" for a discussion of parameter-estimation iterations and convergence). This problem can be addressed in the following ways:

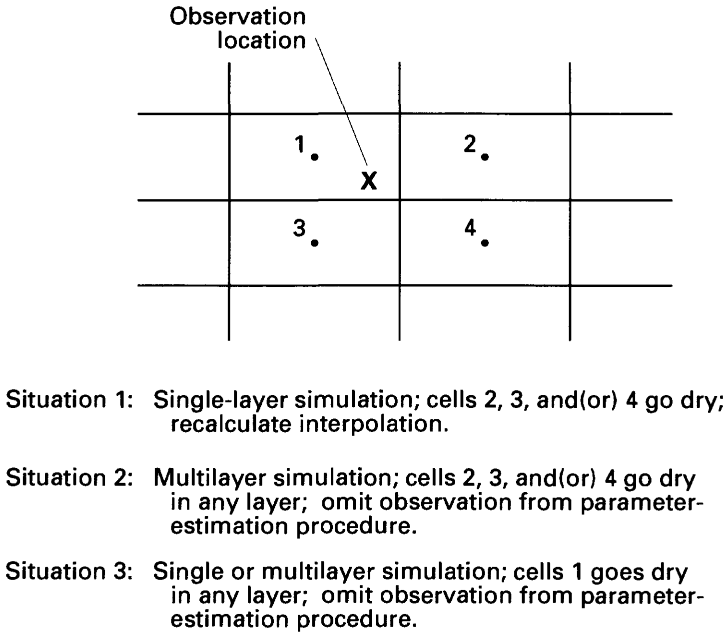


Figure 11.--Effect of dry cells on calculated hydraulic head at an observation location.

1. Eliminate the omitted observations during initial parameter estimation iterations or early in the calibration process, and try including them later when the parameter estimates are closer to the final values or the model is closer to its final form.
2. A water-table layer can be simulated as a confined layer using estimated layer thicknesses early in the calibration process, and represented as a water-table layer later when the parameter estimates are closer to the final values or the model is closer to its final form.
3. For head-dependent boundary gain-and-loss observations, small streambed or riverbed thicknesses can aggravate the problem. Increase these thicknesses if such a change is realistic.
4. Review the method used to represent the ground-water flow system and make changes if needed. This is the same process that a modeler goes through in a trial-and-error calibration, and its goal is to ensure that the physical system is being represented realistically. Unrealistic representations cause problems in nonlinear-regression parameter estimation just as they cause problems when calibrating by trial and error.

REVIEW OF PARAMETER ESTIMATION AND ANALYSIS OF RESULTS USING LINEAR REGRESSION

A brief review of linear regression is included to provide an easier framework within which to present the concepts. Although this review is brief, extensive discussions are available in Draper and Smith (1981) and Cooley and Naff (1990), from which much of the following discussion is condensed. For the less advanced reader, Ott (1988) provides a more elementary discussion.

Assumed Linear Model

In any type of regression procedure, a model structure needs to be assumed. In linear regression, the assumed model can be expressed as a linear function of the parameters. This assumed model is correct if the true process is a linear function of the parameters. For example,

$$\tilde{y} = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_{NP} x_{NP} \tag{6}$$

where $\tilde{y}$ is the true calculated dependent variable (analogous, for example, to hydraulic head in the ground-water flow equation);

NP is the number of parameters;

$\beta_1 \beta_2 \cdots \beta_{NP}$ are true, unknown parameter values; and

$x_1, x_2 \cdots x_{NP}$ are independent variables (analogous to spatial dimensions and time in the ground-water flow equation). These might be functions of dependent variables in some applications.

The parameters $\beta_1, \beta_2 \cdots \beta_{NP}$ are unknown, but can be estimated from observations, y_q, which are observed at different values of the independent variables. The data would be as follows:

$$(y_1, x_{1,1}, x_{1,2}, \cdots x_{1,NP}),\ (y_2, x_{2,1} x_{2,2} \cdots x_{2,NP}),\ \cdots\ (y_{ND}, x_{ND,1}, x_{ND,2} \cdots x_{ND,NP}), \tag{7}$$

where the added subscripts indicate the sequential number of the observation, and ND is the total number of observations. Note that the first subscript on each independent variable, $x_{q,i}$, is the observation number, and the second is the number of the parameter it multiplies. These data can be substituted into the model (eq. 6) to produce ND equations of the form

$$y_q = \tilde{y}_q + \epsilon_q = \beta_1 x_{q,1} + \beta_2 x_{q,2} + \cdots \beta_{NP} x_{q,NP} + \epsilon_q \qquad q = 1,ND \quad (8)$$

where ϵ_q is the difference between observation y_q and the surface predicted by the true model, and is called the true error or disturbance.

The ND equations of (8) can be expressed using vectors and matrices as:

$$\underline{y} = \underline{\underline{X}}\ \underline{\beta} + \underline{\epsilon} \qquad (9)$$

where

$$\underline{y} = \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{ND} \end{Bmatrix}, \quad \underline{\underline{X}} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,NP} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,NP} \\ & & \vdots & \\ x_{ND,1} & x_{ND,2} & \cdots & x_{ND,NP} \end{bmatrix}, \quad \underline{\beta} = \begin{Bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{NP} \end{Bmatrix},$$

$$\underline{\epsilon} = \begin{Bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_{ND} \end{Bmatrix}.$$

The elements of $\underline{\epsilon}$ are assumed to be random variables, and the validity of the regression procedure depends on their statistical properties. It is generally assumed that (1) $E(\underline{\epsilon}) = \underline{0}$, so that the model is unbiased (Benjamin and Cornell, 1970, p. 380) and (2) the variance-covariance matrix of the errors is $\underline{\underline{V}}(\underline{\epsilon}) = \underline{\underline{I}}\ \sigma^2$, so that the elements of $\underline{\epsilon}$ are uncorrelated and have equal variances (Draper and Smith, 1981, p. 108). Violations of the second assumption need to be accounted for by using a weight matrix, as discussed in the section "Weighting Observations".

Parameter Estimation by Least-Squares Regression

The goal of regression is to determine estimates, $\underline{b}$, of the true, unknown parameter values, $\underline{\beta}$, so that an assumed model that approximates the true model produces calculated values, $\hat{y}_q$, at the observation points q, which are similar to the measured values, y_q, as measured by an objective function. Replacing $\underline{\beta}$ with $\underline{b}$ in equation (9) yields:

$$\underline{y} = \underline{\underline{X}}\ \underline{b} + \underline{e} = \hat{\underline{y}} + \underline{e} \tag{10}$$

where $\hat{\underline{y}}$ is a vector of calculated values of y and $\underline{e}$ is a vector of residuals. The variable $\underline{e}$ is used instead of $\underline{\epsilon}$ to clearly indicate that the components of $\underline{e}$ are not true errors. The elements of $\underline{e}$ are random variables and have statistical properties that are consistent with the statistical properties of the elements of $\underline{\epsilon}$ if the assumed model is similar to the true model.

Two objective functions commonly are used in ground-water flow parameter-estimation problems, but both reduce to the sum-of-squares objective function for any single run of the parameter-estimation procedure. The maximum-likelihood objective function is discussed in the section "Parameter Estimation and Analysis of Results Using Nonlinear Regression".

The sum-of-squares objective function is defined as:

$$\begin{aligned} S(\underline{b}) &= \sum_{q=1}^{ND} [y_q - \hat{y}_q]^2 \\ &= \sum_{q=1}^{ND} [y_q - (b_1 x_{q,1} + b_2 x_{q,2} + \cdots + b_{NP} x_{q,NP})]^2 \\ &= [\underline{y} - \underline{\underline{X}}\ \underline{b}]^T [\underline{y} - \underline{\underline{X}}\ \underline{b}] \\ &= \underline{e}^T \underline{e}, \end{aligned} \tag{11}$$

where $S(\underline{b})$ is a scalar, and $\underline{e}$ is the vector of residuals. A regression problem solved using a sum-of-squares objective function is called a least-squares regression problem.

A plot of a sum-of-squares objective-function surface for a problem with two parameters, b_1 and b_2, is shown in figure 12. For linear problems, the contours are elliptic, and the minimum $S(\underline{b})$ value usually is well-defined and unique. The parameter values that produce the minimum objective-function value can be calculated by solving $\partial S(\underline{b})/\partial \underline{b} = \underline{0}$, which is equivalent to:

$$\frac{\partial S(\underline{b})}{\partial b_1} = 0, \qquad \frac{\partial S(\underline{b})}{\partial b_2} = 0, \qquad \cdots\cdots \qquad \frac{\partial S(\underline{b})}{\partial b_{NP}} = 0. \tag{12}$$

The minimum is called a stationary point because all the derivatives in equation (12) equal zero. Objective functions of nonlinear models can have other types of stationary points, such as maxima or inflection points (Hildebrand, 1976, p. 356-362), or relative and absolute minima. Various kinds of stationary points, in addition to alternative terminology that commonly is used to describe them, are shown in figure 13. A problem with a well-defined, unique minimum and no other stationary points on the objective-function surface is said to be well-conditioned. Although this concept can be graphically displayed only in a one- or two-dimensional parameter space, it applies for any number of parameters.

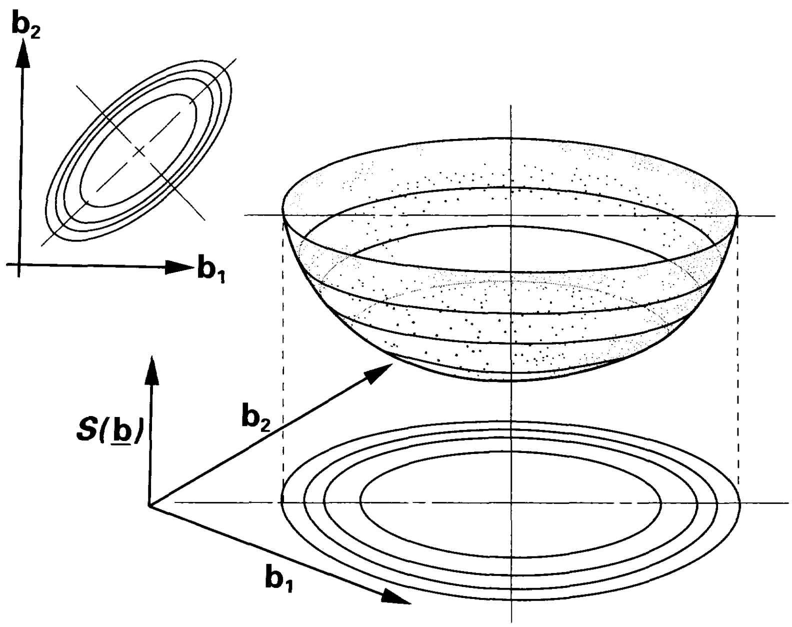


Figure 12.--Sum-of-squares objective-function surface for a linear problem problem with parameters b_1 and b_2 (modified from Himmelblau, 1972, p. 80; used with permission of the publisher).

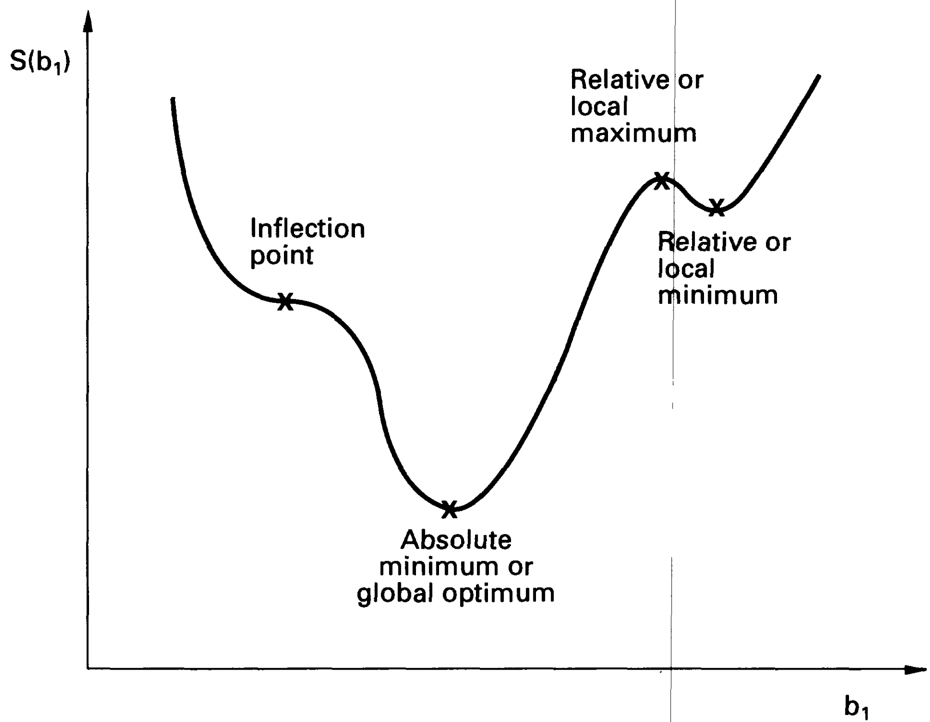


Figure 13.--Stationary points on an objective-function surface, labeled using common terminology (modified from McLaughlin, 1975, p. III-4).

In linear regression, the sensitivities, which are defined as the derivative of the calculated dependent variable, $\hat{y}$, with respect to the parameters, are required to calculate the $\underline{b}$ that satisfies equation (12). The linear model is easy to use because the sensitivities are independent of the parameter values. From equation (6):

$$\frac{\partial \hat{y}}{\partial \beta_1} = x_1, \quad \frac{\partial \hat{y}}{\partial \beta_2} = x_2, \quad \cdots \quad \frac{\partial \hat{y}}{\partial \beta_{NP}} = x_{NP}. \tag{13}$$

Substituting the parameter estimates, $\underline{b}$, for the true parameters, $\underline{\beta}$, sensitivities at observation-point q equal:

$$\frac{\partial \hat{y}_q}{\partial b_1} = x_{q,1}, \quad \frac{\partial \hat{y}_q}{\partial b_2} = x_{q,2}, \quad \cdots \quad \frac{\partial \hat{y}_q}{\partial b_{NP}} = x_{q,NP}. \tag{14}$$

Note that $\underline{\underline{X}}$ of equations (9) and (10) is a matrix of sensitivities, and for linear problems simply equals the values of $x_{q,1}$, $x_{q,2}$, •••. When the model is nonlinear in the parameters, such as in most ground-water flow problems, $\underline{\underline{X}}$ is still a matrix of sensitivities and has the same function in the following equations. The sensitivities, however, are not as easy to calculate and they are dependent on the parameter values.

To proceed, consider a linear problem with a two-parameter model equation, and $x_{q,1}=1$, $q=1,ND$, so the assumed model is a straight line with intercept b_1 and slope b_2. Equation (12) can be solved to produce the following well-known equations, called the normal equations, for b_2 and b_1:

$$b_2 = \frac{\Sigma x_{q,2}y_q - [(\Sigma x_{q,2})(\Sigma y_q)]/ND}{\Sigma x_{q,2}^2 - (\Sigma x_{q,2})^2/ND}, \tag{15}$$

$$b_1 = (\Sigma y_q)/ND - b_2(\Sigma x_{q,2})/ND,$$

where the summations are for the ND observations.

Using the matrix notation of equation (9), equation (15) is the solution of:

$$\underline{\underline{X}}^T\underline{\underline{X}}\ \underline{b} = \underline{\underline{X}}^T\ \underline{y} \tag{16}$$

where, for the simple linear problem with two parameters,

$$\underline{\underline{X}} = \begin{bmatrix} 1 & x_{1,2} \\ 1 & x_{2,2} \\ \vdots & \vdots \\ 1 & x_{ND,2} \end{bmatrix} .$$

Then,

$$\underline{\underline{X}}^T\underline{\underline{X}} = \begin{bmatrix} ND & \Sigma x_{q,2} \\ \Sigma x_{q,2} & \Sigma(x_{q,2})^2 \end{bmatrix} \text{, and}$$

$$\underline{\underline{X}}^T\underline{y} = \begin{bmatrix} \Sigma y_q \\ \Sigma x_{q,2}y_q \end{bmatrix} ,$$

where, again, the summations are for the ND observations. By using these definitions, it can easily be verified that equations (15) and (16) are equivalent. Regardless of the dimensions or linearity of the problem, all least-squares regression problems are solved using normal equations of the form of equation (16).

Weighting Observations

If the observations used in a regression analysis are (1) not equally reliable, (2) have different units, or (3) have true errors that are correlated, it cannot be assumed that the true errors have the statistical properties required to produce a valid regression even if the assumed model is correct--that is, although $E(\underline{\epsilon}) = \underline{0}$ is still valid, it cannot be assumed that the variance-covariance matrix of the true errors $\underline{\underline{V}}(\underline{\epsilon})$, satisfies $\underline{\underline{V}}(\underline{\epsilon}) = \underline{\underline{I}}\sigma^2$ (Draper and Smith, 1981, p. 108). As a result, the residuals used in the regression procedure need to be weighted.

Before proceeding, consider the three conditions for which weighting is required and the function the weights have in the regression for each of these conditions. Observations might not be equally reliable if, for example, they are observed with varying degrees of accuracy. In ground-water flow problems, this variability might occur if the elevations of some observation wells were determined by standard surveying methods to within a few hundredths of a foot, whereas the elevations of other observation wells were determined by an altimeter or from a topographic map to within a few feet. When performing regression with such observations, it is more important for hydraulic-head residuals to be smaller at locations with more accurate elevations than at locations with less accurate elevations, and weights are assigned that indicate this variation in importance. Observations have different units if, for example, both hydraulic heads (L) and streamflow gains and losses (L^3/T) are observed. Weights are used to indicate that a difference of, for example, 1.0, between the observed and simulated values for the two types of observations are not equivalent, to indicate variations in observation accuracy, or both. True errors would be correlated if, for example, hydraulic head at the same well was observed at many times, and errors in determining the elevation or position of the well were common to all observations. Weighting then is used to indicate that the errors associated with those observations are correlated.

It is important to note that model error generally cannot be represented in the weight matrix. If the form and affect of model error can be evaluated, the data can be adjusted accordingly. For example, if a well penetrates only part of a pumped aquifer and the entire aquifer is represented as a model layer, the affects of partial penetration can be evaluated and the observations corrected to reflect full penetration. As the estimated transmissivity near the well changes during the calibration process, the correction can be recalculated.

Structure and Use of the Weight Matrix

Weighting is implemented by using a symmetric weight matrix, $\underline{\underline{\omega}}$, that ideally would be calculated from the variance-covariance matrix of the true errors, $\underline{\underline{V}}(\underline{\epsilon})$, as:

$$\underline{\underline{\omega}} = \sigma^2 [\underline{\underline{V}}(\underline{\epsilon})]^{-1}, \tag{17}$$

where σ^2 is the user-defined common error variance of the true errors (see below). In practice, $\underline{\underline{V}}(\underline{\epsilon})$ is unknown. Its estimate, $\hat{\underline{\underline{V}}}(\underline{\epsilon})$, is estimated by the user, and the estimated weight matrix, $\hat{\underline{\underline{\omega}}}$, is calculated from $\hat{\underline{\underline{V}}}(\underline{\epsilon})$ as in equation (17). If $\hat{\underline{\underline{\omega}}}$ approximately equals $\underline{\underline{\omega}}$, the weighted true errors satisfy the desired conditions, $E(\hat{\underline{\underline{\omega}}}^{1/2}\underline{\epsilon}) = \underline{0}$ and $\underline{\underline{V}}(\hat{\underline{\underline{\omega}}}^{1/2}\underline{\epsilon}) = \underline{\underline{I}}\sigma^2$. If, in addition, the model is correct, σ^2 approximately equals the estimated error variance, s^2, which is calculated as the minimized value of the objective function divided by ND-NP (divide by ND+NPR-NP if there are prior parameter estimates; see "Prior Information on Estimated Parameters"). The square root of s^2 is called the standard error of estimate (Draper and Smith, 1981, p. 207). Discrepancies between σ^2 and s^2 indicated an incorrect model, an incorrect weight matrix, or biased observations; see "Adjustments Commonly Required During Parameter Estimation".

In weighted least-squares regression, the weighted residuals, $\hat{\underline{\underline{\omega}}}^{1/2}\underline{e}$, replace the unweighted residuals, $\underline{e}$. The weighted residuals might be dimensionless or might have the units of any of the dependent variables, depending on how the modeler defines σ^2. The weighted sum-of-squares objective function (eq. 11) is:

$$S(\underline{b}) = \underline{e}^T (\hat{\underline{\underline{\omega}}}^{1/2})^T \hat{\underline{\underline{\omega}}}^{1/2} \underline{e} = \underline{e}^T \hat{\underline{\underline{\omega}}} \underline{e}. \tag{18}$$

The weighted normal equations (eq. 16) are:

$$\underline{X}^T \hat{\underline{\omega}} \underline{X} \underline{b} = \underline{X}^T \hat{\underline{\omega}} \underline{Y}. \tag{19}$$

If all observations have the same units (for example, if all observations are hydraulic heads), the user can define a σ^2 such that the variance-covariance matrix of the true errors can be expressed as:

$$\underline{V}(\underline{\epsilon}) = \sigma^2 \underline{W}, \tag{20}$$

where $\underline{W}$ is an ND by ND matrix and either σ^2 or $\underline{W}$ is dimensionless. Then, by equation (17):

$$\underline{\omega} = \sigma^2(\sigma^{-2}\ \underline{W}^{-1}) = \underline{W}^{-1}. \tag{21}$$

The estimated matrix, $\hat{\underline{\omega}}$, would be of the same form. In this situation, the weight matrix and, thus, the parameter estimates produced by the regression routine are independent of the common variance, and the criterion that $\sigma^2 \approx s^2$ is always satisfied. If $\underline{W}$ is considered to be dimensionless, the units of the weighted residuals are the same as the units of the observations.

If there are observations with two different kinds of units (for example, if hydraulic heads and streamflow gains and losses are observed) and if it is assumed that the true errors of the two kinds of observations are statistically independent (this will be discussed in the section "Simplifying Assumptions"), the variance-covariance matrix of the true errors can be expressed as:

$$\underline{V}(\underline{\epsilon}) = \begin{bmatrix} \underline{W}\ \sigma_h^2 & \underline{0} \\ \underline{0} & \underline{Z}\ \sigma_f^2 \end{bmatrix}, \tag{22}$$

where $\underline{W}\ \sigma_h^2$ is the variance-covariance matrix of errors in observed hydraulic heads and temporal changes in hydraulic head ($\underline{\epsilon}_h$), and $\underline{Z}\ \sigma_f^2$ is the variance-covariance matrix for errors in observed head-dependent boundary gains and losses ($\underline{\epsilon}_f$). Generally, σ^2 is defined to be equal to either σ_h^2 or σ_f^2 or 1.0. If σ^2 equals σ_h^2, the weight matrix equals:

$$\underline{\omega} = \sigma_h^2(\underline{V}(\underline{\epsilon}))^{-1} = \begin{bmatrix} \underline{W}^{-1} & \underline{0} \\ \underline{0} & \underline{Z}^{-1}\dfrac{\sigma_h^2}{\sigma_f^2} \end{bmatrix}. \qquad (23)$$

Again, the estimated matrix, $\hat{\underline{\omega}}$, would be of the same form. The weight matrix and the parameter estimates produced by the regression routine are affected by the ratio σ_h^2/σ_f^2, and, if $\underline{W}$ is dimensionless, the weighted residuals have the same units as the hydraulic-head observations. If σ_f^2 is defined as the common error variance, the ratio σ_f^2/σ_h^2 affects the regression routine, and, if $\underline{W}$ is dimensionless, the weighted residuals have the same units as the head-dependent boundary gain-and-loss observations.

If σ^2 is set equal to 1.0 and is dimensionless, $\underline{\omega} = [\hat{\underline{V}}(\underline{\epsilon})]^{-1}$, and the residuals are dimensionless. In this situation, if the model and $\hat{\underline{V}}(\underline{\epsilon})$ are correct, s^2 is close to 1.0.

If the true errors are all uncorrelated, the nonzero entries in $\underline{V}(\underline{\epsilon})$, and therefore in $\underline{\omega}$ and $\hat{\underline{\omega}}$, occur on the diagonal. A diagonal weight matrix is conceptually and computationally simple, and only diagonal weight matrices are allowed in the Parameter-Estimation Package. However, a data transformation described in the following sections can be used so that some types of temporal correlations of the true errors can be included.

Estimation of the Variance-Covariance Matrix of the True Errors

In transient ground-water flow problems, various dependent variables might be observed at many locations, and they might be observed at many times. The most general variance-covariance matrix of the true errors would include the following correlations: (1) Correlations between errors in observations made at different locations at the same time; (2) correlations between errors in observations made at the same location at different times; and (3) correlations between errors in observations made at different locations and different times. Such a matrix would be impossible to estimate accurately, and laborious to estimate at all. Fortunately, assumptions about the true errors that are realistic for many circumstances can be made to simplify this variance-covariance matrix. Seven assumptions that result in a simple diagonal weight matrix are listed below and are further defined and discussed in the following section. Their validity also is noted in both parts of the text. This set of assumptions is not unique. They are presented so that one set of assumptions leading to a diagonal weight matrix can be thoroughly analyzed. The role of both measurement error and model error is included in the discussion because both commonly plague the development of ground-water flow models.

1. True error includes measurement errors and generally excludes model errors.
   - Required for a valid regression
2. Errors of different kinds of dependent variables are uncorrelated.
   - Commonly realistic for measurement errors
3. Errors of observations at different locations are uncorrelated.
   - Commonly realistic for measurement errors
4. Time-dependent deterministic components of the error are small and can be ignored.
   - Commonly realistic for measurement errors
5. At each observation location, total error for any observation = $\epsilon_1+\epsilon_2+\epsilon_3$.
   ϵ_1 = errors that are constant over time
   ϵ_2 = errors that are temporarily correlated, but not completely correlated like ϵ_1
   ϵ_3 = errors that are temporarily uncorrelated
   - Probably realistic
6. Errors are normally distributed.
   - Probably realistic
7. Either ϵ_1 or ϵ_2 or ϵ_3 errors dominate.
   - Commonly questionable

Simplifying assumptions

The first assumption is that the error includes measurement errors and generally excludes model errors. The sum of all measurement errors equals the ϵ_q of equation (8); model errors are any errors that could be corrected by changes in the model given greater computer capacity, more time, or more complete information about the ground-water flow system. Measurement errors include, for example, errors in the elevation of observation wells and errors in observed streamflow. Model errors are caused by, for example, inaccurate interpolation of calculated hydraulic heads and other problems discussed in the section "Observations" of this report, inability of the model to represent ground-water flows or fluctuations in ground-water properties that are smaller than the grid size, and parameterizations that artificially limit the spatial or temporal variability of parameters. The restrictions of the last example generally are necessary because only a limited number of parameters can be estimated, but they do produce model errors. The first assumption is required for a valid regression, and one of the goals during calibration is to reduce model error as much as possible. Model error generally cannot be accommodated using the weight matrix.

The second assumption is that errors in different kinds of dependent variables are uncorrelated. This is the assumption that was made to produce the weight matrix of equation (22), and indicates, for example, that errors in observed hydraulic heads are independent of errors in observed streamflow gains and losses. This assumption is realistic for measurement errors; for example, it is unlikely that errors incurred when observing hydraulic heads are related to errors incurred when observing streamflow. This assumption, however, might not be realistic for model errors; for example, simulated hydraulic heads and adjacent simulated streamflows generally would be affected by the same deficiencies in the model, and associated model errors would be correlated.

The third assumption is that errors in observations at different locations are uncorrelated. As with the second assumption, the third assumption is realistic for most measurement errors, but might not be realistic for model errors, especially at locations that are close to one another (Carrera, 1984, p. 37-39). One exception occurs for measurement errors of head-dependent boundary gains and losses when one flow measurement is used to calculate more than one gain or loss. For example, if flow is measured at three progressively downstream locations as Q_1, Q_2, and Q_3, the gain or loss measurements will equal Q_2-Q_1, and Q_3-Q_2. If the errors in Q_1, Q_2, and Q_3 are independent, normal, and have variances equal to σ_1^2, σ_2^2, and σ_3^2, the error in Q_2-Q_1 has a variance equal to $\sigma_1^2 + \sigma_2^2$, the error in Q_3-Q_2 has a variance equal to $\sigma_2^2 + \sigma_3^2$, and the covariance between the two gain-and-loss measurements equals $-\sigma_2^2$.

If valid, the result of the second and third assumptions is that all correlations are eliminated except for the temporal correlations at each location.

The fourth assumption is that time-dependent deterministic components of the error (Brockwell and Davis, 1987, p. 15) are small and can be ignored. Such components are more typical of model error than measurement error, so the fourth assumption probably is realistic for measurement errors.

The fifth assumption is that the remaining correlations of the true errors, which are temporal, can be categorized by thinking of the error associated with each observation at a single time and location as the sum of three statistically independent types of errors:

$$\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3, \tag{24}$$

where, ϵ_1 is constant for all time for each observation location, so the temporal correlation coefficients between this error and the errors associated with observations at other times at this location equal 1.0;

ϵ_2 varies with time and has correlation coefficients between this error and the errors associated with observations at other times at this location between 0.0 and 1.0, exclusive; and

ϵ_3 varies independently over time and has correlation coefficients between this error and the errors associated with observations at other times at this location equal to zero.

Examples of observations with these types of errors are shown in figure 14. Considering their definitions, the assumed independence of ϵ_1, ϵ_2, and ϵ_3 is realistic, and this method of characterizing errors probably is valid for all ground-water flow models. The autocorrelated temporal errors considered by Sadeghipour and Yeh (1984), Carrera and Neuman (1986a, p. 203), and Watson and others (1990a,b) would be classified as ϵ_2, as defined above. Lu and others (1988, p. 675) introduced a constant error similar to ϵ_1 to describe spatial correlations of errors in an estimated transmissivity field.

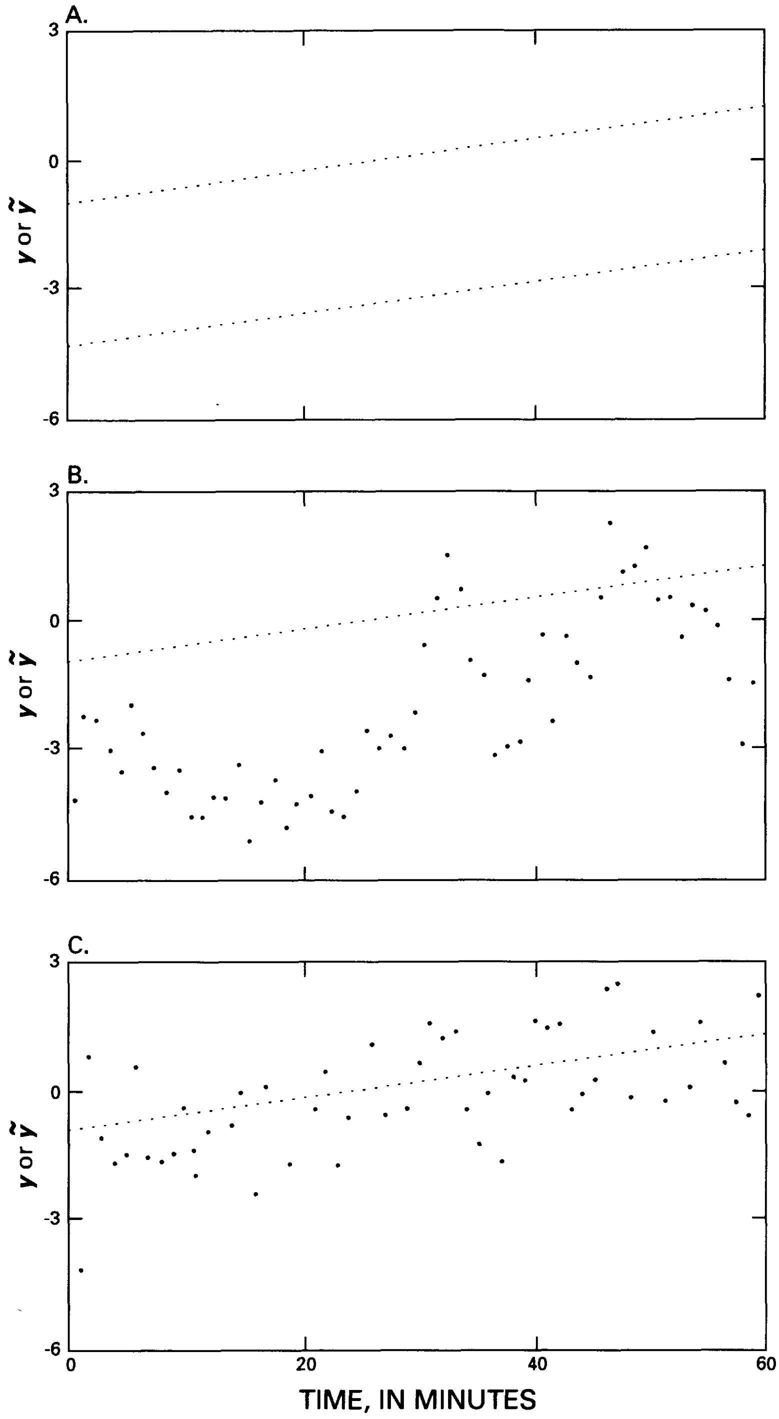


Figure 14. Paired observed and true values for which the errors, $y-\tilde{y}$, are (A) ϵ_1 errors, (B) ϵ_2 errors, and (C) ϵ_3 errors.

Processes that are likely to produce the three error types are different for measurement errors and model errors. For measurement errors, ϵ_1 might be the error in the measured elevation of the well; ϵ_2 might be an autocorrelated error in the recording device; and ϵ_3 might be random, uncorrelated inaccuracy in the recording device. From a practical viewpoint, ϵ_1 and ϵ_3 errors probably dominate measurement errors. For model errors, ϵ_1 might be the error produced because constant pumpage at a well near the observation location is not accounted for in the model; ϵ_2 might be the error produced because the parameterization of transmissivity is not realistic; and ϵ_3 errors probably would be small or equal to zero.

When written for a series of observations over time at an arbitrary observation location ℓ, equation (24) becomes:

$$\underline{\epsilon}_{\ell} = \underline{\epsilon}_{1\ell} + \underline{\epsilon}_{2\ell} + \underline{\epsilon}_{3\ell}, \qquad (25)$$

where the length of each vector equals the number of temporal measurements at observation location ℓ. If only temporal correlations are nonzero and the observations are ordered by location, $\underline{V}(\underline{\epsilon})$ is a block-diagonal matrix with the size of the blocks equal to the length of the vectors in equation (25), which might be different for different locations.

The variance-covariance matrices associated with the ϵ_1 and ϵ_3 errors have structures determined by their definitions. $\underline{V}(\underline{\epsilon}_{1\ell})$ is a matrix with all elements equal to the variance of the error. Thus, $\underline{V}(\underline{\epsilon}_{1\ell}) = \underline{1}\,\sigma^2_{1\ell}$, where $\underline{1}$ is a matrix of ones and $\sigma^2_{1\ell}$ is the error variance associated with observation location ℓ. $\underline{V}(\underline{\epsilon}_{1\ell})$ can be simplified if changes in the dependent variable over time are used in the regression instead of the actual values. For example, if hydraulic head is observed multiple times at one location, the first observation would be the initial hydraulic head, and subsequent observations would be changes from that initial hydraulic head. Because the ϵ_1 error is considered to be constant over time, it would be included only in the initial observation; the subtraction would eliminate this error from subsequent observations. Thus, by this simple transformation, $\underline{V}(\underline{\epsilon}_{1\ell})$ would have one nonzero variance equal to $\sigma^2_{1\ell}$ associated with the first observation at observation location ℓ, and all other variances and covariances in the matrix would equal zero. To invert this matrix to calculate $\underline{\underline{\omega}}$, the zero diagonal elements would have to be replaced by small, nonzero values.

$\underline{\underline{V}}(\underline{\epsilon}_{3\ell})$ is a diagonal matrix because the ϵ_3 errors are uncorrelated. If, in addition, these errors are thought to have the same variance, $\underline{\underline{V}}(\underline{\epsilon}_{3\ell}) = \underline{\underline{I}}\ \sigma^2_{3\ell}$, where $\underline{\underline{I}}$ is the identity matrix and $\sigma^2_{3\ell}$ is the error variance. Under the transformation discussed above for ϵ_1 errors, $\underline{\underline{V}}(\underline{\epsilon}_{3\ell})$ becomes more complicated. The variance associated with the first observation at observation location ℓ remains unchanged, but the covariances associated with the first observation equal $-\sigma^2_{3\ell}$, all other covariances equal $\sigma^2_{3\ell}$, and all other variances equal $2\sigma^2_{3\ell}$.

The variance-covariance matrix of the ϵ_2 errors depends on the correlation between the errors. If the correlation can be expressed by an autoregressive process (Brockwell and Davis, 1987, p. 79) and the observations are made at equally spaced times, differencing can be used to produce an independent set of observations and a diagonal variance-covariance matrix. Sadeguipour and Yeh (1984) use differencing of a one-step autoregressive process and apply it to parameter estimation in a ground-water flow problem.

The sixth assumption is that the sources of error are numerous and varied enough that, by the central-limit theorem (Benjamin and Cornell, 1970, p. 251-253), the joint probability distribution function (pdf) of the errors of equation (25) are normal, so:

$$\underline{\epsilon}_\ell \sim N[\underline{0},\ \underline{\underline{V}}(\underline{\epsilon}_\ell)] = N[\underline{0},\ \underline{\underline{V}}(\underline{\epsilon}_{1\ell}) + \underline{\underline{V}}(\underline{\epsilon}_{2\ell}) + \underline{\underline{V}}(\underline{\epsilon}_{3\ell})], \qquad (26)$$

where ~ means "distributed as", and the three variance-covariance matrices are summable because of the assumed normality and independence of $\underline{\epsilon}_{1\ell}$, $\underline{\epsilon}_{2\ell}$, and $\underline{\epsilon}_{3\ell}$. By using the expressions for $\underline{\underline{V}}(\underline{\epsilon}_{1\ell})$ and $\underline{\underline{V}}(\underline{\epsilon}_{3\ell})$ described above (assuming all ϵ_3 errors have the same variance), the joint pdf can be expressed as:

$$\underline{\epsilon}_\ell \sim N[\underline{0}, \underline{1}\, \sigma^2_{1\ell} + \underline{V}(\underline{\epsilon}_{2\ell}) + \underline{I}\, \sigma^2_{3\ell}]. \quad (27)$$

The sixth assumption probably is realistic for all ground-water flow problems.

If changes in the dependent variable were considered, $\underline{1}\, \sigma^2_{1\ell}$ would be simplified further, but $\underline{I}\, \sigma^2_{3\ell}$ would become more complicated, as discussed above. If $\sigma^2_{3\ell}$ is small compared to $\sigma^2_{1\ell}$, an approximate diagonal matrix with the first variance equal to $\sigma^2_{1\ell}$ and variances for subsequent observations equal to $2\sigma^2_{3\ell}$ can produce valid regression results. If ϵ_2 errors are important and autoregressive, and the times between observations are constant, differencing methods can be used to make $\underline{V}(\underline{\epsilon}_{2\ell})$ diagonal. However, $\underline{V}(\underline{\epsilon}_{1\ell})$ and $\underline{V}(\underline{\epsilon}_{3\ell})$ would then become more complicated.

The seventh assumption is that either ϵ_1 or ϵ_2 or ϵ_3 errors dominate, and if ϵ_2 errors dominate, the errors are autoregressive and the time between observations is constant. Although, as discussed above, it could be argued that ϵ_2 errors generally are a small part of the total measurement error, the assumption that one type of error dominates is questionable under most circumstances.

If all seven assumptions are valid, no flow observations are used to calculate more than one head-dependent boundary gain or loss, and the appropriate transformation is used, the variance-covariance matrix and, therefore, the weight matrix is diagonal. This structure is advantageous computationally and because the effect of a diagonal weight matrix on parameter estimation is easy for users to understand.

In practice, the seventh assumption is most likely to be violated. Violation of the seventh assumption means that more than one classification of error is significant and off-diagonal terms in the weight matrix are needed.

<u>Estimation Procedure</u>

For a diagonal variance-covariance matrix, the quantities that need to be estimated for each observation location are $\sigma^2_{1\ell}$ and the diagonal elements of $\underline{V}(\underline{\epsilon}_{3\ell})$ or, if the ϵ_3 errors are assumed to have equal variances, $\sigma^2_{3\ell}$. If $\sigma^2_{1\ell} \neq 0.0$, $\sigma^2_{1\ell}$ is only applied to the first observation at location ℓ, and subsequent observations need to be calculated as the change from the first observation. If only one observation is made at an observation location, $\underline{V}(\underline{\epsilon}_{1\ell})$ and $\underline{V}(\underline{\epsilon}_{3\ell})$ are scalars.

Estimation of $\sigma^2_{1\ell}$ is based on knowledge of errors that are constant over time. For example, perhaps the elevation of an observation well was determined by an altimeter and is considered to be accurate to within 3 ft. To estimate $\sigma^2_{1\ell}$, this statement needs to be quantified to, for example, the probability that the true elevation is within 3 ft of the measured elevation is 95 percent. Using the sixth assumption--that the $\underline{\epsilon}_{1\ell}$ are normally distributed--a table of the cumulative distribution of a standardized normal distribution (Cooley and Naff, 1990, p. 44, or any basic statistical text, such as Benjamin and Cornell, 1970, p. 655) can be used to determine that $1.96\ \hat{\sigma}_{1\ell} = 3.0$ ft, or $\hat{\sigma}_{1\ell} = 1.53$, where $\hat{\sigma}_{1\ell}$ is the estimated standard deviation. The variance is calculated as the square of the standard deviation, so $\hat{\sigma}^2_{1\ell} = 2.34\ \text{ft}^2$. If elevations of wells are obtained from USGS topographic maps, the accuracy standards of the USGS can be used to quantify errors in elevation. The USGS (1980, p. 6) states that on their topographic maps, "***not more than ten percent of the elevations tested shall be in error more than one-half the contour interval."

Estimation of each diagonal element of $\underline{V}(\underline{\epsilon}_{3\ell})$ can be accomplished by using a similar procedure. For example, consider a loss in streamflow between two gaging stations. The upstream and downstream streamflow measurements are 3.0 ft^3/s and 2.5 ft^3/s, the measurements are each thought to be accurate to within 5 percent (using, for example, Carter and Anderson, 1963, as in Hill and others, in press), and the errors in the two measurements are considered to be independent. Stated quantitatively, perhaps the hydrologist is 90 percent certain that the first measurement is within 0.15 ft^3/s of the true value, and 95 percent certain that the second measurement is within 0.125 ft^3/s of the true value. Assuming that the errors are independent and normally distributed, the standard deviation of the first measurement is calculated from $1.65\ \hat{\sigma}_1 = 0.15\ \text{ft}^3/\text{s}$, so $\hat{\sigma}^2_1 = 0.083\ (\text{ft}^3/\text{s})^2$. The standard deviation of the second measurement is calculated from $1.96\ \hat{\sigma}_2 = 0.125\ \text{ft}^3/\text{s}$, so $\hat{\sigma}^2_2 = 0.0041\ (\text{ft}^3/\text{s})^2$. The variance of the loss of 0.5 ft^3/s equals $\hat{\sigma}_1^2 + \hat{\sigma}_2^2 = 0.0124\ (ft^3/s)^2$. The coefficient of variation (standard deviation, $0.0124^{1/2}$, divided by the loss, 0.5 ft^3/s) for the loss in streamflow is, therefore, 0.22, which indicates a moderately reliable value.

It generally is impossible to identify all errors that contribute to the variance-covariance matrix, and the variances calculated by using the methods discussed in this section are clearly approximate. In the Parameter-Estimation Package, $\hat{\sigma}_h^2$, $\hat{\underline{W}}$, $\hat{\sigma}_f^2$, and $\hat{\underline{Z}}$ of the estimated equivalent of equation (22) are specified independently to allow the user to conveniently change the weight matrix for all hydraulic-head observations and temporal change in hydraulic-head observations, or all head-dependent boundary gain-and-loss observations. This might be necessary if all the weighted residuals of one type of data are larger than for the other, as discussed in the section "Adjustments Commonly Required During Parameter Estimation". It commonly is most convenient initially to define $\hat{\underline{W}}$ and $\hat{\underline{Z}}$ as variance-covariances matrices, and set $\hat{\sigma}_h^2 = 1.0$ and $\hat{\sigma}_f^2 = 1.0$. Changes in $\hat{\sigma}_h^2$ or $\hat{\sigma}_f^2$ can then easily be interpreted as changes in the initial estimate of $\underline{V}(\underline{\epsilon})$. An additional simplification can be achieved by keeping the estimated common variance equal to 1.0 and making all changes to the other variance. Then, the final estimated error variance would ideally equal 1.0 and, if the residuals are nearly independent, they would resemble realizations from a standard normal, N(0,1), distribution.

Data interpolated from observed dependent-variable values are sometimes used to calibrate regression models. Neuman (1982), Clifton and Neuman (1982), Neuman and Jacobson (1984), and Carrera and Neuman (1986a) suggest that kriging can be used to interpolate observed hydraulic-head values, and the kriging variances and variogram can be used to calculate the variance-covariance matrix. The advantage of interpolation methods is that more hydraulic-head values are available for the regression. The disadvantage of interpolation methods is that they are not based on the physics of ground-water flow, and interpolated values might not be realistic. This problem is most severe if aquifer properties change rapidly; the interpolation method might make the hydraulic-head distribution unrealistically smooth. Use of interpolated values in the regression procedure produces correlation between the errors, so the third assumption stated above is not valid, and nonzero off-diagonal components of $\underline{V}(\underline{\epsilon})$ are required. The computer program presented in this report is not designed to accommodate nonzero off-diagonal components of $\underline{V}(\underline{\epsilon})$.

Prior Information on the Estimated Parameters

Data on model parameters that are independent of the observations of dependent variables used in the regression exist in most problems. These data are called prior information on the estimated parameters and can be included in the regression if they can be expressed in a form similar to equation (6) (Neuman and Yakowitz, 1979; Cooley, 1983a). Generally, such an expression is not difficult. If, for example, an independent estimate, P, exists for parameter β_2, P generally can be expressed as:

$$P = a_2 \beta_2 + u, \tag{28}$$

where a_2 is analogous to x_2, the NP-1 other coefficients equal zero, and u is the true error of the prior estimate of the parameter and is analogous to ϵ in equation (6). The new notation is used to emphasize the difference between the variables in equation (28) and those of equation (6). Prior information also might apply to a combination of estimated parameters, so, for example:

$$P = a_2 \beta_2 + a_4 \beta_4 + u. \tag{29}$$

This might be useful, for example, if seasonal recharge rates are being estimated and measurements of annual recharge are available; or if storage coefficients of two model layers are being estimated, and an aquifer test was conducted that measured the combined storage coefficient. Although nonlinear prior relations also could be considered by nonlinear regression, this option is not included in the Parameter-Estimation Package.

In some situations, many approximately equally reliable estimates of a parameter might exist within its applicable region, as defined by parameterization. For example, a parameter equal to the hydraulic conductivity of a large area of an aquifer can be estimated using the results of many specific-capacity tests (if the parameter is log-transformed, the natural log of the estimates needs to be used). The goal is to use the individual estimates to obtain an estimate of the hydraulic conductivity of the entire region and a variance that represents the reliability of this estimate. One way to achieve this goal is to use the arithmetic mean of the individual estimates and calculate the variance of the estimate as the variance of the mean, s_p^2/n, where $s_p^2 = (\sum_{i=1}^{n} (p_i-\bar{p})^2)/(n-1)$, n is the number of individual estimates, p_i is an individual estimate, and $\bar{p} = \sum_{i=1}^{n} p_i/n$ (Benjamin and Cornell, p. 11 and 385). There are two problems related to this method. First, recent literature (Gómez-Hernández and Gorelick, 1989) has indicated that the arithmetic mean is not always the correct averaging method to use, and the reader might want to consider alternatives, such as the geometric mean. Note that if the parameter is log-transformed, $\bar{p}$ is the geometric mean of the untransformed parameter. Second, if n is very large, s_p^2 might be so small that its use would make the parameter value virtually equal to its prior estimate. In such cases, there might be a discrepancy between what is being observed and what is represented by the model parameter, and a larger variance might be justified.

If estimates that are not approximately equally reliable exist, as would occur if hydraulic-conductivity estimates were available from aquifer and specific-capacity tests, several options exist. First, if the two types of data represent the same part of the ground-water flow system, the less accurate values can be ignored. If the two types of data represent different parts of the system and the two mean values of hydraulic conductivity are quite different, reparameterization might be justified. Other options can be developed based on the situation involved, but in no situation can the data be combined into one mean estimate using the equations from the preceding paragraph.

If many prior estimates on the estimated parameters exist, they can be expressed similarly to equation (7):

$$(P_1, a_{1,1}, a_{1,2} \cdots a_{1,NP}), (P_2, a_{2,1}, a_{2,2} \cdots a_{2,NP}), \cdots (P_{NPR}, a_{NPR,1}, a_{NPR,2} \cdots a_{NPR,NP}), \qquad (30)$$

where the added subscripts indicate the sequential number of the prior estimate, and NPR is the total number of prior estimates on the parameters. (Note that NPR as used in this part of the report equals NPR+MPR as defined in Appendix A and used in the FORTRAN code.) Once in this form, the NPR equations can be added to equation (9) by augmenting $\underline{y}$, $\underline{\underline{X}}$, $\hat{\underline{y}}$ and $\underline{\epsilon}$.

A regression equation analogous to equation (10) is produced by replacing the true, but unknown, vector $\underline{\beta}$ with the vector of estimated parameters, $\underline{b}$, and substituting the residual vector, $\underline{e}$, for the vector of true errors, $\underline{\epsilon}$. The regression equation augmented to include prior information on the parameters can be expressed as:

$$\underline{y} = \underline{\underline{X}}\,\underline{b} + \underline{e} = \hat{\underline{y}} + \underline{e} \qquad (31)$$

where,

$$\underline{y} = \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{ND} \\ \hline P_1 \\ P_2 \\ \vdots \\ P_{NPR} \end{Bmatrix}, \qquad \underline{\underline{X}} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,NP} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,NP} \\ & & \vdots & \\ x_{ND,1} & x_{ND,2} & \cdots & x_{ND,NP} \\ \hline a_{1,1} & a_{1,2} & \cdots & a_{1,NP} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,NP} \\ & \vdots & & \\ a_{NPR,1} & a_{NPR,2} & \cdots & a_{NPR,NP} \end{bmatrix},$$

$$\hat{\underline{y}} = \begin{Bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_{ND} \\ \hline \hat{P}_1 \\ \hat{P}_2 \\ \vdots \\ \hat{P}_{NPR} \end{Bmatrix}, \qquad \underline{e} = \begin{Bmatrix} e_1 \\ e_2 \\ \vdots \\ e_{ND} \\ \hline u_1 \\ u_2 \\ \vdots \\ u_{NPR} \end{Bmatrix}$$

where the true errors, u, have been replaced by the residuals, u. The $\underline{b}$ is the same as in equation (10).

The unweighted sum-of-squares objective function (eq. 11) now becomes:

$$S(\underline{b}) = \sum_{q=1}^{ND} (y_q - \hat{y}_q)^2 + \sum_{p=1}^{NPR} (P_p - \hat{P}_p)^2. \qquad (32)$$

In a sense, a penalty function has been introduced that increases the objective-function value as the parameter estimates produced by the regression routine diverge from the prior estimates. By using the augmented vectors and matrices of equation (32), $S(\underline{b})$ still can be expressed as $(\underline{y}-\underline{\underline{X}}\ \underline{b})^T(\underline{y}-\underline{\underline{X}}\ \underline{b})$ or $\underline{e}^T\underline{e}$, and the unweighted normal equations are still as in equation (16).

Prior estimates of the parameters are weighted to indicate their reliability relative to each other and relative to the observations of dependent variables, and to accommodate the different units the prior estimates might have. Assuming that the errors on the hydraulic head, streamflow gain-and-loss, and parameter data are uncorrelated, the weight matrix of equation (23), in which σ_h^2 is assumed to be the common error variance, can be augmented to become:

$$\underline{\underline{\omega}} = \begin{bmatrix} \underline{\underline{W}}^{-1} & \underline{\underline{0}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{Z}}^{-1} \dfrac{\sigma_h^2}{\sigma_f^2} & \underline{\underline{0}} \\ \underline{\underline{0}} & 0 & \underline{\underline{U}}^{-1} \sigma_h^2 \end{bmatrix}, \qquad (33)$$

where $\underline{\underline{U}}$ is the variance-covariance matrix for true errors in the prior estimates of the parameters.

The most general form of $\underline{\underline{U}}$ could be as complicated as the most general form of the variance-covariance matrix on observations of dependent variables, which was discussed in previous sections of this report. $\underline{\underline{U}}$ can be simplified by using assumptions that are nearly identical to the seven assumptions discussed in those earlier sections. The resulting $\underline{\underline{U}}$ has nonzero elements only on its main diagonal, and these elements equal the variances of the measurement errors of the prior information on the estimated parameters. $\underline{\underline{U}}$ can be estimated by using methods similar to those presented in the earlier sections.

If the parameters with prior information are divided into groups with the same units (for example, one group might be hydraulic-conductivity values and another might be recharge rates), the variance-covariance matrix for each group can be expressed as a dimensionless matrix multiplied by a group variance. For example, with three groups, $\underline{U}$ can be expressed as:

$$\underline{U} = \begin{bmatrix} \underline{U}_{p_1} \sigma^2_{p_1} & \underline{0} & \underline{0} \\ \underline{0} & \underline{U}_{p_2} \sigma^2_{p_2} & \underline{0} \\ \underline{0} & \underline{0} & \underline{U}_{p_3} \sigma^2_{p_3} \end{bmatrix}, \qquad (34a)$$

and $\underline{U}^{-1}\sigma^2_h$ can be expressed as:

$$\underline{U}^{-1}\sigma^2_h = \begin{bmatrix} \underline{U}^{-1}_{p_1} \dfrac{\sigma^2_h}{\sigma^2_{p_1}} & \underline{0} & \underline{0} \\ \underline{0} & \underline{U}^{-1}_{p_2} \dfrac{\sigma^2_h}{\sigma^2_{p_2}} & \underline{0} \\ \underline{0} & \underline{0} & \underline{U}^{-1}_{p_3} \dfrac{\sigma^2_h}{\sigma^2_{p3}} \end{bmatrix}, \qquad (34b)$$

where $\sigma^2_{p_1}$, $\sigma^2_{p_2}$, and $\sigma^2_{p_3}$ are the variances for groups p_1, p_2, and p_3, respectively. This representation is sometimes helpful when trying to estimate the weight matrix of the prior parameter estimates.

The weighted sum-of-squares objective function is still expressed by equation (18), except that $\underline{e}$ is augmented as in equation (31), and $\underline{\omega}$ is augmented as in equation (33). Similarly, the weighted normal equations can be calculated by using equation (19) with the augmented vectors and matrices of equations (31) and (33).

Input Data Used by the Parameter-Estimation Package to Calculate the Complete Weight Matrix

In the Parameter-Estimation Package, diagonal elements of $\hat{\underline{V}}(\underline{\epsilon})$ are calculated using $\hat{\sigma}_h^2$, $\underline{W}$, $\hat{\sigma}_f^2$, $\hat{\underline{Z}}$ and $\hat{\underline{U}}$, which are estimated equivalents of the terms in equations (22) and (34). From a practical perspective, $\hat{\sigma}_h^2$ and $\hat{\sigma}_f^2$ can be thought of as scaling factors, so that the diagonal elements of $\underline{W}$ and $\hat{\underline{Z}}$ are scaled variances. These scaled variances can be specified directly, or the user can specify scaled standard deviations or scaled coefficients of variation. When scaled standard deviations are specified, the scaled variances are calculated as the square of the scaled standard deviation; when scaled coefficients of variation are specified, the scaled variances are calculated as $(\gamma y)^2$, where γ is the scaled coefficient of variation and y is the observed value of hydraulic head, temporal change in hydraulic head, or head-dependent boundary gain or loss. The diagonal weight matrix is calculated from the variances using equation (17), (23), or (33) and (34b), where the common error variance is set equal to σ_h^2 or σ_f^2.

Components of the diagonal weight matrix related to prior parameter estimates are calculated similarly, but the variances, standard deviations, or coefficients of variation specified by the user are not scaled. For example, the $\sigma_{p_i}^2$, $i = 1,3$, of equation (34a) would be included in the value specified by the user.

Analysis of Results for Linear Problems

Analysis of final parameter estimates and the assumed model are required to: (1) Discriminate between models that use different parameterizations, boundary conditions, or other choices of model construction by comparing the results achieved from the different models, and (2) determine if the simulated results are similar enough to observed values and if the parameter estimates are reliable enough to justify using the calibrated model to draw conclusions about the aquifer system or to predict aquifer response. The second objective would benefit from calculation of confidence intervals on the predicted quantities (Draper and Smith, 1981, p. 94), but these confidence intervals are not included in the Parameter-Estimation Package.

Many of the methods discussed in this section are discussed in more detail in Cooley and Naff (1990, ch. 5). See Cooley and others (1986) for an example of how the methods can be used.

Parameter estimates are analyzed to determine: (1) How reliably they are estimated by the available data; and (2) whether some of the parameters are strongly correlated. A parameter cannot be reliably estimated if the values of the dependent variables at all observation locations are insensitive to that parameter and prior estimates are missing or inaccurate. This might occur, for example, when trying to estimate hydraulic conductivity in an area of an aquifer that is remote from all observation locations. Strong correlation between parameters occurs when similar values of the dependent variables at the observation locations are produced by numerous, different combinations of parameter values and prior estimates of the parameters do not exist or are inaccurate. For example, consider a Darcy cylinder (fig. 15) packed with sand that has a known hydraulic head, h_1, at X=0, and the unknown parameters K, the hydraulic conductivity of the sand, and Q, the flow through the cylinder. Hydraulic head at any distance, X, along the cylinder can be calculated as $h=h_1-\frac{Q}{KA}X$. As long as the ratio Q/K is constant, the hydraulic-head distribution along the length of the cylinder will be the same. Thus, Q and K are perfectly correlated and cannot be estimated independently by using observed hydraulic heads.

The reliability and correlation of parameter estimates can be analyzed by using the variance-covariance matrix, $\underline{V}(\underline{b}')$, for the final estimated parameters, $\underline{b}'$ (Bard, 1974, p. 59):

$$\underline{V}(\underline{b}') = s^2(\underline{X}^T\underline{\omega}\ \underline{X})^{-1} \tag{35}$$

where $\underline{V}(\underline{b}')$ is an NP by NP matrix; s^2, the estimated error variance, is the minimized value of the objective function divided by ND+NPR-NP; and $\underline{X}$ and $\underline{\omega}$ are augmented as in equations (31) and (33) if there is prior information on the parameters. The validity of equation (35) depends on the model being nearly linear in the vicinity of $\underline{b}'$.

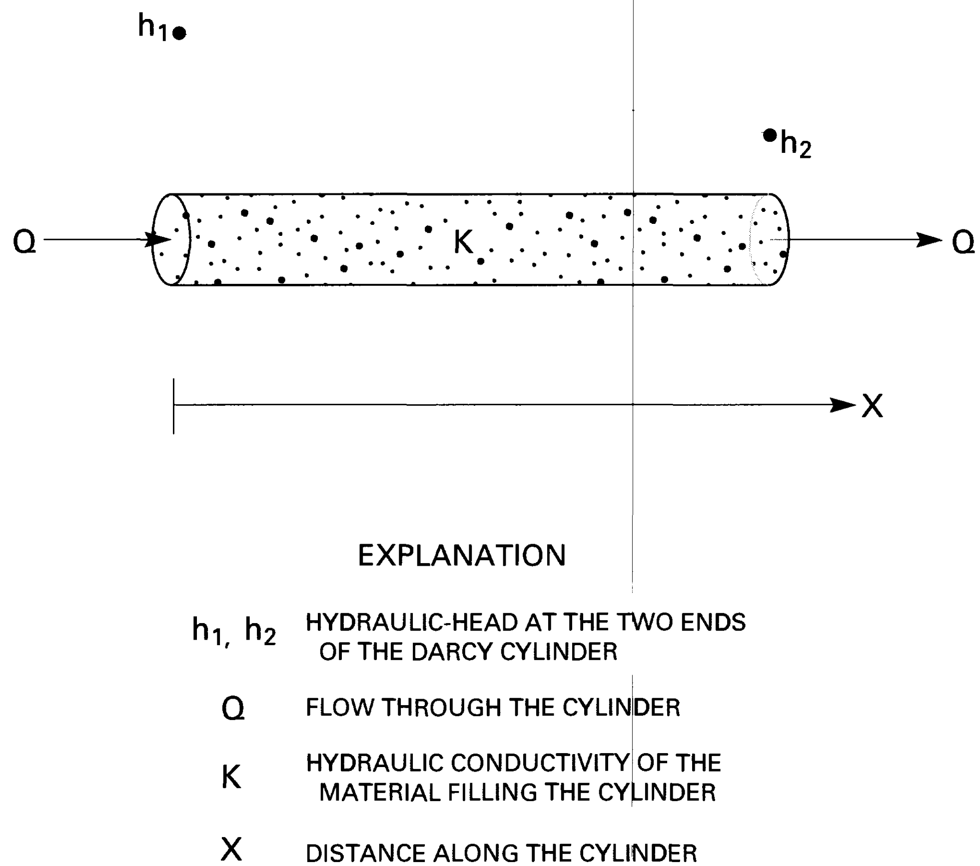


Figure 15.--Darcy cylinder.

Each element along the main diagonal of the variance-covariance matrix, v_{ii}, equals the estimated variance of final parameter estimate b_i'. The square root of each variance is the standard deviation of the parameter estimate, and the coefficient of variation equals $(v_{ii})^{½}/b_i'$, which is dimensionless. The coefficient of variation is the easiest statistic by which to compare the reliability of different parameters estimated in a single parameter-estimation run. However, as discussed in item 2 in the section "Output for Test Case 1" in Appendix A, interpretation of the coefficients of variation can be confusing if there are log-transformed parameters. For linear problems, the variances calculated by using equation (35) can be used to calculate confidence intervals on the estimated parameters (Draper and Smith, 1981, p. 94) as:

$$b_i \pm (v_{ii})^{½} \times t(ND+NPR-NP,\ 1.0-\alpha/2) \qquad (36)$$

where, t(•,•) is a student-t probability distribution in which the first argument equals the degrees of freedom, the second argument equals the probability that β_i occurs within the confidence interval, and α equals the probability that β_i occurs outside the confidence interval. The student-t probability distribution is presented in table 2. For log-transformed parameters, the confidence interval needs to be calculated on the log-transformed parameter estimate. The exponential of the estimate equals the mode of the lognormal probability distribution; the exponential of the confidence limits are confidence limits on the mode, which might not be symmetric about the mode.

Parameter reliability also can be characterized by using the eigenvalues and eigenvectors of $\underline{\underline{V}}(\underline{b}')$ (Carrera and Neuman, 1986c). Unlike the previous work, in the Parameter-Estimation Package $\underline{\underline{V}}(\underline{b}')$ is scaled with the final estimated parameter values before eigenvalues and eigenvectors are calculated. This is necessary to ensure that the eigenvalues and eigenvectors, when used as described below, are indicative of parameter reliability instead of differences in parameter values. Elements of the scaled matrix are calculated as $(v_{ij})/b'_i b'_j$. The least reliable parameter values can be identified by inspecting the eigenvectors associated with the largest eigenvalues. The parameters with the largest elements of these eigenvectors are estimated least reliably. The most reliable parameter values can be identified by inspecting the eigenvectors associated with the smallest eigenvalues. The parameters related to the largest elements of these eigenvectors are estimated most reliably.

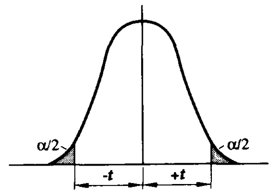
Parameter reliability can be characterized by using either the coefficient of variation or eigenvalues and eigenvectors, but the conclusions generally are identical. Use of the coefficient of variation is easier, so its use is encouraged.

Table 2.--*Student t probability distribution*

[Modified from Draper and Smith (1981, p. 532) with permission from publisher]

Degrees of freedom	Probability[1]									
	0.9	0.7	0.5	0.3	0.2	0.1	0.05	0.02	0.01	0.001
1	0.158	0.510	1.000	1.963	3.078	6.314	12.706	31.821	63.657	636.619
2	0.142	0.445	0.816	1.386	1.886	2.920	4.303	6.965	9.925	31.598
3	0.137	0.424	0.765	1.250	1.638	2.353	3.182	4.541	5.841	12.924
4	0.134	0.414	0.741	1.190	1.533	2.132	2.776	3.747	4.604	8.610
5	0.132	0.408	0.727	1.156	1.476	2.015	2.571	3.365	4.032	6.869
6	0.131	0.404	0.718	1.134	1.440	1.943	2.447	3.143	3.707	5.959
7	0.130	0.402	0.711	1.119	1.415	1.895	2.365	2.998	3.499	5.408
8	0.130	0.399	0.706	1.108	1.397	1.860	2.306	2.896	3.355	5.041
9	0.129	0.398	0.703	1.100	1.383	1.833	2.262	2.821	3.250	4.781
10	0.129	0.397	0.700	1.093	1.372	1.812	2.228	2.764	3.169	4.587
11	0.129	0.396	0.697	1.088	1.363	1.796	2.201	2.718	3.106	4.437
12	0.128	0.395	0.695	1.083	1.356	1.782	2.179	2.681	3.055	4.318
13	0.128	0.394	0.694	1.079	1.350	1.771	2.160	2.650	3.012	4.221
14	0.128	1.393	0.692	1.076	1.345	1.761	2.145	2.624	2.977	4.140
15	0.128	1.393	0.691	1.074	1.341	1.753	2.131	2.602	2.947	4.073
16	0.128	0.392	0.690	1.071	1.337	1.746	2.120	2.583	2.921	4.015
17	0.128	0.392	0.689	1.069	1.333	1.740	2.110	2.567	2.898	3.965
18	0.127	1.392	0.688	1.067	1.330	1.734	2.101	2.552	2.878	3.922
19	0.127	1.391	1.688	1.066	1.328	1.729	2.093	2.539	2.861	3.883
20	0.127	1.391	0.687	1.064	1.325	1.725	2.086	2.528	2.845	3.850
21	0.127	0.391	0.686	1.063	1.323	1.721	2.080	2.518	2.831	3.819
22	0.127	0.390	0.686	1.061	1.321	1.717	2.074	2.508	2.819	3.792
23	0.127	0.390	0.685	1.060	1.319	1.714	2.069	2.500	2.807	3.767
24	0.127	0.390	0.685	1.059	1.318	1.711	2.064	2.492	2.797	3.745
25	0.127	0.390	0.684	1.058	1.316	1.708	2.060	2.485	2.787	3.725
26	0.127	0.390	0.684	1.058	1.315	1.706	2.056	2.479	2.779	3.707
27	0.127	0.389	0.684	1.057	1.314	1.703	2.052	2.473	2.771	3.690
28	0.127	0.389	0.683	1.056	1.313	1.701	2.048	2.467	2.763	3.674
29	0.127	0.389	0.683	1.055	1.311	1.699	2.045	2.462	2.756	3.659
30	0.127	0.389	0.683	1.055	1.310	1.697	2.042	2.457	2.750	3.646
40	0.126	0.388	0.681	1.050	1.303	1.684	2.021	2.423	2.704	3.551
60	0.126	0.387	0.679	1.046	1.296	1.671	2.000	2.390	2.660	3.460
120	0.126	0.386	0.677	1.041	1.289	1.658	1.980	2.358	2.617	3.373
∞	0.126	0.385	0.674	1.036	1.282	1.645	1.960	2.326	2.576	3.291

[1]Probability = Area in two tails of distribution outside ±t-value in table.



Distribution of t

Table 2. Student-t probability distribution.

The elements off the main diagonal of $\underline{V}(\underline{b}')$, v_{ij}, $i \neq j$, are the covariances. The correlation between any two parameters is measured by correlation coefficients, which are calculated as $(v_{ij})/(v_{ii})^{1/2}(v_{jj})^{1/2}$. Correlation coefficients can have values ranging from -1.0 to +1.0: values of about zero indicate no correlation between b'_i and b'_j; values of about -1.0 or +1.0 indicate strong correlation. In the simple, two-parameter problem presented at the beginning of this section, the correlation between Q and K was +1.0.

Correlations also might occur between groups of parameters. If these correlations are strong, similar dependent-variable values are produced by different combinations of values of all the parameters in the group. For example, in the two-layer aquifer system described as test case 1 in Appendix A, doubling all hydraulic-conductivity values and all fluxes produces an identical hydraulic-head distribution. The correlation coefficients between pairs of parameters from a group of parameters that are correlated generally are about 1.0 or -1.0, but it might not be obvious from the correlation coefficients that the parameters form a group. Such a group can be identified by using the eigenvectors of $\underline{V}(\underline{b}')$ scaled as described above. Parameters in such groups are associated with nearly equal elements within individual eigenvectors. The implied functional dependence is stronger if the elements associated with the parameters are nearly equal in several eignevectors, and if the eigenvector elements and the eigenvalue associated with each eigenvector are large.

The assumed model also is analyzed to determine if the simulated dependent-variable values are as expected--that is, are indicative of a valid regression. Unexpected results would indicate that the assumed model is biased. Evaluation of the assumed model is accomplished through statistical and graphical analyses of the weighted observations, $\underline{\omega}^{1/2}\underline{y}$, the weighted final simulated values, $\underline{\omega}^{1/2}\hat{\underline{y}}$, and the weighted final residuals, $\underline{\omega}^{1/2}\underline{e} = \underline{\omega}^{1/2}(\underline{y}-\hat{\underline{y}})$. Three commonly used statistics are the average weighted residual, the estimated error variance, and the correlation coefficient. The average weighted residual is calculated as the sum of the weighted residuals divided by the number of weighted residuals, and needs to be about zero. The estimated error variance, s^2, was discussed after equation (35) and is a measure of how similarly the final simulated values match the observed values. For simulations with the same weight matrix, smaller values of s^2 indicate a better match; s^2 cannot be used to compare regressions which use different weight matrices. The correlation coefficient, R, measures how closely trends in the simulated values match trends in the observed values and is calculated as (Cooley and Naff, 1990, p. 166):

$$R = \frac{(\hat{\underline{\omega}}^{-1/2}\underline{y} - \underline{m}_y)^T(\hat{\underline{\omega}}^{-1/2}\hat{\underline{y}} - \underline{m}_{\hat{y}})}{[(\hat{\underline{\omega}}^{-1/2}\underline{y} - \underline{m}_y)^T(\hat{\underline{\omega}}^{-1/2}\underline{y} - \underline{m}_y)(\hat{\underline{\omega}}^{-1/2}\hat{\underline{y}} - \underline{m}_{\hat{y}})^T(\hat{\underline{\omega}}^{-1/2}\hat{\underline{y}} - \underline{m}_{\hat{y}})]^{1/2}} \tag{37}$$

where $\underline{y}$, $\hat{\underline{y}}$, and $\hat{\underline{\omega}}$ are defined in equations (9), (10), and (23).

$$\underline{m}_y = \left[\left(\sum_{q=1}^{ND} (\hat{\underline{\omega}}^{-1/2}\underline{y})_q\right)/ND\right] \underline{1},$$

$$\underline{m}_{\hat{y}} = \left[\left(\sum_{q=1}^{ND} (\hat{\underline{\omega}}^{-1/2}\hat{\underline{y}})_q\right)/ND\right] \underline{1},$$

and $\underline{1}$ is a vector of length ND with each element equal to 1. Note that $\underline{m}_y$ is simply a vector with each component equal to the average value of the weighted dependent-variable observations, and that $\underline{m}_{\hat{y}}$ is an analogous vector using the weighted calculated dependent-variable values. Generally R needs to be greater than 0.90. R also is calculated with $\underline{y}$, $\hat{\underline{y}}$, and $\underline{\omega}$ augmented as in equations (31) and (33), in which case ND+NPR replaces ND when calculating $\underline{m}_y$ and $\underline{m}_{\hat{y}}$.

The statistical properties of the weighted residuals are checked against their expected statistical properties to identify unexpected patterns that would indicate bias in the model. One might expect that if the weighted true errors satisfied the conditions $E(\underline{\omega}^{-1/2}\underline{\epsilon}) = \underline{0}$ and $\underline{V}(\underline{\omega}^{-1/2}\underline{\epsilon}) = \underline{I}\sigma^2$, the same would be true of the weighted residuals. If this were the situation, the weighted residuals would be independent and have equal variances. Although they actually might be correlated, as discussed below, a first step toward analyzing the weighted residuals is accomplished by testing whether they satisfy these properties. For the purposes of the first of the two tests performed, as described below, it also is assumed that $\underline{\epsilon}$ and, therefore, $\underline{e}$ are normally distributed. If the tests indicate that the residuals are normal, independent, and have equal variance, the model probably is unbiased.

The correlation coefficient between the weighted residuals ordered from smallest to largest and the order statistics from a N(0,1) probability distribution function (Brockwell and Davis, 1987, p. 304) is the first statistic used to test for independent, normally distributed weighted residuals. This statistic was chosen instead of other statistics, such as chi-squared and Kolomogorov-Smirnov, because it is more powerful for commonly used sample sizes (Shapiro and Francia, 1972). The correlation coefficient, R_N^2, is calculated as:

$$R_N^2 = \frac{[(\underline{e}_o-\underline{m})^T \underline{\tau}]^2}{[(\underline{e}_o-\underline{m})^T (\underline{e}_o-\underline{m})](\underline{\tau}^T\underline{\tau})}, \qquad (38)$$

where all vectors are of length ND, $\underline{m}$ is a vector with all components equal to the average of the weighted residuals, $\underline{e}_o$ are the weighted residuals ordered from smallest to largest, and $\underline{\tau}$ is a vector with the ith element equal to the ordinate value of a N(0,1) probability distribution function for a cumulative probability equal to $u_i = (i-0.5)/ND$. If, for example, $u_i = 0.8531$, $\tau_i = 1.05$.

The hypothesis that the weighted residuals are derived from an independent, normal distribution is rejected if R_N^2 is too much smaller than its ideal value of 1.0. The critical value below which the hypothesis is rejected depends on the value of ND and on the significance level (Benjamin and Cornell, 1970, p. 406) chosen by the user (table 3). If R_N^2 indicates that the weighted residuals are not independent and normally distributed, the hypothesis that they are correlated and normally distributed needs to be tested, as discussed below. In any case, the three graphical analyses described below need to be performed.

Table 3: *Critical values of R_N^2 below which the hypothesis that the weighted residuals are independent and normally distributed is rejected (From Shapiro and Francia, 1972; Brockwell and Davis, 1987, p. 304)*

ND or ND+NPR+MPR	Significance level 0.05	0.10	ND or ND+NPR+MPR	Significance level 0.05	0.10
35	0.943	0.952	81	0.970	0.975
			83	0.971	0.976
50	0.953	0.963	85	0.972	0.977
51	0.954	0.964	87	0.972	0.977
53	0.957	0.964	89	0.972	0.977
55	0.958	0.965			
57	0.961	0.966	91	0.973	0.978
59	0.962	0.967	93	0.973	0.979
			95	0.974	0.979
61	0.963	0.968	97	0.975	0.979
63	0.964	0.970	99	0.976	0.980
65	0.965	0.971			
67	0.966	0.971	131	0.980	0.983
69	0.966	0.972	200	0.987	0.989
71	0.967	0.972			
73	0.968	0.973			
75	0.969	0.973			
77	0.969	0.974			
79	0.970	0.975			

The correlation coefficient of equation (38) is also calculated using the vector of weighted residuals augmented for prior parameter estimates. The augmented residual vector is shown after equation (31); the weight matrix is shown in equation (33); all vectors of equation (38) would be of length ND+NPR. Inconsistency between the dependent-variable data and the prior parameter estimates is indicated if this second evaluation of R_N^2 is smaller than the value calculated using only dependent-variable weighted residuals.

The runs test (Draper and Smith, 1981, p. 157-162) is used to test for independent residuals. In the runs test, the number of sequences of residuals of the same sign (u) is counted, along with the total number of positive residuals (n_1), and the total number of negative residuals (n_2). The expected number of runs equals $\mu = [2n_1n_2/(n_1+n_2)]+1.0$, and the variance equals $\sigma^2 = [2n_1n_2(2n_1n_2-n_1-n_2)]/[(n_1+n_2)^2(n_1+n_2-1)]$. The test statistic for too few runs equals $z_f = (u-\mu+0.5)/\sigma$; the test statistic for too many runs equals $z_m = (u-\mu-0.5)/\sigma$. If $n_1>10$ and $n_2>0$, u is normally distributed and critical values for z_f and z_m are printed by the Parameter-Estimation Package. Otherwise, a table such as the table in Draper and Smith (1981, p. 160-161) needs to be used to evaluate whether u is too small or too large.

The runs test is included in the Parameter-Estimation Package because it takes the order of the residuals into account, which is ignored in the correlation coefficient of equation (38). Normally, observations are grouped by location in transient simulations, and too few runs commonly indicates positive serial correlation between residuals at individual locations.

Except when ND>>NP (Draper and Smith, 1981, p. 152; Cooley and others, 1986, p. 1771) the weighted residuals generally are correlated, even if the weighted true errors are not. The expected variance-covariance matrix for the weighted residuals is $\underline{V}(\underline{\omega}^{-\frac{1}{2}}\underline{e}) = (\underline{I} - \underline{X}(\underline{X}^T\underline{\omega}\ \underline{X})^{-1}\ \underline{X}^T\underline{\omega})\sigma^2$ (Bard, 1974, p. 194; Cooley and Naff, 1990, p. 168), and graphical analyses of the weighted residuals is done by comparing plots of the weighted residuals to plots of realizations that have the same variance-covariance matrix. Assuming that the weighted true errors are normally distributed, the realizations would be derived from the joint normal probability distribution function $N(\underline{0},(\underline{I} - (\underline{X}(\underline{X}^T\underline{\omega}\ \underline{X})^{-1}\ \underline{X}^T\underline{\omega})\sigma^2)$. A slightly modified version of the FORTRAN program presented by Cooley and Naff (1990, p. 176-183) can be used to produce realizations from this probability distribution function. The modifications are described in Appendix B of this report.

Three graphical analyses of the residuals are: (1) Plot the weighted residuals and several realizations on maps of each of the simulated model layers; (2) plot the weighted residuals and several realizations against their respective $\hat{y}_q$ values on graph paper; and (3) plot the weighted residuals and several realizations on probability paper. See Cooley and Naff (1990, p. 168-170) for more information about graphical analyses.

Model discrimination can be performed by using the methods described above to compare the results from alternative models. Models with smaller parameter coefficients of variation, parameter correlations less than 0.95, smaller values of s^2 (if the weight matrices are identical), larger values of R^2, and weighted residuals that more closely match their expected statistical characteristics are considered to be better. Additional statistics are presented in the section "Analysis of Results for Nonlinear Problems". Model discrimination is difficult because usually no one model is the best for all the measures indicated above. If several models appear to be equally good, it might be helpful to perform some predictive runs by using all of these models.

Adjustments Commonly Required During Parameter Estimation

As the modeler learns about the physical system, the model, and the data through modeling and parameter estimation, adjustments commonly are made to the model, parameter definition, weight matrix, and observation data sets. In part, this is the same process a modeler goes through when calibrating by trial and error, but use of regression allows hypotheses to be tested more rigorously and provides more information about the effects of the data on parameter estimates and the reliability of the estimates.

For a model to be satisfactory, the weighted residuals produced by the final model need to satisfy two criteria: (1) The estimated common error variance ($\hat{\sigma}_h^2$ or $\hat{\sigma}_f^2$) needs to approximately equal the calculated error variance, s^2; and (2) the weighted residuals need to satisfy $E(\underline{\omega}^{½}\underline{e}) = 0$ and $\underline{V}(\hat{\underline{\omega}}^{½}\underline{e}) = (\underline{I}-\underline{X}(X^T\hat{\underline{\omega}}\,\underline{X})^{-1}\underline{X}^T\hat{\underline{\omega}})\hat{\sigma}^2$. As discussed in the previous section, violation of the second criterion is indicative of bias in the model.

One conspicuous indication of a problem is the presence of weighted residuals at convergence of parameter estimation (or after several parameter-estimation iterations if convergence is not achieved) that are so large that they dominate the regression, or so small that they are ignored by the regression. In such situations, the model needs to be evaluated to determine whether: (1) Some aspect of model construction is incorrect; (2) the parameter definition needs to be modified; (3) the weight matrix needs to be changed so that the data affect the regression more equally; or (4) dependent-variable observations or prior parameter estimates need to be eliminated or corrected because they are clearly biased.

Model construction can be modified, for example, by changing the representation of physical boundary conditions, such as conversion of an overlying constant-head boundary to a recharge boundary; and correction of input data, such as known pumpage. Many of these are the same kind of changes a modeler would consider as part of trial and error calibration.

Parameter definition can be modified, for example, by adding, omitting, dividing, or combining parameters, or making other changes in the parameterization. Note that the final parameterization needs to be consistent with what is known about hydrogeology of the ground-water flow system.

Changes to the weight matrix are common because $\hat{\sigma}_h^2$, $\hat{\sigma}_f^2$, and $\hat{\sigma}_{p_i}^2$ commonly are unknown, and there might be some question about the values in $\hat{\underline{W}}$, $\hat{\underline{Z}}$, and $\hat{\underline{U}}_{p_i}$ (see equations 17, 23, or 33 and 34). If all the weighted residuals for one data type, such as head-dependent boundary gains and losses, are extremely large or extremely small, $\hat{\sigma}_h^2$, $\hat{\sigma}_f^2$, and $\hat{\sigma}_{p_i}^2$ might need to be adjusted to achieve a more even distribution. To adjust $\hat{\sigma}_h^2$, $\hat{\sigma}_f^2$, and $\hat{\sigma}_{p_i}^2$ such that the desired effect on the weights is achieved, consider how the weights are calculated. If $\hat{\sigma}_h^2$ is the common variance, $\hat{\sigma}^2$, the weighting of hydraulic heads and temporal changes in hydraulic heads can be adjusted only by adjusting components of $\hat{\underline{W}}$. The weighting of head-dependent boundary gains and losses would be sensitive to the ratio $\hat{\sigma}_h/\hat{\sigma}_f$. For example, if $\hat{\sigma}_f^2$ is increased by a factor of 4 the weighting of all gains and losses would be decreased by a factor of 2. The weighting of prior parameter estimates would be sensitive to the ratio $\hat{\sigma}_f/\hat{\sigma}_{p_i}$.

If $\hat{\underline{\omega}}$ is realistic and the weighted residuals have all desired properties, but the estimated common error variance does not equal the calculated error variance ($\hat{\sigma}^2 \neq s^2$), the following procedure can be followed if changing $\hat{\underline{W}}$ or $\hat{\underline{Z}}$, and $\hat{\underline{U}}$ of equation (33) is acceptable. Set the estimated common error variance equal to s^2 and multiply all elements of $\hat{\underline{U}}$ and either $\hat{\underline{W}}$ or $\hat{\underline{Z}}$ (depending on whether $\hat{\sigma}_f^2$ or $\hat{\sigma}_h^2$ is defined as the common error variance, respectively) by $s^2/\hat{\sigma}_{OLD}^2$. These changes will produce an identical weight matrix, so that the regression will be the same.

In some cases, individual observations dominate the regression or are ignored. Components of $\hat{\underline{W}}$, $\hat{\underline{Z}}$, and $\hat{\underline{U}}_{p_i}$ can be adjusted to achieve a more even distribution, but the reason for the problem needs to be considered before making such adjustments. For example, Cooley and others (1986, p. 1764) anticipated that the small error with which shallow wells could be measured would produce accurate hydraulic-head observations at these points, and thus assigned them large weights. During calibration it was determined that "***the model fit no better at these points than elsewhere." (Cooley and others, 1986, p. 1772). Apparently, the shallow wells were affected by shallow, local flow systems not represented in the regional-scale model, and this situation produced residuals that were as large as those associated with inaccurately observed deep wells. Decreasing the weights for observations from shallow wells produced more satisfactory results.

Although the process described above can become mechanical, the user needs to remember that the weight matrix reflects data accuracy. The final weight matrix needs to be justifiable in terms of known errors and inadequacies of the data.

Dependent-variable observations or prior parameter estimates from the regression can be eliminated only if careful consideration of the problem indicates that they are clearly biased. The danger is in eliminating data that is indicative of a legitimate problem in the model.

Once residuals are fairly evenly distributed, most of the anticipated changes have been made to the model, and parameter estimates are close to satisfactory, an attempt needs to be made to test whether the goals stated in the beginning of this section--that is, the estimated common error variance needs to approximately equal the calculated error variance, and the weighted residuals should satisfy $E(\hat{\underline{\omega}}^{½}\underline{e}) = \underline{0}$ and $\underline{V}(\hat{\underline{\omega}}^{½}\underline{e}) = (\underline{I}-\underline{X}(\underline{X}^T\hat{\underline{\omega}}\,\underline{X})^{-1}\underline{X}^T\hat{\underline{\omega}})\sigma^2$--are satisfied by using the tests described in the previous section. More changes in the model and the weight matrix might be required.

With so many things that might be changed, the user might wonder about the uniqueness of the final model. Even if any single run of the regression routine produces a unique set of estimates (and this has its own set of problems, as discussed by many authors such as Carrera and Neuman, 1986b), there is no guarantee that the final model is unique. The enhanced capability of regression modeling does, however, provide the modeler with an effective means of evaluating and discriminating between different possible models.

PARAMETER ESTIMATION AND ANALYSIS OF RESULTS USING NONLINEAR REGRESSION

In nonlinear regression, the assumed model need not be a linear function of the parameters, as in equation (6). Nonlinear regression needs to be used for ground-water flow problems because hydraulic head, as the solution of the ground-water flow equation, is a nonlinear function of many of the parameters a modeler would want to estimate. In contrast, for confined aquifers, the ground-water flow equation is classified as linear because hydraulic head is a linear function of time and space. An example is discussed in the next section.

Nonlinearity of the Ground-Water Flow Equation

The nonlinearity of the ground-water flow equation can be illustrated by considering Darcy's Law, the constitutive relation on which the ground-water flow equation is based. The differential form of Darcy's Law as applied to the Darcy cylinder shown in figure (15) is:

$$Q = -KA \frac{\partial h}{\partial X}, \tag{39}$$

where

Q is the flux $[L^3/T]$;

K is the hydraulic conductivity of the saturated material [L/T];

A is the cross-sectional area $[L^2]$;

h is hydraulic head [L]; and

X is distance along an axis parallel to the direction of flow [L].

Equation (39) can be solved for the hydraulic head at any distance, X, along the cylinder to achieve:

$$h = h_1 - \frac{Q}{KA} X, \tag{40}$$

where h_1 is the hydraulic head at X = 0. The derivatives of equation (40) with respect to X, Q and K are:

$$\frac{\partial h}{\partial X} = -\frac{Q}{KA}$$

$$\frac{\partial h}{\partial Q} = -\frac{1}{KA} X \qquad (41)$$

$$\frac{\partial h}{\partial K} = \frac{Q}{K^2 A} X.$$

Thus, the hydraulic head is a linear function of X because ∂h/∂X is independent of X. Hydraulic head also is a linear function of Q, but is a nonlinear function of K; ∂h/∂Q and ∂h/∂K are both functions of K. The derivatives ∂h/∂Q or ∂h/∂K would be sensitivities in a parameter-estimation problem in which Q or K were being estimated. As in this simple example, sensitivities with respect to flows, such as Q, are nearly always functions of aquifer properties; sensitivities with respect to aquifer properties, such as K, are nearly always functions of the aquifer properties and the flows.

It is interesting to note that the example discussed above is linear if the ratio Q/K is considered. That is,

$$\frac{\partial h}{\partial (Q/K)} = -\frac{X}{A} \qquad (42)$$

which is independent of any parameters. In this case, simple linear regression could be used to estimate Q/K. Ground-water flow problems can sometimes be linearized by redefining parameters in this way, but the prospects become less likely as the flow system becomes more complicated.

Assumed Nonlinear Model

For the nonlinear regression considered in this report, the true model can be expressed as:

$$\tilde{y} = g^* (\xi_1, \xi_2 \cdots \xi_k; \beta_1, \beta_2 \cdots \beta_{NP}) \qquad (43)$$

where

$\tilde{y}$ is a dependent variable, such as simulated hydraulic head, temporal change in hydraulic head, or head-dependent boundary gain or loss;

$g^*(\cdot)$ is general notation for a function of the values in the parentheses, and represents the true model;

ξ_i is an independent variable, such as spatial dimension or time; and

β_i is a true, unknown parameter value.

By using the same reasoning applied after equation (6) to linear problems, a vector of observations of dependent variables, $\underline{y}$, can be expressed in terms of parameter estimates $\underline{b}$ and model estimate $g(\cdot)$ as:

$$\underline{y} = \underline{g}\ (\underline{\xi},\ \underline{b}) + \underline{e} = \hat{\underline{y}} + \underline{e} \tag{44}$$

which is analogous to equation (10) of this report and equivalent to equation (3.3-3) of Cooley and Naff (1990, p. 61).

Although the function, $g(\cdot)$, of equation (44) commonly is thought of as an analytical function of the independent variables and parameters, it also might be a numerical model. In this report $g(\cdot)$ is calculated using the ground-water flow model presented by McDonald and Harbaugh (1988) and Prudic (1989), which was described briefly in the first part of this report. The model inputs that can be calculated with parameters estimated by nonlinear regression are listed in table 1; other model inputs need to be specified by the modeler.

Maximum-Likelihood Objective Function

The maximum-likelihood objective function has been used extensively in the time-series literature (Brockwell and Davis, 1987, ch. 8), and also has been used to estimate parameters in ground-water flow problems (Carrera and Neuman, 1986a). Although, as indicated below, it reduces to the least-squares objective function for the parameter-estimation procedure, the maximum-likelihood objective function can be used to calculate statistics with which different models can be compared. The statistics are discussed in the section "Analysis of Results for Nonlinear Problems".

The maximum-likelihood objective function is developed by considering the random nature of $\underline{y}$, the observations. This random nature is a consequence of conceptualizing measurement error as random, as discussed in the section "Weighting Observations". If $\underline{Y}$ is the vector of jointly distributed random variables of which $\underline{y}$ is a realization, the joint probability distribution function (pdf), $f_{\underline{Y}}(\underline{y})$, depends on the true model and true parameter values. For the purpose of estimating parameters for a given assumed model, consider the joint pdf conditioned on a particular set of parameter values, $f_{\underline{Y}}(\underline{y}|\underline{b})$. This joint pdf can be thought of as the probability that different sets of possible observations would occur given the parameter values $\underline{b}$. In parameter estimation, the elements of $\underline{y}$ are known and we would like to estimate $\underline{b}$. A reasonable requirement of the estimates is that they maximize the probability of obtaining the observations, $\underline{y}$. This requirement is imposed by defininig the objective function using the likelihood function, $\ell(\underline{b}|\underline{y})$, which is:

$$\ell(\underline{b}|\underline{y}) = f_{\underline{Y}}(\underline{y}|\underline{b}). \tag{45}$$

If the true errors are from a joint, normal distribution, the likelihood function equals (Brockwell and Davis, 1987, p. 247):

$$\ell(\underline{b}|\underline{y}) = \left(\frac{1}{2\pi}\right)^{ND/2} |\underline{V}(\underline{\epsilon})|^{-1/2} \exp\left\{-\frac{1}{2}\,\underline{e}^T(\underline{V}(\underline{\epsilon}))^{-1}\underline{e}\right\}, \tag{46}$$

where, by equation (44), $\underline{e} = \underline{y} - \hat{\underline{y}}$, $\hat{\underline{y}}$ is a function of $\underline{b}$, and ND is the number of observations. Replacing $\underline{V}(\underline{\epsilon})$ using equation (17), taking the natural log, and multiplying by -2 produces the maximum-likelihood objective function:

$$S'(\underline{b}) = -2\ell n(\ell(\underline{b}|\underline{y})) = ND\ \ell n 2\pi - \ell n\left|\frac{1}{\sigma^2}\,\underline{\omega}\right| + \underline{e}^T\left(\frac{1}{\sigma^2}\,\underline{\omega}\right)\underline{e}, \tag{47}$$

where σ^2 is the common variance, which was defined after equation (17) of this report. Because of the multiplication by a negative number, the maximization problem becomes a minimization problem, and the objective is to determine the parameter estimates that minimize equation (47). To include prior estimates of the parameters, $\underline{e}$ and $\underline{\omega}$ are augmented as in equations

(31) and (33), and ND is replaced by ND+NPR. To calculate equation (47), the determinant is expanded by using equation (33) to yield:

$$S'(\underline{b}) = (ND+NPR)\ \ell n\ 2\pi + (ND+NPR)\ \ell n\ \sigma^2 - \ell n|\underline{\omega}_d| - \ell n|\underline{\omega}_p| + \underline{e}^T(\frac{1}{\sigma^2}\underline{\omega})\underline{e}, \quad (48)$$

where $\underline{\omega}_d$ and $\underline{\omega}_p$ are the sections of the weight matrix applicable to dependent variable observations and prior estimates of the parameters, respectively.

For any assumed model, set of observations, and estimated weight matrix used in the parameter-estimation procedure, $\underline{\omega}$ is replaced by $\hat{\underline{\omega}}$, and ND, σ^2, and $\hat{\underline{\omega}}$ are constant. Eliminating terms of equation (48) that do not depend on $\underline{b}$ and multiplying by σ^2 yields:

$$S(\underline{b}) = \underline{e}^T\ \hat{\underline{\omega}}\ \underline{e}. \quad (49)$$

Thus, for the optimization process, the maximum-likelihood objective function equals the sum-of-squares objective function (eq. 18).

The development of the least-squares objective function from the maximum-likelihood objective function differs from the development of equations (11) or (18) because equation (46) requires that the true errors be from a joint, normal distribution.

The Sum-of-Squares Objective Function in Nonlinear Problems

In nonlinear problems, the sum-of-squares objective function generally is not elliptical as it was for linear problems, as shown in figures 16 and 17. The minimum cannot be determined in one solution of the normal equations, as was possible in linear regression, because the sensitivities change as the parameter values change. Thus, optimal parameter values are determined by an iterative process comprised of repeatedly solving the normal equations. How two parameter values changed in one ground-water flow problem for eight parameter-estimation iterations is shown in figure 17. As discussed in the following sections, the iterations stop when specified convergence criteria are met.

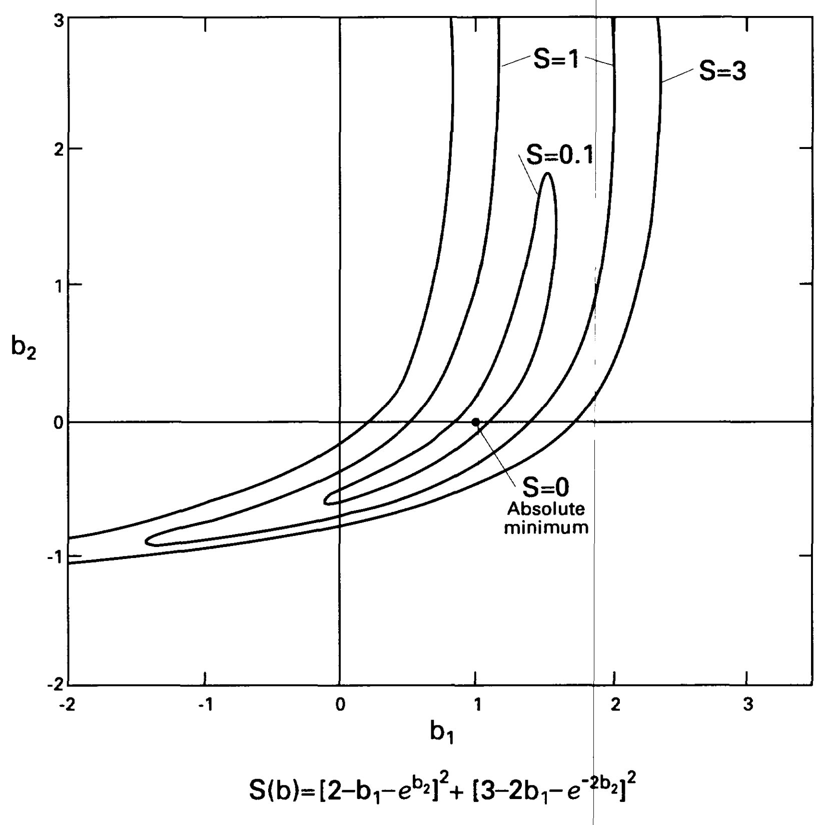


Figure 16.--Sum-of-squares objective function [S(b)] surface for a problem that is nonlinear in parameters β_1 and β_2 (modified from Beck and Arnold, 1977, p. 347; used with permission of the publisher).

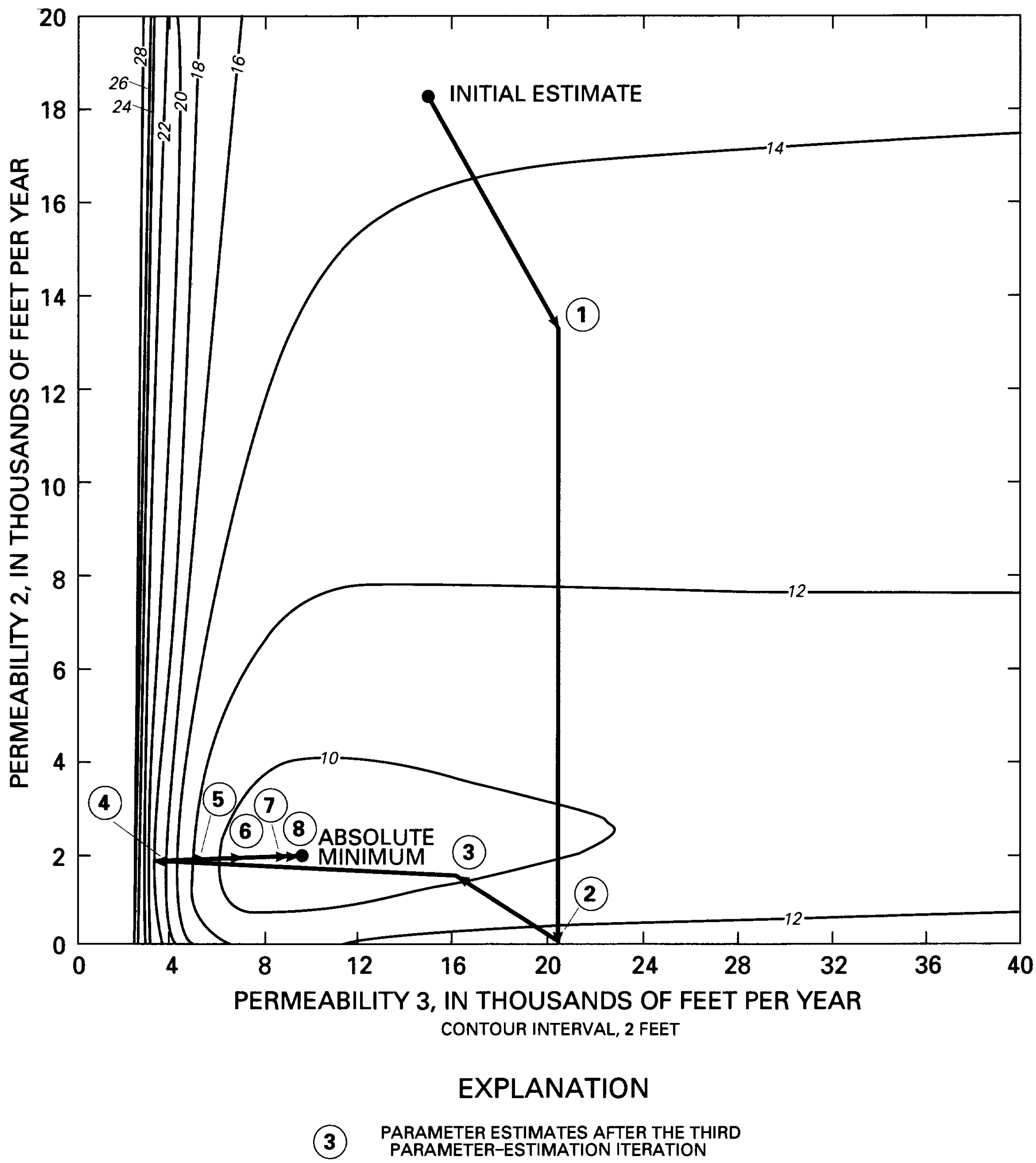


Figure 17.--Sum-of-squares objective function surface for a nonlinear problem showing parameter estimates at parameter-estimation iterations (modified from McLaughlin, 1975, p. VI-9).

The Modified Gauss-Newton Optimization Method and Quasi-Newton Updating

The Parameter-Estimation Package is designed to perform modified Gauss-Newton optimization for any of the problems that the package can represent. Problems with observed head-dependent boundary flow gains and losses and problems with water-table aquifers that go dry at some finite-difference cells for any of the estimated parameter values are included. Modified Gauss-Newton optimization and related methods are discussed by Himmelblau (1972), Bard (1974), Beck and Arnold (1977), Cooley (1983b), and Cooley and Naff (1990), and much of this section is condensed from their discussions.

Iterative Equations

The normal equations and the iterative process for the modified Gauss-Newton optimization method are discussed by Cooley and Naff (1990, ch. 3), and can be expressed as:

$$(\underline{\underline{C}}^T \underline{\underline{X}}_r^T \hat{\underline{\underline{\omega}}} \, \underline{\underline{X}}_r \underline{\underline{C}} + \underline{\underline{I}} \, \mu_r) \underline{\underline{C}}^{-1} \underline{d}_r = \underline{\underline{C}}^T \underline{\underline{X}}_r^T \, \underline{\underline{\omega}} \, (\underline{y} - \hat{\underline{y}}_r) \quad (50a)$$

$$\underline{b}_{r+1} = \rho_r \underline{d}_r + \underline{b}_r \quad (50b)$$

where

- $\underline{\underline{C}}$ is a diagonal scaling matrix with element c_{ii} equal to $(\underline{\underline{X}}_r^T \hat{\underline{\underline{\omega}}} \, \underline{\underline{X}}_r)_{ii}^{-½}$, which produces a scaled matrix with the smallest possible condition number (Forsythe and Strauss, 1955);
- $\underline{\underline{X}}_r$ is the sensitivity matrix evaluated at parameter estimates $\underline{b}_r$;
- r is the parameter-estimation iteration number;
- $\hat{\underline{\underline{\omega}}}$ is the weight matrix;
- $\underline{d}_r$ is a vector used to update the parameter estimates;
- μ_r is the Marquardt parameter (Marquardt, 1963); and
- ρ_r is a damping parameter.

Calculation of the elements of $\underline{\underline{X}}_r$ is described in the section of this report "Sensitivity-Equation Method of Calculating Sensitivities".

For problems with large residuals and a large degree of nonlinearity, Dennis, Gay, and Welsch (1981) suggested substituting $\underline{X}_r^T \hat{\underline{\omega}} \underline{X}_r + \underline{R}_r$ for $\underline{X}_r^T \hat{\underline{\omega}} \underline{X}_r$ in equation (50a) at selected itertions, where $\underline{R}$ is an estimate of the difference between $\underline{X}_r^T \hat{\underline{\omega}} \underline{X}_r$ and the Hessian matrix, and is calculated by quasi-Newton updating as (Dennis, Gay, and Welsch, 1981):

$$\underline{R}_r = \underline{0} \qquad r = 0$$

$$\underline{R}_r = t\underline{R}_{r-1} + \frac{\underline{u}\,\Delta g_r^T + \Delta g_r\,\underline{u}^T}{\underline{d}_{r-1}^T \Delta g_r} - \frac{\underline{d}_{r-1}^T\,\underline{u}\,\Delta g_r \Delta g_r^T}{\left(\underline{d}_{r-1}^T\,\Delta g_r\right)^2} \qquad r > 0 \qquad (51)$$

where

$$\Delta g_r = g_r - g_{r-1}$$

$$g_r = -\,\underline{X}_r^T \hat{\underline{\omega}}\,\underline{e}_r$$

$$\underline{u} = (\underline{X}_r - \underline{X}_{r-1})^T \hat{\underline{\omega}}\,\underline{e}_r - t\underline{R}_{r-1}\underline{d}_{r-1}$$

$$t = \min\left\{\left|\underline{d}_{r-1}^T(\underline{X}_r - \underline{X}_{r-1})^T \hat{\underline{\omega}}\,\underline{e}_r / (\underline{d}_{r-1}^T \underline{R}_{r-1}\underline{d}_{r-1})\right|; 1.0\right\},$$

and all other variables are defined after equations (10) and (50). $\underline{R}_r$ is calculated starting at r=1, but is only included in equation (50a) in later iterations. Performance of the method depends on when $\underline{R}_r$ is included. R.L. Cooley and M.C. Hill (in press) found that generally it is best to include $\underline{R}_r$ after the sum of squared, weighted residuals is no longer changing very much at each parameter-estimation iteration. In the Parameter-Estimation Package, $\underline{R}_r$ is included for all iterations after the sum of squared, weighted residuals decreases by less than a user-defined percentage over two iterations (see SOSR of DATA SET 13 of the INPUT FILE instructions in DATA SET 13), or after a user-specified number of iterations (NFIT of DATA SET 13). The more elaborate criteria for inclusion of $\underline{R}_r$ suggested by Dennis, Gay and Welsch (1981) require additional model simulations. Considering the large problems that are expected to be simulated with MODFLOWP, the more elaborate criteria seemed impractical and were not included in the Parameter-Estimation Package. When $\underline{R}_r$ is included in equation (50a), the elements of the diagonal scaling matrix, $\underline{C}$, are calculated as $(\underline{X}_r^T \hat{\underline{\omega}} \underline{X}_r + \underline{R})_{ii}^{1/2}$.

In the Parameter-Estimation Package, initially $\mu_r=0$ for each parameter-estimation iteration r. For iterations in which the condition described by Cooley and Naff (1990, p. 71-72) is met using $\mu_r=0$, μ_r is increased according to $\mu_r^{new} = 1.5\ \mu_r^{old} + 0.001$ until the condition is no longer met.

The damping parameter, ρ_r, can vary in value from 0.0 to 1.0, and changes the magnitude, but not the direction of $\underline{d}_r$. The damping parameter is used to ensure that the absolute values of fractional parameter value changes, defined for any parameter i as $(b_i^{r+1}-b_i^r)/b_i^r$, where b_i^r is the ith element of vector $\underline{b}_r$, are all less than a value specified by the user (DMAX of Appendix A and C), and to damp oscillations that occur when elements in $\underline{d}_r$ and $\underline{d}_{r-1}$ define opposite directions.

For log-transformed parameter i, requiring that the absolute value of $(b_i^{r+1}-b_i^r)/b_i^r$ be less than DMAX produces inconsistent results, as illustrated by the following example. If the estimated parameter is ℓn K, where K is hydraulic conductivity, and DMAX=2.0, placing the restriction on ℓn K would require that $(\ell n\ K)^{r+1}$ be between $(\ell n\ K)^r-2.0(\ell n\ K)^r$ and $(\ell n\ K)^r+2.0(\ell n\ K)^r$. If K at parameter-estimation iteration r is close to 1.0, say K=1.1, the restriction would require $(\ell n\ K)^{r+1}$ to be between -0.95 and 2.85, so that K^{r+1} would be required to be within the narrow range 0.91 and 1.33. If K at parameter-estimation iteration r is far from 1.0, say $K=1\times10^{-4}$, the restriction requires that $(\ell n\ K)^{r+1}$ be between -27.63 and 9.21, so that K^{r+1} would be allowed to vary within the very wide range of 1×10^{-12} and 2.22. More physically meaningful limitations are produced if the restriction is placed on the exponential of the estimated parameter, which, in this example, is $\exp(\ell n\ K)=K$. Then, in the first situation, DMAX=2.0 would require K to be between 0.0 and 3.3, where the lower limit of 0.0 is a result of estimating a log-transformed parameter, and is always the lower limit for a log-transformed parameter when DMAX $\geq$ 1.0. In the second situation, DMAX = 2.0 would require K to be between 0.0 and 3×10^{-4}.

Calculation of the damping parameter is based on the method described by Cooley (1983b, p. 1274), modified for when one or more of the parameters are log-transformed. In this method, $ADMX_r$ is defined as the maximum absolute value of the fractional parameter changes defined by $\underline{d}_r$. Thus,

$$ADMX_r = \max_i \begin{cases} |d_i^r/b_i^r| & \text{for untransformed parameters} \\ |\exp(d_i^r)-1.0| & \text{for log-transformed parameters} \end{cases} \quad (52)$$

where d_i^r is the ith element of vector $\underline{d}_r$, b_i^r is the ith element of vector $\underline{b}_r$, and b_i^0 replaces b_i^r if $|b_i^r| < |b_i^0|/10^3$. The expression for log-transformed parameters is derived by noting that the fractional change of the physically relevant parameter equals $(\exp(b_i^{r+1}) - \exp(b_i))/\exp(b_i^r)$, which equals $(\exp(b_i^{r+1})/\exp(b_i^r))-1.0$. Substituting $\exp(d_i^r) = \exp(b_i^{r+1})/\exp(b_i^r)$, which is derived from equation (50b) with $\rho_r = 1.0$, yields the expression for log-transformed parameters. An exception to equation (52) occurs if DMAX ≥ 1.0, because, as mentioned previously, the exponential of a log-transformed parameter is always greater than 0.0, and can never decrease enough to require ρ_r to be less than 1.0 if DMAX ≥ 1.0. Thus, if $d_i^r < 0$ for a log-transformed parameter and DMAX ≥ 1.0, parameter i is excluded from consideration in equation (52).

If j is the parameter for which $ADMX_r$ is a maximum, DMX_r is defined as d_j^r/b_j^r if parameter j is untransformed (again, b_j^0 is used if $|b_j^r| < |b_j^0|/10^3$), and $\exp(d_j^r)-1.0$ if parameter j is log-transformed. A preliminary damping parameter, ρ_r^*, is calculated to minimize oscillations according to:

$$\rho_r^* = 1 \qquad r=0 \text{ or j is not the same for iterations r and r+1} \quad (53a)$$

$$\left.\begin{array}{ll} s = DMX_r/(\rho_{r-1}\, DMX_{r-1}) & \\ \text{If } s \geq -1 & \rho_r^* = \dfrac{3+s}{3+|s|} \\ \text{If } s < -1 & \rho_r^* = 1/(2|s|) \end{array}\right\} \; r > 0 \text{ and j is the same for iterations r and r+1,} \quad (53b)$$

where thé condition on j has been added to Cooley's (1983b) method. The damping parameter is then calculated as:

$$\rho_r = \rho_r^* \qquad \text{if } \rho_r^* \text{ ADMX}_r \le \text{DMAX}$$

$$\rho_r = \text{DMAX}/\text{ADMX}_r \qquad \text{if } \rho_r^* \text{ ADMX}_r > \text{DMAX} \qquad (54a)$$

if parameter j is untransformed, and

$$\rho_r = \rho_r^* \qquad \text{if } d_j^r > 0.0 \text{ and } \rho_r^* d_j^r \le \ell n\ (\text{DMAX}+1.0),$$

$$\text{or } d_j^r < 0.0,\ \rho_r^* d_j^r \le \ell n\ (1.0\text{-DMAX}), \text{ and DMAX} < 1.0 \qquad (54b)$$

$$\rho_r = (\ell n\ (\text{DMAX}+1.0))/d_j^r \qquad \text{if } d_j^r > 0.0 \text{ and } \rho_r^* d_j^r > \ell n\ (\text{DMAX}+1.0),$$

$$\rho_r = (\ell n\ (1.0\text{-DMAX}))/d_j^r \qquad \text{if } d_j^r < 0.0,\ \rho_r^* d_j^r > \ell n\ (1.0\text{-DMAX}), \text{ and DMAX} < 1.0.$$

if parameter j is log-transformed. The expressions for log-transformed parameters are derived using the properties discussed after equation (52): the DMAX restriction requires that, if $d_j^r > 0.0$, $(\exp(b_j^{r+1})/\exp(b_j^r))-1.0 \le$ DMAX, or $\rho_r d_j^r \le \ell n$ (DMAX+1.0); if $d_j^r < 0.0$ and DMAX < 1.0, $(\exp(b_j^{r+1})/\exp(b_j^r))-1.0 >$ -DMAX, or $\rho_r d_j^r > \ell n$ (1.0-DMAX). No special restrictions are required to prohibit a large decrease in a log-transformed parameter for DMAX ≥ 1.0 because the log-transform itself restricts the physical value of the parameter to positive values.

Typically DMAX is larger than 1.0 and less than about 2.0. Values less than 1.0 can be used if parameter-value oscillations are a problem, but note that values less than 1.0 prohibit parameter values from changing sign.

As discussed by Cooley and Naff (1990, p. 70), modified Gauss-Newton optimization typically converges within "***a number of iterations equal to five or twice the number of parameters, whichever is greater." Convergence will tend to occur sooner for well-conditioned problems, and later for poorly conditioned problems.

Using double precision as suggested by Stewart (1972, p. 226-227), equation (50a) has been solved accurately and efficiently in many applications of MODFLOWP using Cholesky LDL^T decomposition (Dennis and Schnabel, 1983, p. 50-51). Exceptions were plagued by strong correlations between parameters or insensitive parameters, and were resolved by reparameterization. Dennis and Schnabel (1983, p. 221) and Seber and Wild (1989, p. 621) suggest that solving the alternative formulation $\underline{\underline{X}}\ \underline{d} = (\underline{y}-\hat{\underline{y}})$ using QR or singular-value decomposition (Dennis and Schnabel, 1983, p. 49-51; Seber and Wild, 1989, p. 680-681; Press and others, 1989, p. 52-63) is more stable, but it is unclear whether or not they used the scaling and Marquardt parameter which adds stability to equation (50a). Press and others (1989, p. 515-520) suggest using singular-value decomposition for linear regression, but use Gauss-Jordon elimination to solve a variation of equation (50a) that includes similar scaling and implementation of the Marquardt parameter for nonlinear regression. Considering the success experienced using Cholesky decomposition and the lack of any clear indication that the alternatives are better, Cholesky decomposition is used in the Parameter-Estimation Package of this report.

Convergence Criteria

Convergence of the modified Gauss-Newton iterative process is achieved when either of two criteria are satisfied. The first convergence criterion is based on the fact that, for well-posed problems, the iterative process defined by equations (50a) and (50b) produces parameter estimates, $\underline{b}_r$, that are progressively closer to the values associated with the minimum of the sum-of-squares objective function. As this minimum is reached, the right-hand side of (50a) and, therefore, $\underline{d}_r$ approach zero. Thus, the first criterion is that the largest absolute value of d_i/b_i, i=1,NP, is less than a user-defined convergence criterion (TOL of Appendices A and C), where d_i is the ith element of $\underline{d}_r$ and b_i is the ith element of $\underline{b}_r$(eq. 50a). For log-transformed parameters, a process similar to the one described above produces the requirement that the largest absolute value of $\exp(d_i)-1$, i=1,NP, needs to be less than the convergence criterion.

The second convergence criterion is satisfied if the sum of squared, weighted residuals does not change by more than a user-defined percentage over three consecutive iterations. This criterion allows parameter estimation to be completed and final statistics to be printed when parameters vary in the vicinity of the minimum, but never quite satisfy the first convergence criterion. This most frequently occurs when there is at least one insensitive parameter. Insensitive parameters generally have large variances, and the user needs to consider reparameterizing the problem if there are insensitive parameters.

Conjugate-Direction Optimization Methods

The Parameter-Estimation Package is designed to perform conjugate-direction optimization when hydraulic heads or changes in hydraulic heads are the only dependent-variable observations, and water-table aquifers remain saturated at all finite-difference cells for all estimated parameter values. These restrictions are not required for conjugate-direction methods, but the program presented in this report does not support the use of conjugate-direction methods on these more complicated problems. Conjugate-direction optimization is discussed by Himmelblau (1972), Neuman (1980, 1982), Cooley (1985), and Carrera and Neuman (1986b), and this section is condensed from their discussions.

Neuman (1980, 1982) and Carrera and Neuman (1986b) use the term conjugate gradient to describe the methods classified as conjugate direction in this report. A change in terminology is justified because the methods are not truly conjugate-gradient methods, as defined by Himmelblau (1972, p. 88). The choice of the term conjugate direction is derived from Himmelblau (1972, p. 105).

Iterative Equations

Conjugate-direction optimization is accomplished by using the gradient vector, $\underline{g}_r$, which has components equal to:

$$g_i = \frac{\partial S(\underline{b}_r)}{\partial b_i} \qquad i = 1,NP \qquad (55)$$

where r is the parameter-estimation iteration number. The gradient is equivalent to the unscaled right-hand side of equation (50a) multiplied by -2, but is calculated by the adjoint-state method, which is discussed in the section of this report "Adjoint-State Method of Calculating the Gradient of the Objective Function". The adjoint-state method of calculating the gradient takes much less computer time in most problems than using a method that requires the sensitivity matrix, $\underline{X}$. To define the direction of parameter change, $\underline{d}_r$, the gradient vector is modified by an approximation of $(\underline{X}^T\hat{\underline{\omega}}\,\underline{X})^{-1}$, the inverse of the unscaled matrix on the left-hand side of equation (50a) with μ_r=0. An approximation is used to avoid calculating the sensitivity matrix, $\underline{X}$. The Fletcher-Reeves and quasi-Newton methods discussed below differ in how the matrix inverse is approximated. Carrera and Neuman (1986b) also discuss a combined method in which the Fletcher-Reeves method is used in the first parameter-estimation iterations, and the quasi-Newton method is used in later iterations. In the Parameter-Estimation Package, the user can specify how many Fletcher-Reeves iterations to use in the combined method.

To produce a better-conditioned problem, $\underline{g}$ is scaled as $\underline{C}\,\underline{g}$, $\underline{d}$ is scaled as $\underline{C}^{-1}\underline{d}$, and the matrix to be approximated is scaled as $(\underline{C}^T\,\underline{X}_r^{\,T}\,\hat{\underline{\omega}}\,\underline{X}_r\,\underline{C})^{-1}$ where the diagonal elements of the scaling matrix, $\underline{C}$, are defined by the user. The performance of the conjugate-direction methods is substantially affected by the scaling. Ideally, the c_{ii} would be equal to the values used in modified Gauss-Newton optimization (discussed after eq. 50a), but because $\underline{X}$ is unknown, those values can not be calculated for conjugate-direction optimization. Scale factors are similar in value to the related parameter (Cooley, 1985, tables 1, 4, and 9), but optimal values can be determined only by trial and error. Taking the natural log of the parameter also is a form of scaling and might decrease the number of parameter-estimation iterations required for some problems.

For each parameter-estimation iteration, the updated parameter estimates are calculated as:

$$\underline{b}_{r+1} = \rho_r\underline{d}_r + \underline{b}_r, \qquad (56)$$

where ρ_r^{*} is calculated by Newton's method, as described below. If the maximum fractional parameter change calculated by equation (56) is greater than the value specified by DMAX of the INPUT FILE (see Appendix A), ρ_r is decreased using the method described for the modified Gauss-Newton method (eqs. 52, 53, and 54), except that $\rho_r^* = 1.0$ for all iterations.

Conjugate-direction optimization typically requires many more iterations than is required by modified Gauss-Newton optimization. Even simple problems might require about 50 iterations (Cooley, 1985; Hill, 1990a), and 100 iterations might not be excessive for some problems.

Fletcher-Reeves Method of Approximating the Matrix Inverse

In the Fletcher-Reeves method, the scaled direction of parameter change is calculated as:

$$\underline{C}^{-1}\underline{d}_{r+1} = -\underline{C}g_r + \alpha_r \underline{C}^{-1}\underline{d}_r \qquad (57)$$

$$\alpha_r = 0 \qquad r = n\times NP$$

$$\alpha_r = \frac{(\underline{C}g_r)^T \underline{C}g_r}{(\underline{C}g_{r-1})^T \underline{C}g_{r-1}} \quad r \neq n\times NP,$$

where

g_r is the gradient calculated by using parameter estimates $\underline{b}_r$;

$\underline{d}_r$ is the direction of parameter change for iteration r;

α_r is an iteration parameter for iteration r; n=0,1,2 •••; and

NP is the number of estimated parameters.

Equation (57) can be rewritten as (modified from Cooley, 1985, p. 1531):

$$\underline{C}^{-1}\underline{d}_{r+1} = -\left[\underline{I} - \frac{\underline{C}^{-1}\underline{d}_r(\underline{C}g_r)^T}{(\underline{C}g_{r-1})^T \underline{C}g_{r-1}}\right]\underline{C}g_r, \qquad (58)$$

where $\underline{I}$ is an NP by NP identity matrix. The approximation of the inverse of the matrix on the left-hand side of equation (50a) with μ_r=0 is the expression multiplying $\underline{C}g_r$ on the right-hand side of equation (58).

Quasi-Newton Method of Approximating the Matrix Inverse

In the quasi-Newton method, the direction of parameter change is calculated as:

$$\underline{C}^{-1}\underline{d}_{r+1} = -\underline{H}_r \underline{C}g_r, \qquad (59)$$

where

$$\underline{\underline{H}}_r = \text{diag}((b_1/c_{1,1})^2, (b_2/c_{2,2})^2, \cdots (b_{NP}/c_{NP,NP})^2) \qquad r = 0$$

$$\underline{\underline{H}}_r = \left[\underline{\underline{H}}_{r-1} - \frac{\underline{\underline{H}}_{r-1}\,\underline{\underline{C}}(\Delta g_r)\underline{d}_r^T\,\underline{\underline{C}}^{-1} + \underline{\underline{C}}^{-1}\underline{d}_r(\Delta g_r)^T\underline{\underline{C}}\,\underline{\underline{H}}_{r-1}}{\underline{d}_r^T\,(\Delta g_r)}\right]\gamma_r$$

$$+ \left[\rho_{r-1} + \frac{\gamma_r(\Delta g_r)^T\underline{\underline{C}}\,\underline{\underline{H}}_{r-1}\,\underline{\underline{C}}(\Delta g_r)}{\underline{d}_r^T\,(\Delta g_r)}\right]\frac{\underline{\underline{C}}^{-1}\,\underline{d}_r^T\,\underline{d}_r\underline{\underline{C}}^{-1}}{\underline{d}_r^T\,(\Delta g_r)} \qquad r > 0$$

$$\Delta g_r = g_r - g_{r-1}$$

$$\gamma_r = \frac{\rho_{r-1}\,\underline{d}_r^T\,(\Delta g_r)}{(\Delta g_r)^T\underline{\underline{C}}\,\underline{\underline{H}}_{r-1}\,\underline{\underline{C}}(\Delta g_r^T)} \qquad r = 1$$

$$\gamma_r = 1 \qquad r > 1,$$

and ρ_{r-1} is the iteration parameter calculated using Newton's method, as described below.

Newton's Method of Calculating the Iteration Parameter

Once the direction of parameter change is calculated, the updated parameter values are calculated using equation (56). Calculation of ρ_r by Newton's method (Neuman, 1980, p. 337) is accomplished in three steps. First, $\partial\underline{h}/\partial\rho_r$, the derivative of hydraulic head with respect to the iteration parameter, is calculated by solving:

$$\underline{\underline{A}}(n)\,\frac{\partial\underline{h}(n)}{\partial\rho_r} = \underline{\underline{B}}(n)\,\frac{\partial\underline{h}(n-1)}{\partial\rho_r} - \underline{a}(n), \qquad (60)$$

where $\underline{\underline{A}}(n)$, $\underline{h}(n)$, and $\underline{\underline{B}}(n)$ were defined after equation (3), and

$$\underline{a}(n) = \sum_{i=1}^{NP} d_i \left[\frac{-\partial\underline{\underline{A}}(n)}{\partial b_i}\,\underline{h}(n) + \frac{\partial\underline{\underline{B}}(n-1)}{\partial b_i}\,\underline{h}(n-1) - \frac{\partial\underline{f}(n)}{\partial b_i}\right], \qquad (61)$$

where, d_i is the i^{th} element of $\underline{d}_r$. For steady-state conditions (n=0), $\underline{\underline{B}}(n)$ of equations (60) and (61) equals zero, and the storage terms in $\underline{\underline{A}}$ equal zero.

In the second step, the $\partial \underline{h}/\partial \rho_r$ are used to calculate $\partial \hat{\underline{y}}/\partial \rho_r$ by using the methods used to calculate $\hat{\underline{y}}$ from $\underline{h}$ discussed in the "Observations" section of this report, where the $\partial \underline{h}/\partial \rho_r$ and $\partial \hat{\underline{y}}/\partial \rho_r$ are substituted for $\underline{h}$ and $\hat{\underline{y}}$. The resulting ND-length vector is used to accomplish the third step of Newton's method, in which:

$$\rho_r = \frac{\underline{t}_r^T \hat{\underline{\underline{\omega}}}_d \underline{e}_d + \underline{d}_{r+1}^T \hat{\underline{\underline{\omega}}}_p \underline{e}_p}{\underline{t}_r^T \hat{\underline{\underline{\omega}}}_d \underline{t}_r + \underline{d}_{r+1}^T \hat{\underline{\underline{\omega}}}_p \underline{\underline{C}} \underline{d}_{r+1}}, \tag{62}$$

where, $\underline{t}_r = \partial \hat{\underline{y}}/\partial \rho_r$; $\underline{e}_d$ is the partition of $\underline{e}$ which relates to the dependent-variable data; $\hat{\underline{\underline{\omega}}}_d$ is the partition of $\hat{\underline{\underline{\omega}}}$ which relate to the dependent variable data; $\underline{e}_p$ is the partition of $\underline{e}$ which relates to the prior estimates of the parameters; and $\hat{\underline{\underline{\omega}}}_p$ is the partition of $\hat{\underline{\underline{\omega}}}$, which relates to the prior estimates of the parameters (see section "Prior Information on the Estimated Parameters").

Convergence Criteria and Iteration Control

The conjugate-direction optimization methods are said to converge when the absolute value of the largest element of $\underline{g}_r$ (eq. 55) and the absolute value of the largest element of $\rho_r \underline{d}_r$ divided by the corresponding element of $\underline{b}$, (eq. 56) are smaller than a user-defined convergence criterion.

At each conjugate-direction iteration, two items are checked to ensure that the parameter-estimation procedure is progressing satisfactorily. First, the value of the objective function is checked. If the value increases by more than 0.0001 (a small nonzero value is used to accommodate round-off error), it is assumed that the iteration parameter calculated by Newton's method is too large. A new set of parameters is calculated as:

$$\underline{b}'_{r+1} = \underline{b}_{r+1} - \sum_{m=1}^{n} \frac{1}{2^m} \rho_r \underline{d}_r, \tag{63}$$

where n is the number of times that the change in the parameters has been reduced and m is a dummy variable used in the summation. As many as five reductions might occur within a single parameter-estimation iteration. The iteration process continues if the new parameters produce an objective-function value that is no more than 0.0001 larger than the objective-function value from the previous iteration. If the objective-function value has not satisfied this criterion after five reductions, execution stops. [The above discussion is modified from Neuman (1980) and R. L. Cooley (U.S. Geological Survey, written commun., 1989.)

Second, the progress of quasi-Newton iterations is checked by comparing the change in the gradient between iterations ($g_r - g_{r-1}$) and the parameter change direction vector, $\underline{d}_{r-1}$. If the value of $(g_r - g_{r-1})^T \underline{d}_{r-1}$ is zero or negative, $\underline{d}_{r-1}$ is not a descent direction, and $\underline{\underline{H}}$ is not updated for that iteration (Shanno, 1978; R.L. Cooley, written commun., 1989).

<u>Relative Efficiency of the Optimization Methods</u>

The modified Gauss-Newton and conjugate-direction optimization methods discussed in this report have been compared by using steady-state ground-water flow test cases (Cooley, 1985), and using steady-state and transient ground-water flow test cases (Hill, 1990a). Their results indicate that the modified Gauss-Newton method estimates parameters by using much less computer processing time (less than 50 percent with a direct D4 solver; less than 80 percent with a conjugate-gradient solver with the modified incomplete Cholesky preconditioner), and slightly more computer storage (as much as 6 percent) than the conjugate-direction optimization methods. The decrease in computer storage is lost if statistics requiring the calculation of the sensitivity matrix, $\underline{\underline{X}}_r$, are calculated for the conjugate-direction methods. The modified Gauss-Newton method also has the advantage of not requiring user-defined scaling. Despite its apparent inefficiency and disadvantages, conjugate-direction optimization was retained in the Parameter-Estimation Package so it could be compared with the modified Gauss-Newton method for a greater range of problems.

Odenwald and Herrling (1990) optimized a regression problem using several iterations of a conjugate-direction optimization method followed by modified Gauss-Newton optimization iterations. For the final iterations, a full Newton optimization method (Gill and others, 1981) was used. Although Odenwald and Herrling (1990) did not compare this optimization procedure with using each method individually, it is likely that combined methods produce more efficient optimization, and this is a promising area of further research.

R.L. Cooley and M.C. Hill (in press) showed that using the quasi-Newton updating of equation (51) can dramatically reduce the number of iterations required to estimate parameters for problems in which modified Gauss-Newton converges slowly. Convergence is considered to be slow if the number of parameter-estimation iterations required exceeds the number of estimated parameters. If modified Gauss-Newton converges quickly, there is no advantage to using the quasi-Newton updating.

Analysis of Results for Nonlinear Problems

When using nonlinear regression, final parameter estimates and the assumed model can be analyzed by using the same methods used for linear models (see "Analysis of Results for Linear Problems" of this report). When optimizing with the modified Gauss-Newton method, the statistics calculated by using the sensitivity matrix, $\underline{\underline{X}}$, such as variance-covariance matrix for the final parameters (eq. 35), are generally calculated by using the sensitivity matrix from the last parameter-estimation iteration ($\underline{\underline{X}}_r$ of eq. 50a). When optimizing with a conjugate-direction method, these statistics are available only if the sensitivity matrix is calculated by using the final parameter estimates. This capability is provided as an option in the Parameter-Estimation Package.

Nonlinearity produces several problems in the analysis of results. First, in nonlinear problems, the estimated error variance, s^2, might not be an unbiased estimate of the true error variance, even if the model is correct (Draper and Smith, 1981, p. 484), and unbiasedness generally is not possible in nonlinear problems (Bard, 1974, p. 40). Fortunately, for typical ground-water flow problems the bias caused by nonlinearity is not large, and s^2 can be used as described for linear problems. Second, the confidence intervals on parameter values for linear problems (eq. 36) might not be valid (Donaldson and Schnabel, 1987), and alternative methods, such as those described by Vecchia and Cooley (1987) might need to be used. The alternative methods are not included as part of the Parameter-Estimation Package presented in this report.

Additional statistics developed with the maximum-likelihood objective function can be used for model discrimination. The first statistic is the value of the maximum-likelihood objective function itself (eq. 48), evaluated at the minimized parameter values, b'. In many circumstances, smaller values of $S'(\underline{b}')$ indicate better models. As more parameters are added, however, the residuals, $\underline{e}$, and, therefore, $S'(\underline{b})$, generally become smaller, and the user might draw the incorrect conclusion that models with more parameters generally are better. To reflect the fact that adding too many parameters produces unreliable parameter estimates, two additional statistics have been developed that are equal to the sum of $S'(\underline{b}')$ and terms that become large as more parameters are added. Although these statistics were developed for time-series problems, Carrera and Neuman (1986a and c) successfully used them to discriminate between different parameterizations of a test case of ground-water flow. The statistics are stated below; see the cited references for their derivations and additional discussion.

The second statistic (AIC) was developed by Akaike (1974) and equals:

$$AIC(\underline{b}') = S'(\underline{b}') + 2 \times NP. \qquad (64)$$

The third statistic (BIC) also was developed by Akaike (1978) as a response to concern that $AIC(\underline{b}')$ sometimes promoted use of more parameters than was required. The version of this statistic used by Carrera and Neuman (1986a and c) is:

$$BIC(\underline{b}') = S'(\underline{b}') + NP \times \ell n(ND+NPR+MPR). \qquad (65)$$

For both statistics, smaller values indicate a more accurate model. If the statistics for a model with fewer parameters are only slightly larger than the statistics of another model, however, it might be better to select the model with fewer parameters, unless the investigator has other information indicating the validity of the more complicated model.

Dry Cells in Water-Table Layers

If the top model layer is designated as a water-table layer (Block-Centered Flow Package: LAYCON=1), some cells might go dry and become inactive. In the Parameter-Estimation Package, only the modified Gauss-Newton optimization method with ISN=-1 (LINE 6 of the INPUT FILE) is designed to accommodate dry cells. At the beginning of each parameter-estimation iteration, the hydraulic-conductance and IBOUND arrays (McDonald and Harbaugh, 1988) are adjusted to make all dry cells active again, so conditions are the same at the beginning of each parameter-estimation iteration.

CALCULATING SENSITIVITY-EQUATION SENSITIVITIES AND THE GRADIENT OF THE OBJECTIVE FUNCTION

Modified Gauss-Newton optimization requires ND derivatives with respect to each of the parameters. These derivatives, or sensitivities, can be expressed as $\partial \hat{y}_q/\partial b_\ell$, q=1,ND, ℓ=1,NP, and are the elements of an ND by NP matrix. Conjugate-direction optimization requires one derivative with respect to each of the parameters. These derivatives can be expressed as $\partial S(\underline{b})/\partial b_\ell$, ℓ=1,NP, and are the elements of the gradient of the objective function, $\underline{g}_r$ (eq. 55), which is a vector. Yeh (1986) showed that when the required number of derivatives with respect to each parameter was greater than the number of parameters, the derivatives could be calculated most efficiently using the sensitivity-equation method. Thus, the sensitivity-equation method is used to calculate the derivatives required for modified Gauss-Newton optimization. When the required number of derivatives with respect to each parameter is less than or equal to the number of parameters, Yeh (1986) showed that the adjoint-state method is most efficient. Thus, the adjoint-state method is used to calculate the derivatives required for conjugate-direction optimization. The sensitivity-equation method and the adjoint-state method are described in the following sections.

Sensitivity-Equation Method of Calculating Sensitivities

In modified Gauss-Newton optimization, the elements of the first ND rows of the sensitivity matrix ($\underline{\underline{X}}_r$ of eq. 50a) equal $\partial \hat{y}_q/\partial b_\ell$, q=1,ND,$\ell$=1,NP, where $\hat{y}_q$ is an element of vector $\hat{\underline{y}}$ (eq. 44). The columns of this partition of $\underline{\underline{X}}_r$ can each be expressed as the vector $\partial \hat{\underline{y}}/\partial b_\ell$. For simplicity in the following discussion, the subscript q is omitted, and a single component of the vector $\partial \hat{\underline{y}}/\partial b_\ell$ is represented by $\partial \hat{y}/\partial b_\ell$. In the sensitivity-equation method, these elements are calculated from sensitivity-equation sensitivities, $\partial \underline{h}/\partial b_\ell$, ℓ=1,NP, where $\underline{h}$ is defined after equation (3) and is a vector of hydraulic heads at every grid cell center and time step. For observations of hydraulic heads and changes in hydraulic heads, the $\partial \hat{\underline{y}}/\partial b_\ell$ are calculated from the $\partial \underline{h}/\partial b_\ell$ exactly as the $\hat{\underline{y}}$ are calculated from the $\underline{h}$, as described in the section entitled "Observations".

For observations of head-dependent boundary gains and losses, the sensitivities are calculated by taking the derivative of equation (4). For parameters, such as transmissivity, that are not directly involved in the calculation of a gain or loss along the head-dependent boundary:

$$\partial\hat{y}/\partial b_\ell = \sum_{i=1}^{NQCL} \frac{\partial q_i}{\partial b_\ell} = -\sum_{i=1}^{NQCL} \frac{K_i}{D_i} A_i \frac{\partial h_i}{\partial b_\ell} \qquad (66)$$

where all terms were defined after equation (4). Two examples are used to illustrate how $\partial\hat{y}/\partial b_\ell$ is calculated when b_ℓ is a function of K_i (b_ℓ=f(K_i)). First, if b_ℓ=K_i for all NQCL cells of the head-dependent boundary:

$$\partial\hat{y}/\partial b_\ell = \sum_{i=1}^{NQCL} \left[-\frac{K_i}{D_i} A_i \frac{\partial h_i}{\partial b_\ell} + \frac{A_i}{D_i} (H_i - h_i) \right]. \qquad (67)$$

Second, if b_ℓ = K_i/D_i for all NQCL cells of the head-dependent boundary:

$$\partial\hat{y}/\partial b_\ell = \sum_{i=1}^{NQCL} \left[-\frac{K_i}{D_i} A_i \frac{\partial h_i}{\partial b_\ell} + A_i (H_i - h_i) \right]. \qquad (68)$$

Three circumstances can affect the calculated $\partial\hat{y}/\partial b_\ell$, all of which decrease the effect of the head-dependent boundary gain-or-loss observation on parameter estimation.

First, if the head-dependent boundary is simulated using the Streamflow-Routing or River Package, and the simulated hydraulic head, h_i, is below the bottom of the streambed, the ith term of $\partial\hat{y}/\partial b_\ell$ equals 0.0 unless b_ℓ is a function of K_i (b_ℓ=f(K_i)). If b_ℓ=K_i, the ith term of $\partial\hat{y}/\partial b_\ell$ equals $(A_i/D_i)(H_i\text{-}RBOT_i)$, where $RBOT_i$ is the elevation of the bottom of the streambed.

Second, if the head-dependent boundary is simulated using the Drain Package, and the simulated hydraulic head, h_i, is below the elevation of the drain, the i^{th} term of $\partial\hat{y}/\partial b_\ell$ equals 0.0 for all b_ℓ.

Third, if the head-dependent boundary is simulated by using the Streamflow-Routing Package, and the loss in streamflow exceeds the flow entering reach i, the ith term of $\partial\hat{y}/\partial b_\ell$ equals 0.0 for all b_ℓ.

In the extreme, these conditions can eliminate the effect of the head-dependent boundary gain-or-loss observation on parameter estimation. For example, if the simulated hydraulic heads at all cells or reaches used to simulate a head-dependent boundary are below the bottom of the streambed, or drain, the observation has no effect on estimated parameter values unless $b_\ell = f(K_i)$, in which case only the estimation of parameter b_ℓ would be affected.

The $\partial \underline{h}/\partial b_\ell$ are calculated from the numerical solution of the sensitivity equation. If the temporal discretization of the ground-water flow equation is fully implicit, as in MODFLOW and MODFLOWP, the matrix form of the sensitivity equations for the ℓ^{th} parameter of a problem in which all aquifers are confined is:

$$\underline{\underline{A}}(0) \frac{\partial \underline{h}(0)}{\partial b_\ell} = - \frac{\partial \underline{\underline{A}}(0)}{\partial b_\ell} \underline{h}(0) - \frac{\partial \underline{f}(0)}{\partial b_\ell} \qquad n = 0, \qquad (69a)$$

and

$$\underline{\underline{A}}(n) \frac{\partial \underline{h}(n)}{\partial b_\ell} = \frac{\partial \underline{\underline{B}}(n)}{\partial b_\ell} \underline{h}(n-1) + \underline{\underline{B}}(n) \frac{\partial \underline{h}(n-1)}{\partial b_\ell} \qquad (69b)$$

$$- \frac{\partial \underline{\underline{A}}(n)}{\partial b_\ell} \underline{h}(n) - \frac{\partial \underline{f}(n)}{\partial b_\ell} \qquad n \geq 1,$$

where $\underline{\underline{A}}(0)$ equals $\underline{\underline{K}} + \underline{\underline{P}}(0)$ $[L^2/T]$;
- b_ℓ is the ℓ^{th} parameter;
- $\underline{f}(0)$ is the forcing function of the steady-state problem (n=0); and
- $\underline{f}(n)$ is the forcing function at the n^{th} time step.

All terms except b_ℓ were defined after equation (3). Note that $\underline{\underline{A}}(0)$, $\underline{h}(0)$, and $\underline{f}(0)$ are just $\underline{\underline{A}}(n)$, $\underline{h}(n)$, and $\underline{f}(n)$ evaluated for n=0 (steady state). The boundary conditions are:

$$\frac{\partial \underline{h}}{\partial b_\ell}\bigg|_{\Gamma_1} = 0, \qquad (70)$$

$$H \bigg|_{\Gamma_2} = 0,$$

where Γ_1 are the constant-head boundaries of the physical system, except when b_ℓ is used to define the hydraulic head along a constant-head boundary. In that situation, $\partial \underline{h}/\partial b_\ell$ along the boundary is calculated from the relation between the constant heads and b_ℓ. Γ_2 are the head-dependent boundaries and H is the constant head on one side of the boundary.

By using equation (69) and (70), $\partial \underline{h}/\partial b_\ell$ for each time step can be calculated for all parameters before progressing to the next time step. By using this method, $\underline{\underline{A}}$ is formulated only once for each time step, but solutions of hydraulic heads and sensitivities for all parameters needs to be saved for use in the next time step. Alternatively, sensitivities for all time steps can be calculated for each parameter before progressing to the next parameter. By using this method, $\underline{\underline{A}}$ is formulated more times, but the quantity of computer temporary storage required is smaller if the number of parameters is less than the number of time steps. Both methods are available in the Parameter-Estimation Package by appropriate specification of ISN of LINE 6 of the INPUT FILE (Appendix A).

If any of the layers are unconfined, $\underline{\underline{K}}$, and, therefore, the $\underline{\underline{A}}$ are functions of $\underline{b}$ and $\underline{h}(\underline{b})$, and the first term on the right-hand side of equation (69a) and the third term on the right-hand side of equation (69b) need to be expanded. By using iterative updating as suggested by Shah and others (1978), to avoid solving a problem with an unsymmetric matrix, the equations for problems with unconfined aquifers are:

$$\underline{\underline{A}}(0) \left(\frac{\partial \underline{h}(0)}{\partial b_\ell}\right)^r = - \frac{\partial \underline{\underline{A}}(0)}{\partial \underline{h}(0)} \left(\frac{\partial \underline{h}(0)}{\partial b_\ell}\right)^{r-1} \underline{h}(0) - \frac{\partial \underline{\underline{A}}(0)}{\partial b_\ell} \underline{h}(0) - \frac{\partial \underline{f}(0)}{\partial b_\ell}, \quad n = 0, \quad (71a)$$

and

$$\underline{\underline{A}}(n) \left(\frac{\partial \underline{h}(n)}{\partial b_\ell}\right)^r = - \frac{\partial \underline{\underline{A}}(n)}{\partial \underline{h}(n)} \left(\frac{\partial \underline{h}(n)}{\partial b_\ell}\right)^{r-1} \underline{h}(n) + \frac{\partial \underline{\underline{B}}(n)}{\partial b_\ell} \underline{h}(n-1) +$$

$$\underline{\underline{B}}(n) \frac{\partial \underline{h}(n-1)}{\partial b_\ell} - \frac{\partial \underline{\underline{A}}(n)}{\partial b_\ell} \underline{h}(n) - \frac{\partial \underline{f}(n)}{\partial b_\ell}, \qquad n \geq 1, \quad (71b)$$

where the equations need to be solved with an iterative solver such as PCG2 (Hill, 1990b), and r is the solver iteration. All other variables were defined after equations (3) or (69). By using index notation to clarify the multiplications involved, the first term on the right-hand side of equations (71a) and (71b) equal:

$$\frac{\partial \underline{\underline{A}}(0)}{\partial \underline{h}(0)} \left[\frac{\partial \underline{h}(0)}{\partial b_\ell}\right]^{r-1} \underline{h}(0) = \frac{\partial A_{ij}(0)}{\partial h_m(0)} \left[\frac{\partial h_m(0)}{\partial b_\ell}\right]^{r-1} h_j(0) \tag{72a}$$

and

$$\frac{\partial \underline{\underline{A}}(n)}{\partial \underline{h}(n)} \left[\frac{\partial \underline{h}(n)}{\partial b_\ell}\right]^{r-1} \underline{h}(n) = \frac{\partial A_{ij}(n)}{\partial h_m(n)} \left[\frac{\partial h_m(n)}{\partial b_\ell}\right]^{r-1} h_j(n), \tag{72b}$$

which are vectors.

Sensitivity-equation sensitivites for different parameters might vary from hydraulic-head values and from each other by several orders of magnitude. When using an iterative solver, the convergence criteria specified for hydraulic head needs to be adapted for each parameter. In this report, the convergence criteria for the sensitivity-equation sensitivities for a parameter b_ℓ are calculated as CONV/($|b_\ell^0|\times100$), where CONV is a convergence criterion for hydraulic head, and b_ℓ^0 is the initial estimate of the parameter. For example, when using PCG2 (Hill, 1990b), the convergence criteria for b_ℓ are calculated as HCLOSE/($|b_\ell^0|\times100$) and RCLOSE/($|b_\ell^0|\times100$).

Adjoint-State Method of Calculating the Gradient of the Objective Function

In adjoint-state theory, adjoint states are calculated first, and then are used to calculate the gradient. Adjoint states are obtained by solving differential equations that are developed from, and are very similar to, the ground-water flow equations. If the temporal discretization of the ground-water flow equation is fully implicit, as in MODFLOW and MODFLOWP, the matrix form of the adjoint-state equations are (after Townley and Wilson, 1985):

$$\underline{\underline{A}}(n)\ \underline{\phi}(n-1) = \underline{\underline{B}}(n+1)\ \underline{\phi}(n) + \frac{\partial S(\underline{b})}{\partial \underline{h}(n)} \qquad n > 0 \tag{73a}$$

$$\underline{\underline{A}}(0)\ \underline{\phi}(-1) = \underline{\underline{B}}(1)\ \underline{\phi}(0) + \frac{\partial S(\underline{b})}{\partial \underline{h}(0)} \qquad n = 0, \tag{73b}$$

where $S(\underline{b})$ is the objective function;

$\frac{\partial S(\underline{b})}{\partial \underline{h}(n)}$ is a vector of zeros unless components of $S(\underline{b})$ are at time step n; and

$\underline{\phi}(n)$ is the adjoint state at the end of the n^{th} time step $[T/L^2]$.

All other terms were defined after equation (3). The boundary conditions are:

$$\begin{aligned} \phi \Big|_{\Gamma_1} &= 0 \\ \left(\frac{\partial \underline{\phi}}{\partial x} n_x + \frac{\partial \underline{\phi}}{\partial y} n_y + \frac{\partial \underline{\phi}}{\partial z} n_z\right)\Big|_{\Gamma_2} &= 0, \end{aligned} \tag{74}$$

where Γ_1 are the constant-head boundaries of the physical system;

Γ_2 are the defined-flux boundaries of the physical system; and

n_x, n_y, n_z are the components of an inward-looking unit vector along Γ_2.

The initial condition is $\underline{\phi}(N)=0$, where N is the latest time step at which a component of $S(\underline{b})$ occurs. Equations (73) and (74) are solved by starting at n=N, and working backwards to $\underline{\phi}(-1)$. In a sense, the system is solved backwards in time, except that different time-step lengths are used on the two sides of the equation.

The derivative with respect to the ℓ^{th} parameter is calculated by (Townley and Wilson, 1985):

$$\frac{\partial S(\underline{b})}{\partial b_\ell} = \frac{\partial S_f(\underline{b})}{\partial b_\ell} + \frac{\partial S_p(\underline{b})}{\partial b_\ell} + \underline{\phi}(-1) \left\{ - \frac{\partial \underline{\underline{A}}(0)}{\partial b_\ell} \underline{h}(0) - \frac{\partial \underline{f}(0)}{\partial b_\ell} \right\}$$

$$+ \sum_{n=1}^{N} \underline{\phi}(n-1) \left\{ - \frac{\partial \underline{\underline{A}}(n)}{\partial b_\ell} \underline{h}(n) + \frac{\partial \underline{\underline{B}}(n)}{\partial b_\ell} \underline{h}(n-1) - \frac{\partial \underline{f}(n)}{\partial b_\ell} \right\}, \tag{75}$$

where $S(\underline{b}) = S_h(\underline{b}) + S_f(\underline{b}) + S_p(\underline{b})$;

$S_h(\underline{b})$ are terms of the objective function related to observed hydraulic heads or changes in hydraulic head;

$S_f(\underline{b})$ are terms of the objective function related to head-dependent boundary gain and loss observations; and

$S_p(\underline{b})$ are terms of the objective function related to prior estimates of the parameters.

All other terms were defined after equations (3) or (69). A clear derivation of equations (73), (74) and (75) is presented by Townley and Wilson (1985, p. 1854-1855), and is not repeated in this report.

A simplistic understanding of the adjoint states can be developed by considering equations (73) and (74) when $h_u(\underline{b})$ is substituted for $S(\underline{b})$, where $h_u(\underline{b})$ is a single hydraulic head, and u indicates its time and location. In this circumstance, $\partial h_u(\underline{b})/\partial b_\ell$, ℓ=1,NP, would be calculated. In a steady-state system, the adjoint states are the solution of:

$$\underline{\underline{A}}(\underline{0})\ \underline{\phi} = \frac{\partial h_u(\underline{b})}{\partial \underline{h}}. \tag{76}$$

The right-hand side is equal to a vector of zeros except for a 1.0 in the location of u. The adjoint states $\underline{\phi}$ then could be interpreted as the change in hydraulic head at each grid node caused by a unit recharge at location u. An equally valid interpretation is that each value in $\underline{\phi}$ is the change in $h_u(\underline{b})$ that would occur because of a unit recharge at the grid node associated with the value of $\underline{\phi}$. The latter interpretation was presented by Wilson and Metcalfe (1985), and is consistent with the view that the adjoint states relate changes in ground-water flow to the sensitivities.

For a simplistic understanding of the transient adjoint states, consider first $\underline{\phi}(N-1)$, where N is the time step at which $h_u(\underline{b})$ occurs. Because $\underline{\phi}(N)=0$, $\underline{\phi}(N-1)$ is the solution of:

$$\underline{\underline{A}}(N-1)\ \underline{\phi}(N-1) = \frac{\partial h_u(\underline{b})}{\partial \underline{h}(N)}. \tag{77}$$

The right-hand side of (77) is a vector of zeros, except for a 1 in the location identified by u. If all time steps are the same length and the direction of the time sequence is disregarded, it is simple to imagine that $\underline{\phi}(N-2)$, $\underline{\phi}(N-3)$, ... $\underline{\phi}(0)$ represent the dissipation of this unit recharge over time. This representation is not correct when the time steps are of different lengths.

In equation (75), $\underline{\phi}(-1)$ is used to relate the initial conditions of the problem to the calculated derivative. In solving for $\underline{\phi}(-1)$ (eq. 73), the storage term is deleted from the left-hand side of the equation. Thus, $\underline{\phi}(-1)$ is larger than $\underline{\phi}(n)$, $n > -1$, which indicates the enduring importance of the initial conditions. As N increases, $\lambda(-1)$ becomes smaller, and the initial conditions become less important.

When $h_u(\underline{b})$ is substituted for $S(\underline{b})$, the sum of the adjoint states for transient problems equals the steady-state adjoint state ($\underline{\phi}_{ss}$). That is, for any N:

$$\sum_{i=-1}^{N} \underline{\phi}(i) = \underline{\phi}_{ss}. \qquad (78)$$

This equality means that the same total weighting is used to relate changes in ground-water flow to the sensitivities (see discussion after eq. 76), but that the weighting is spread out over the time steps involved. This property does not apply when calculating the gradient.

To calculate the gradient for problems with water-table layers, which are nonlinear, $\underline{\underline{A}}(n)$ and $\underline{\underline{A}}(0)$ of equation (73) would be replaced by $\frac{\partial \underline{\underline{A}}(n)}{\partial \underline{h}(n)} \underline{h}(n) + \underline{\underline{A}}(n)$ and $\frac{\partial \underline{\underline{A}}(0)}{\partial \underline{h}(0)} \underline{h}(0) + \underline{\underline{A}}(0)$, respectively. Although iterative updating, as discussed above, perhaps could be used to solve the resulting equations; this was not done in the Parameter-Estimation Package. Thus, for problems with water-table layers, parameter estimation needs to be accomplished by using modified Gauss-Newton optimization.

REFERENCES

Akaike, Hirotugu, 1974, A new look at statistical model identification: Institute of Electrical and Electronics Engineers Transactions on Automatic Control, v. AC-19, no. 6, p. 716-723.

_____ 1978, Time series analysis and control through parametric models, in Findley, D.F., ed., Applied time series analysis: New York, Academic Press, p. 1-25.

Aziz, Khalid, and Settari, Antonin, 1979, Petroleum reservoir simulation: Elsevier, 476 p.

Bard, Jonathon, 1974, Nonlinear parameter estimation: New York, Academic Press, 341 p.

Beck, J.V., and Arnold, K.J., 1977, Parameter estimation in engineering and science: New York, John Wiley and Sons, 501 p.

Benjamin, J.R., and Cornell, C.A., 1970, Probability, statistics and decision for civil engineers: New York, McGraw-Hill, 684 p.

Brockwell, P.J., and Davis, R.A., 1987, Time series--Theory and methods: New York, Springer-Verlag, 519 p.

Carrera, Jesus, 1984, Estimation of parameters under transient and steady-state equations: Tucson, Arizona, University of Arizona, Department of Hydrology and Water Resources, unpublished Ph.D. dissertation, 258 p.

Carrera, Jesus, and Neuman, S.P., 1986a, Estimation of aquifer parameters under transient and steady-state conditions--1. Maximum likelihood method incorporating prior information: Water Resources Research, v. 22, no. 2, p. 199-210.

_____ 1986b, Estimation of aquifer parameters under transient and steady-state conditions--2. Uniqueness, stability, and solution algorithms: Water Resources Research, v. 22, no. 2, p. 211-227.

_____ 1986c, Estimation of aquifer parameters under transient and steady-state conditions--3. Applications to synthetic and field data: Water Resources Research, v. 22, no. 2, p. 228-242.

Carter, R.W. and Anderson, I.E., 1963, Accuracy of current meter measurements: American Society of Civil Engineers Journal, v. 89, no. HV4, p. 105-115.

Clifton, P.M. and Neuman, S.P., 1982, Effects of kriging and inverse modeling on conditional simulation of the Avra Valley aquifer in southern Arizona: Water Resources Research, v. 18, no. 4, p. 1215-1234.

Cooley, R.L., 1982, Incorporation of prior information on parameters into nonlinear regression groundwater flow models--1. Theory: Water Resources Research, v. 18, no. 4, p. 965-976.

_____ 1983a, Incorporation of prior information on parameters into nonlinear regression groundwater flow models--2. Applications: Water Resources Research, v. 19, no. 3, p. 662-676.

_____ 1983b, Some new procedures for numerical solution of variably saturated flow problems: Water Resources Research, v. 19, no. 5, p. 1271-1285.

_____ 1985, A comparison of several methods of solving nonlinear regression groundwater flow problems: Water Resources Research, v. 21, no. 10, p. 1525-1538.

Cooley, R.L. and Hill, M.C., in press, A comparison of three Newton-like nonlinear least-squares method for estimating parameters of ground-water flow models in Russell, T.F., Ewing, R.E., Brebbia, C.A., Gray W.G., and Pinder, G.F., eds., Computational methods in subsurface hydrology: International Conference on Computational Methods in Water Resources, 9th, Denver, 1992, Proceedings.

Cooley, R.L., Konikow, L.F., and Naff, R.L., 1986, Nonlinear-regression groundwater flow modeling of a deep regional aquifer system: Water Resources Research, v. 22, no. 13, p. 1759-1778.

Cooley, R.L., and Naff, R.L., 1990, Regression modeling of ground-water flow: U.S. Geological Survey Techniques of Water-Resources Investigations, Book. 3, Chapter B4, 232 p.

Cryer, J.D., 1986, Time series analysis: Boston, Duxbury Press, 286 p.

Davis, S.N., 1969, Porosity and permeability of natural materials, in DeWeist, R.J.M., ed., Flow through porous media: New York, Academic Press, p. 54-89.

de Marsily, Ghislain, Lavedon, G., Boucher, M., and Fasinino, G., 1984, Interpretation of interference tests in a well field using geostatistical techniques to fit the permeability distribution in a reservoir model, in Verly, Georges, David, M., Journel, A.G., and Marechal, Alain, eds., Geostatistics for natural resources characterization: New York, Reidel, D., NATO ASI Series C, v. 122, p. 831-849.

Dennis, J.E., Gay, D.M. and Welsch, R.E., 1981, An adaptive nonlinear least-squares algorithm: ACM Transactions on Mathematical Software, v. 7, no. 3, p. 348-368.

Dennis, J.E., and Schnabel, R.B., 1983, Numerical methods for unconstrained optimization and nonlinear equations: Englewood Cliffs, New Jersey, Prentice-Hall, 378 p.

Donaldson, J.R. and Schnabel, R.B., 1987, Computational experience with confidence regions and confidence intervals for nonlinear least squares: Technometrics, v. 29, no. 1, p. 67-82.

Draper, N.R., and Smith, Harry, 1981, Applied regression analysis (2nd ed.): New York, John Wiley and Sons, 709 p.

Forsythe, G.E. and Strauss, E.G., 1955, On best conditioned matrices: American Mathematical Society Proceedings, v. 10, no. 3, p. 340-345.

Freeze, R.A., 1975, A stochastic-conceptual analysis of one-dimensional groundwater flow in nonuniform homogeneous media: Water Resources Research, v. 11, no. 5, p. 725-741.

Fuller, W.A., 1987, Measurement error models: New York, John Wiley and Sons, 440 p.

Gere, J.M., and Weaver, W.W., 1965, Matrix algebra for engineers: New York, D. van Nordstrom, 168 p.

Gill, P.E., Murray, Walter, and Wright, M.H., 1981, Practical optimization: New York, Academic Press, 401 p.

Gomez-Hernandez, J.J., and Gorelick, S.M., 1989, Effective groundwater model parameter values--influence of spatial variability of hydraulic conductivity, leakance, and recharge: Water Resources Research, v. 25, no. 3, p 405-419.

Harbaugh, A.W., 1990, A simple contouring program for gridded data: U.S. Geological Survey Open-File Report 90-144, 37 p.

Hildebrand, F.B., 1976, Advanced calculus for applications: Englewood Cliffs, New Jersey, Prentice-Hall, 733 p.

Hill, M.C., 1989, Analysis of accuracy of approximate, simultaneous, nonlinear confidence intervals on hydraulic heads in analytical and numerical test cases: Water Resources Research, v. 25, no. 2, p. 177-190.

_____ 1990a, Relative efficiency of four parameter-estimation methods in steady-state and transient ground-water flow models in Gambolati, Giuseppe, Rinaldo, Andrea, Brebbia, C.A., Gray, W.G., and Pinder, G.F., eds., Computational methods in subsurface hydrology: International Conference on Computational Methods in Water Resources, 8th, Venice, 1990, Proceedings, p. 103-108.

_____ 1990b, Preconditioned Conjugate-Gradient 2 (PCG2)--A computer program for solving ground-water flow equations: U.S. Geological Survey Water-Resources Investigations Report 90-4048, 43 p.

Hill, M.C., Lennon, G.P., Hebson, C.S., Brown, G.A., and Rheaume, S.J., in press, Geohydrology of, and simulation of ground-water flow in, the valley-fill deposits in the Ramapo River valley, New Jersey: U.S. Geological Survey Water-Resources Investigations Report 90-4151.

Himmelblau, D.M., 1972, Applied nonlinear programming: New York, McGraw-Hill, 477 p.

IMSL, 1987, Math/Library User's Manual, FORTRAN Subroutines for Mathematical Applications--Version 1.0, April 1987, Houston, 1,232 p.

Keidser, Allan, and Rosbjerg, Dan, 1991, A comparison of four inverse approaches to groundwater flow and transport parameter identification: Water Resources Research, v. 27, no. 9, p. 2219-2232.

Keidser, Allan, and Rosbjerg, Dan, Jensen, K.H., Bitsch, K., 1990, A joint kriging and zonation approach to inverse groundwater modelling, in Kovar, Karel, ed., ModelCARE 90--Calibration and reliability in groundwater modelling: International Association Hydrologic Sciences Publication 195, p. 171-183.

Kuiper, L.K., 1987, Computer program for solving ground-water flow equations by the preconditioned conjugate gradient method: U.S. Geological Survey Water Resources Investigations Report 87-4091, 34 p.

Leake, S.T., 1990, Applications of user-supplied transformations in computer-graphics programs, in Selected papers in the applied computer sciences, 1990: U.S. Geological Survey Bulletin 1908, ch. A, 5 p.

Lu, A.H., Schittroth, F., and Yeh, W.W-G., 1988, Sequential estimation of aquifer parameters: Water Resources Research, v. 24, no. 5, p. 670-682.

Marquardt, D.W., 1963, An algorithm for least-squares estimation of nonlinear parameters: Journal of the Society of Industrial and Applied Mathematics, v. 11, no. 2, p. 431-441.

McDonald, M.G., and Harbaugh, A.W., 1988, A modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Techniques of Water-Resources Investigations, Book 6, Chapter A1, 548 p.

McLaughlin, Dennis, 1975, Investigation of alternative procedures for estimating ground-water basin parameters: U.S. Deptartment of the Interior Office of Water Research and Technology, 171 p.

Meyer, P.D., Valocchi, A.J., Ashby, S.F., and Saylor, P.E., 1989, A numerical investigation of the conjugate gradient method as applied to three-dimensional groundwater flow problems in randomly heterogeneous porous media: Water Resources Research, v. 25, no. 6, p. 1440-1446.

Neuman, S.P., 1980, A statistical approach to the inverse problem of aquifer hydrology--3. Improved solution method and added perspective: Water Resources Research, v. 16, no. 2, p. 331-346.

_____ 1982, Statistical characterization of aquifer heterogeneities--An overview, in Narasimhan, T.N., ed., Recent trends in hydrology: Geological Society of America Special Paper 189, p. 81-102.

Neuman, S.P. and Jacobson, E.A., 1984, Analysis of nonintrinsic spatial variability by residual kriging with application to regional groundwater levels: Mathematical Geology, v. 16, no. 5, p. 499-521.

Neuman, S.P. and Yakowitz, Sidney, 1979, A statistical approach to the inverse problem of aquifer hydrology, 1, Theory: Water Resources Research, v. 15, no. 4, p. 845-860.

Nielsen, D.R., Biggar, J.W., and Erb, K.T., 1973, Spatial variability of field-measured soil-water properties: Hilgardia, v. 42, no. 7, p. 215-259.

Odenwald, B., and Herrling, Bruno, 1990, A quick and stable algorithm for groundwater flow parameter estimation under uncertainty, in Kovar, Karel, ed., ModelCARE90--Calibration and reliability in groundwater modeling: International Association Hydrologic Sciences Publication 195, p. 75-86.

Ott, Lyman, 1988, An introduction to statistical methods and data analysis: Boston, PWS-Kent Publishing, 945 p.

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., 1989, Numerical recipes: Cambridge, Great Britain, Press Syndicate of the University of Cambridge, 702 p.

Prudic, D.E., 1989, Documentation of a computer program to simulate stream-aquifer relations using a modular, finite-difference, ground-water flow model: U.S. Geological Survey Open-File Report 88-729, 114 p.

Sadeghipour, Jamshid, and Yeh, W.W-G., 1984, Parameter identification of groundwater aquifer models--A generalized least squares approach: Water Resources Research, v. 20, no. 7, p. 971-979.

Scandrett, Clyde, 1989, Comparison of several iterative techniques in the solution of symmetric banded equations on a two-pipe Cyber 205: Applied Mathematics and Computation, v. 34, no. 2, p. 95-112.

Schmittroth, F., 1979, A method for data evaluation with lognormal distributions: Nuclear Science and Engineering, no. 72, p. 19-34.

Seber, G.A.F. and Wild, C.J., 1989, Nonlinear regression: New York, John Wiley, 768 p.

Segerlind, L.J., 1976, Applied Finite Element Analysis: New York, John Wiley and Sons, 422 p.

Shah, P.C., Gavalas, G.R., and Seinfeld, J.H., 1978, Error analysis in history matching--The optimum level of parameterization: Journal of the Society of Petroleum Engineers, v. 18, no. 3, p. 219-228.

Shanno, D.F., 1978, On the convergence of a new conjugate gradient algorithm: SIAM Journal of Numerical Analysis, v. 15, no.6, p. 1247-1257.

Shapiro, S.S., and Francia, R.S., 1972, An approximate analysis of variance test for normality: Journal of the American Statistical Association, v. 67, p. 215-216.

Stewart, G.W., 1972, Introduction to matrix computations: New York, Academic Press, 423 p.

Sudicky, E.A., 1986, A natural gradient experiment on solute transport in a sand aquifer--Spatial variability of hydraulic conductivity and its role in the dispersion process: Water Resources Research, v. 13, p. 2069-2083.

Townley, L.R., and J.L. Wilson, 1985, Computationally efficient algorithms for parameter estimation and uncertainty propagation in numerical models of groundwater flow: Water Resources Research, v. 21, no. 12, p. 1851-1860.

Trescott, P.C., and Larson, S.P., 1977, Comparison of iterative methods of solving two-dimensional groundwater flow equations: Water Resources Research, v. 13, no. 1, p. 125-136.

U.S. Geological Survey, 1990, Accuracy specifications for topographic mapping, in Technical instructions of the National Mapping Division: Reston, Virginia, Chapter 1B4, p. 1-13.

Vecchia, A.V., and Cooley, R.L., 1987, Simultaneous confidence and prediction intervals for nonlinear regression models with application to a groundwater flow model: Water Resources Research, v. 23, no. 7, p. 1237-1250.

Wang, H.F, and Anderson, M.P., 1982, Introduction to groundwater modeling: San Francisco, W. H. Freeman, 237 p.
Watson, T.A., Gatens III, J.M., Lee, J.W., and Rahim, Zillus, 1990a, An analytical model for history matching naturally fractured reservoir production data: Society of Petroleum Engineers, Reservoir Engineering, August 1990, p. 384-388.
Watson, T.A., Lane, H.S., and Gatens III, J.M., 1990b, History matching with cumulative production data: Society of Petroleum Engineers, Reservoir Engineering, January 1990, p. 96-100.
Wilson, J.L., and Metcalfe, 1985, Illustration and verification of adjoint sensitivity theory for steady state groundwater flow: Water Resources Research, v. 21, no. 11, p. 1602-1610.
Yager, R.M., 1991, Estimation of hydraulic conductivity of a riverbed and aquifer system on the Susquehanna River in Broome County, New York: U.S. Geological Survey Open-File Report 91-457, 54 p.
Yeh, W.W-G., 1986, Review of parameter identification procedures in groundwater hydrology--The inverse problem: Water Resources Research, v. 22, no. 2, p. 95-108.
Yeh, W.W-G., and Yoon, Y.S., 1981, Aquifer parameter identification with optimum dimension in parameterization: Water Resources Research, v. 17, no. 3, p. 664-672.

APPENDIX A: MODFLOWP INPUT AND OUTPUT

Input Instructions

These input instructions are intended to be used in conjunction with the documentation for the modular, three-dimensional, ground-water flow model (MODFLOW) presented by McDonald and Harbaugh (1988) and the Streamflow-Routing Package presented by Prudic (1989). As needed, their variable names are used here with their lower- and upper-case lettering conventions. If the Parameter-Estimation Package is not activated, MODFLOWP operates identically to MODFLOW, with a minor exception discussed in the section "Hydraulic Heads," Appendix C.

In the following discussion, the input file for the Parameter-Estimation Package will be referred to as the INPUT FILE. IUNIT(15), which is read from the Basic Package input file, is used to store the input unit number for the INPUT FILE. If desired, another element of IUNIT can be used by changing all references to IUNIT(15); see text in the first section of Appendix B.

The Parameter-Estimation Package can be used to: (1) calculate hydraulic heads and flows like MODFLOW, but using the parameter values specified in the INPUT FILE; (2) calculate hydraulic heads, and sensitivity-equation sensitivities of simulated hydraulic heads by using equations (69), (70), and (71), or the adjoint states and the gradient of the objective function by using equations (73), (74), and (75); or (3) perform parameter estimation by nonlinear regression. Nonlinear regression can be accomplished by modified Gauss-Newton optimization by using sensitivity-equation sensitivities, or by conjugate-direction optimization by using the gradient of the objective function calculated by using adjoint states. When parameter estimation converges, final calculated hydraulic heads and flows can be printed and saved as they are for MODFLOW by specifying KPRINT=1 in DATA SET 13 of the INPUT FILE.

Calculating Hydraulic Heads and Flows

Hydraulic heads and flows first need to be calculated without the Parameter-Estimation Package [IUNIT(15)=0]. Without the Parameter-Estimation Package, MODFLOWP performs almost exactly like MODFLOW and requires the same input files. Constructing the MODFLOW input files and simulating hydraulic heads and flows, and then constructing an INPUT FILE that is supposed to produce the same hydraulic heads and flows produces an effective check on the parts of the INPUT FILE used to calculate model inputs from the parameter values.

Calculating hydraulic heads and flows by using the Parameter-Estimation Package [IUNIT(15)≠0] involves modifying the MODFLOW input files as described in the following section, constructing the INPUT FILE as described below, and setting IPAR of LINE 6 of the INPUT FILE equal to -1. Calculating hydraulic heads and flows by using the Parameter-Estimation Package differs from calculating hydraulic heads and flows with MODFLOW in three ways. First, problems with both steady-state and transient conditions are simulated with one model run and one set of input files. This change was required for parameter estimation and alleviates problems related to keeping the data sets for the two types of problems consistent. Second, the parameter values defined in DATA SET 8 of the INPUT FILE are used in the simulation. This feature makes the parameter values easy for the user to change, and this feature can be used to help determine the initial parameter values of the parameter-estimation procedure or to facilitate trial-and-error calibration for models that are not calibrated by nonlinear regression. Calibration by nonlinear regression might not be possible, for example, if most of the available computer storage or execution capacity is used just to calculate hydraulic heads. Third, the simulation continues only for as many time steps as are needed to include all the time steps at which there are observations (see DATA SETS 6, 6C, and 7A).

The hydraulic heads and flows calculated with IUNIT(15)=0 and with IUNIT(15)≠0 need to be nearly identical. Discrepancies generally indicate errors in DATA SETS 2, 2A, 2B, 3, and 4 of the INPUT FILE, which are used to define the parameterization and are described below.

Calculating Sensitivity-Equation Sensitivities and the Gradient of the Objective Function and Performing Parameter Estimation

Use of the Parameter-Estimation Package involves modifying the input files from other packages and constructing the INPUT FILE as described below.

Changes to input files from other packages

When used with the Parameter-Estimation Package, the input files for some of the packages described by McDonald and Harbaugh (1988) and the package described by Prudic (1989) need to be changed slightly. For steady-state problems, these changes are made to the files used to calculate steady-state hydraulic heads with IUNIT(15)=0; for problems that include steady-state and transient or just transient conditions, these changes are made to the files used to calculate transient hydraulic heads with IUNIT(15)=0. Examples are included for the test cases at the end of Appendix A. The needed changes are as follows:

1. The Basic Package input file needs to be changed in the following ways:

a. NPER needs to be equal to or greater than the time step of the latest observation (see DATA SETS 6, 6C, and 7A in the INPUT FILE). (In MODFLOWP, time steps are numbered from the beginning of the simulation, so remember to include all the time steps from preceding stress periods when calculating the time step of the latest observation). If TOFF (DATA SETS 6, 6C, and 7A) of the latest observation is greater than zero, NPER needs to be equal to or greater than the latest time step plus 1. NPER=0 can be used for steady-state problems.

b. IUNIT(15) needs to equal the FORTRAN unit number from which the INPUT FILE is read.

c. Starting hydraulic heads (Shead) from the steady-state Basic Package input file need to be used in the Basic Patkage input file.

2. For transient simulations, input files for the River, Recharge, Well, Drain, Evapotranspiration, General-Head Boundary, and Streamflow-Routing Packages need to include the data "FOR EACH STRESS PERIOD" (McDonald and Harbaugh, 1988; Prudic, 1989) from the steady-state simulation, followed by the data "FOR EACH STRESS PERIOD" from the transient simulation. The Ouput Control input file needs to include the data "FOR EACH TIME STEP" from the steady-state simulation, followed by the data "FOR EACH TIME STEP" from the transient simulation. For steady-state simulations, these input files need not be changed.

3. For transient simulations, if the IBOUND array of the Basic Package input file is different for steady-state and transient conditions, a new input file is needed. This input file contains the steady-state IBOUND array followed by the transient IBOUND array and is read from FORTRAN unit IUBD (LINE 7 of the INPUT FILE). These IBOUND arrays replace the IBOUND array read

from the Basic Package input file. Note that any cells that are inactive in the steady state IBOUND array must be inactive in the transient array. If any layers are unconfined, only the steady-state IBOUND array is read from IUBD.

4. If the top simulated layer is a water-table layer (Block-Centered Flow Package: LAYCON=1), and any cells go dry for any of the sets of parameter values that occur in the parameter-estimation iterations, the IBOUND array needs to be read from a new input file, as described in item 3 above. For a steady-state simulation, this input file contains the IBOUND array for steady-state conditions. For transient problems, this input file needs to contain two IBOUND arrays; repeat the steady-state IBOUND array if it is used for steady-state and transient conditions.

5. Many packages of MODFLOW permit arrays to be read from separate files using subroutines U2DREL and U2DINT (McDonald and Harbaugh, 1988, ch. 14). The Parameter-Estimation Package also permits arrays to be read from separate files using U2DREL and U2DINT, and care needs to be taken that one file is not read by two packages, which results in an end-of-file error. A common example occurs when defining a transmissivity parameter and U2DREL is called to read cell values of aquifer thickness from a separate file. The user might try to read the same separate file from the Parameter-Estimation Package that has already been read from the Block-Centered Flow Package. The problem can be resolved by creating a new copy of the file, or by realizing that the parameter values specified by the INPUT FILE (DATA SET 8) are used to replace the values read from the Block-Centered Flow Package. Thus, the Block-Centered Flow Package input file can be changed to assign an arbitrary constant value of transmissivity [positive, nonzero values are needed so that submodule SBCF1N (McDonald and Harbaugh, 1988, p. 5-68) will not change IBOUND values to zero], and the separate file is read only from the Parameter-Estimation Package.

6. Output Control input-file variables IHEDFM, IHEDUN, IHDDFL, Hdpr, and Hdsv are used to control the printing and saving of hydraulic heads, adjoint states, and sensitivity-equation sensitivities for all cells when using the Parameter-Estimation Package. Read the instructions for IPAR of LINE 6 of the INPUT FILE and change the listed variables as needed to produce the desired output.

7. Superposition simulations, in which changes in hydraulic head are calculated, can conform to the input-file specifications described above by including steady-state conditions in which all hydraulic heads equal zero.

8. To estimate parameters for NSIM simulations of one flow system (for example, if there are NSIM aquifer tests), begin by estimating the parameters for each individual simulation using the directions in this appendix. To combine the simulations, the Basic Package input file from the first simulation needs to be modified in the following ways:

a. NPER needs to equal the sum of the NPER values from each of the simulations, plus NSIM-1. The NSIM-1 time steps account for the steady-state conditions for all but the first simulation.

b. Add the data lines from the other NSIM-1 Basic Package input files that specify PERLEN, NSTP, and TSMULT. Preceding the lines for all but the first simulation, add a line with PERLEN=1.E23, NSTP=1, TSMULT=1.0. The added lines specify the large time step needed to simulate steady-state conditions.

The following changes involve the INPUT FILE, which is discussed in the following sections. These changes are listed here to keep all changes together, but the following sections need to be read first.

c. Simulations with different known hydraulic heads at constant hydraulic-head boundaries can be represented using the methods discussed in the section "IUNHEA."

d. If the NSIM simulations are superposition simulations, the hydraulic heads and sensitivities at the beginning of all but the first simulation can be set to zero using NZER of LINE 8 and IZER of DATA SET 1B of the INPUT FILE. The time-step numbers of NZER equal the time step associated with what was steady state when the simulations were considered separately. Thus, the time step of the steady state preceding the second simulation equals NPER of the first simulation plus 1; the time step of the steady state preceding the third simulation equals NPER of the first simulation, plus 1, plus NPER of the second simulation, plus 1. This process is continued until all simulations are included.

e. The time steps of the observations in the INPUT FILE (see discussion of DATA SETS 6, 6C, and 7A of the section "Data entry format and instructions for the INPUT FILE") for all but the first simulation need to be changed. For the second simulation, add NPER of the first simulation to all time-step numbers. For the third simulation, add NPER of the first simulation plus 1+NPER of the second simulation; the 1 accounts for the steady-state conditions of the second simulation. For the fourth simulation, add NPER of the first simulation, plus 1+NPER of the second simulation, plus 1+NPER of the third simulation. Continue until all simulations are included. Once these changes are made, all observations need to be included in one INPUT FILE, and NH, MOBS, MAXM, NQ, NQC, and NQT of DATA LINE 5 need to be changed to accommodate the total number of observations.

Discussion of the INPUT FILE

The INPUT FILE is described below. Some of the more complicated variables are discussed first, and the data entry and format instructions follow.

ISN

ISN (LINE 6) is used to select between the different optimization and numerical methods available in the Parameter-Estimation Package, and computer-storage allocation depends on ISN. Usually ISN=-1; the other values discussed in this section need to be considered only under special circumstances.

If ISN<0, sensitivity-equation sensitivities are calculated (see section "Sensitivity-Equation Method of Calculating Sensitivities), and modified Gauss-Newton optimization is performed. Two options are available for calculating the sensitivity-equation sensitivities:

1. If ISN=-1, all sensitivities are calculated at each time step before progressing to the next time step. This method permits matrix $\underline{\underline{A}}$ on the left-hand side of equation (69) or (71) to be calculated only once for each time step, so if a direct solver, such as the D4 solver (S.A. Leake, U.S. Geological Survey, written commun., 1989), is used, $\underline{\underline{A}}$ only needs to be decomposed once for each time step. Because decomposition of direct solvers might use more than 75 percent of the computer time required by a direct solver, using one decomposition can be advantageous. When ISN=-1, NP files

numbered starting with IUHEAD are used to store nodal arrays of sensitivities (NP and IUHEAD are read from LINES 3 and 7 of the INPUT FILE, respectively). If the number of parameters is less than the number of time steps, the number of files will be smaller with ISN=-1 than with ISN=-2.

2. If ISN=-2, sensitivities for each parameter are calculated for all time steps before progressing to the next parameter. Unless the problem is steady state or $\underline{\underline{A}}$ is the same for all time steps (constant time-step length), $\underline{\underline{A}}$ needs to be decomposed at each time step when using a direct solver, and direct solvers generally are not as efficient with ISN=-2 as with ISN=-1. Thus, a conjugate-gradient solver probably needs to be used for most problems with ISN=-2. When ISN=-2, files numbered starting with IUHEAD of LINE 3 are used to store nodal arrays of hydraulic head for each time step. If the number of time steps is less than the number of parameters, the number of files will be smaller with ISN=-2 than with ISN=-1.

If ISN>0, the gradient of the objective function is calculated by the adjoint-state method, and parameter estimates are calculated by a conjugate-direction method. NFIT of DATA SET 13 of the INPUT FILE is used to designate the number of Fletcher-Reeves iterations that precede quasi-Newton iterations, and NFIT needs to be coordinated with ISN. Options for ISN are as follows:

1. If ISN=1, enough computer storage is allocated to permit quasi-Newton iterations, which need slightly more computer storage than Fletcher-Reeves iterations. Fletcher-Reeves iterations can be performed with ISN=1, so that NFIT can be greater than zero.

2. If ISN=2, enough computer storage is allocated to permit Fletcher-Reeves iterations. The allocated computer storage is insufficient for quasi-Newton iterations, so NFIT needs to be greater than or equal to ITMXP of DATA SET 13 to ensure that no quasi-Newton iterations are executed.

3. If ISN=4, enough computer storage is allocated to permit quasi-Newton and, therefore, Fletcher-Reeves iterations, and to calculate statistics that need the sensitivity matrix, $\underline{\underline{X}}_r$. The statistics are calculated if parameter estimation converges and LASTX of DATA SET 13 of the INPUT FILE is larger than zero.

IPAR

IPAR (LINE 6) is used to identify whether to (1) just calculate hydraulic heads and flows, (2) calculate hydraulic heads and flows and sensitivity-equation sensitivities (ISN<0) or adjoint states and the gradient of the objective function (ISN>0), or (3) perform parameter estimation.

If IPAR<0, hydraulic heads are calculated. Hydraulic-head arrays at each time step are printed or saved or both as specified by the Output Control Package input file; calculated and observed hydraulic heads or changes in hydraulic heads at the NH times and locations (see DATA SETS 6 and 6C) and flows at the NQT times and head-dependent boundaries (see DATA SET 7A) are printed.

If IPAR=0, hydraulic heads and sensitivity-equation sensitivities (ISN<0) or adjoint states (ISN>0) are calculated. If ISN<0, hydraulic head and sensitivity-equation sensitivity arrays are printed or saved or both for time steps when at least one of the NH hydraulic heads or NQT flows occurs and Hdpr and Hdsv of the Output Control input file are set to allow the output. If ISN>0, hydraulic heads and adjoint states are printed or saved or both for all time steps indicated by Hdpr and Hdsv. Adjoint states are calculated backwards in time (eq. 73), but the Output Control Package input file is read forward in time, so the Hdpr and Hdsv values for time step KSTP are used to print the adjoint states $\underline{\phi}$(N-KSTP), where N is the latest time step at which a component of $S(\underline{b})$ occurs. (See the sections "Adjoint-State Method of Calculating the Gradient of the Objective Function" and "Calculated Values and Volumetric Budgets for the Entire Model" for more information.) In addition, the following values are printed: calculated and observed hydraulic heads at the NH times and locations and flows at the NQT times and boundaries; for ISN<0, NH+NQT sensitivities; for ISN>0, the gradient of the objective function.

If IPAR=1, parameter estimation by nonlinear regression is performed. The following values are printed: calculated and observed hydraulic heads or changes in hydraulic head at the NH times and locations; NQT calculated and observed gains and losses at head-dependent boundaries; NH+NQT sensitivities (ISN<0) or the gradient of the objective function (ISN>0); and data from the regression procedure. IPRINT of DATA SET 13 of the INPUT FILE controls whether or not all of these items are printed at every iteration of the parameter-estimation procedure. If parameter estimation converges, the printing or saving or both of final hydraulic heads is controlled by KPRINT of DATA SET 13 and Hdpr and Hdsv of the Output Control Package input file.

IUNHEA

IUNHEA (LINE 7) is used to indicate that there are temporal changes in known hydraulic heads at constant hydraulic-head boundaries. Known hydraulic heads at constant hydraulic-head boundaries are specified by the user and are not affected by the parameters defined by using the CH identifier (see DATA SETS 2 and 2A of the INPUT FILE). Known hydraulic heads can be changed at time steps specified by the user by defining IUNHEA≠0. Module SEN1RH (see Appendix C) is called from MAIN when IUNHEA≠0, and the present version functions as described below. Although this version is general, it requires an input file that might be extremely large for some problems. Users are encouraged to modify SEN1RH to coordinate with their problems and to decrease the size of or eliminate the input file.

The present version of SEN1RH reads from unit number IUNHEA. The first data card of the input file is a list of as many as 16 time steps at which the known hydraulic heads at constant hydraulic-head boundaries change. The format is 16F5.0 (note that integers can be read using this format; they need to be right-justified). For every nonzero time-step number (all nonzero numbers need to be to the left of blanks or zeros), the input file contains an array of hydraulic heads for each of the NLAY model layers. These arrays are read using U2DREN (Appendix C), a utility module that operates like U2DREL (McDonald and Harbaugh, 1988, p. 14-26) but prints nothing; use the instructions for U2DREL to construct this part of the input file. At each of the time steps listed on the first data card, the hydraulic heads at all constant hydraulic-head boundaries are replaced using values from the associated arrays of hydraulic heads. The first set of NLAY arrays read is associated with the first time step listed, the second set of NLAY arrays read is associated with the second time step, and so forth.

Parameter Definition: DATA SETS 2, 2A, 2B, 3, and 4

DATA SETS 2, 2A, 2B, 3, and 4 define how parameters (with values initially specified in DATA SET 8 and subsequently calculated by nonlinear regression) are used to calculate the model inputs described in the MODFLOW input instructions (McDonald and Harbaugh, 1988; Prudic, 1989), and listed in table 1. A DATA SET 2 is repeated for each parameter except for parameters used to define constant hydraulic-head boundaries, in which instance many paramters can be defined under one DATA SET 2. The model input(s) involved are identified by PID, the first variable of DATA SET 2. PID is a character variable; available options and their associated model inputs are listed in table A1. Use of array NLL, which is part of DATA SET 2, and DATA SETS 2A and 2B is mentioned in table A1, and is discussed in detail in the following paragraphs.

For PID=ANI and ANIV, array NLL is used to specify layer numbers. The parameter needs to equal the quantity specified as the model input in table A1, and only DATA SET 2 is needed.

DATA SET 2 is followed by DATA SET 2A if the MODFLOW input instructions specify that the model input indicated by PID be defined by a list of cells. This occurs for pumpage (called the volumetric recharge rate by McDonald and Harbaugh, 1988) of the Well Package (PID=Q), and the conductances of the River, Drain, Streamflow-Routing, and General-Head Boundary Packages (PID=KRB, KDR, KST, and GHB). DATA SET 2A is a list of cells (specified by layer, row, and column or, for the Streamflow-Routing Package, by reach and segment) where the parameter applies, and can include cell-by-cell multiplicative factors that allow the user to define the parameter to be the most physically significant quantity. For example, noting that the conductance for a cell equals $(K_i/D_i)A_i$ (eq. 4), the parameter can be defined as K_i if K_i can be considered to be constant for all listed cells. Then the multiplicative factors would equal (A_i/D_i), which might be different for each cell. As a default, all multiplicative factors can be set to 1.0 using NLL(1) of DATA SET 2, which implies that the parameter equals the required model input for all listed cells.

Table A1:--*Available options for the parameter identifier (PID), the INPUT FILE data sets used to calculate the contribution of this parameter to the MODFLOW model input (McDonald and Harbaugh, 1988; Prudic, 1989), and the MODFLOW model input*

PID	Multipliers and spatial extent defined by	Time defined by	MODFLOW Model input
Q	NLL(1); DATA SET 2A	NLL(I)	Volumetric recharge rate of the Well Package.
KRB[1]	NLL(1); DATA SET 2A	NLL(I)	Conductance of the River Package.
KDR[1]	NLL(1); DATA SET 2A	NLL(I)	Conductance of the Drain Package.
KST[1]	NLL(1); DATA SET 2A	NLL(I)	Conductance of the Streamflow-Routing Package.
GHB[1]	NLL(1); DATA SET 2A	NLL(I)	Conductance of the General-Head Boundary Package.
CH	NLL(I); DATA SET 2A	NLL(2)	Constant hydraulic head linearly interpolated between specified nodes.
T[1]	NLL(I); DATA SET 2B		Horizontal hydraulic conductivity or transmissivity of the Block-Centered Flow Package[2].
ANI[1]	NLL(I)		Horizontal anisotropy of the Block-Centered Flow Package.
KV[1]	NLL(I); DATA SET 2B		Leakance of the Block-Centered Flow Package.
TKV[1]	NLL(I); DATA SET 2B		Model inputs described for PID=T and KV.
ANIV[1]	NLL(I)		Vertical anisotropy used to calculate leakance of the Block-Centered Flow Package.
S1[1]	NLL(I); DATA SET 2B		Storage coefficient or specific yield of the Block-Centered Flow Package[3].
RCH	NLL(1); DATA SET 2B	NLL(I)	Recharge rate of the Recharge Package.
ETM	NLL(1); DATA SET 2B	NLL(I)	Maximum ET rate of the Evapotranspiration Package.

[1]The natural log of the indicated quantity can be defined as the parameter to be estimated using LN in DATA SET 2 of the INPUT FILE.

[2]If LAYCON of the Block-Centered Flow Package=0, the T model input is transmissivity. If LAYCON=1, the T model input is hydraulic conductivity.

[3]If LAYCON of the Block-Centered Flow Package=0, the S1 model input is the storage coefficient. If LAYCON=1, the S1 model input is specific yield.

DATA SET 2A is also required for parameters used to define constant hydraulic-head boundaries (PID=CH). In this circumstance, only row and column numbers are read from DATA SET 2A, and the multiplicative constants are replaced with numbers that indicate cells at which constant hydraulic head is to be linearly interpolated from specified end points, the end points, and the constant hydraulic-head parameters defined at the end points. The linear interpolation is only allowed along rows or columns, as shown in figure A1. Interpolation along diagonals and around corners is prohibited. The model layers in which the constant hydraulic-head cells occur are listed in part of array NLL, of DATA SET 2. There are restrictions on how constant hydraulic-head boundaries that span multiple model layers can be defined by the Parameter-Estimation Package, as shown in figure A2. The relevant rules

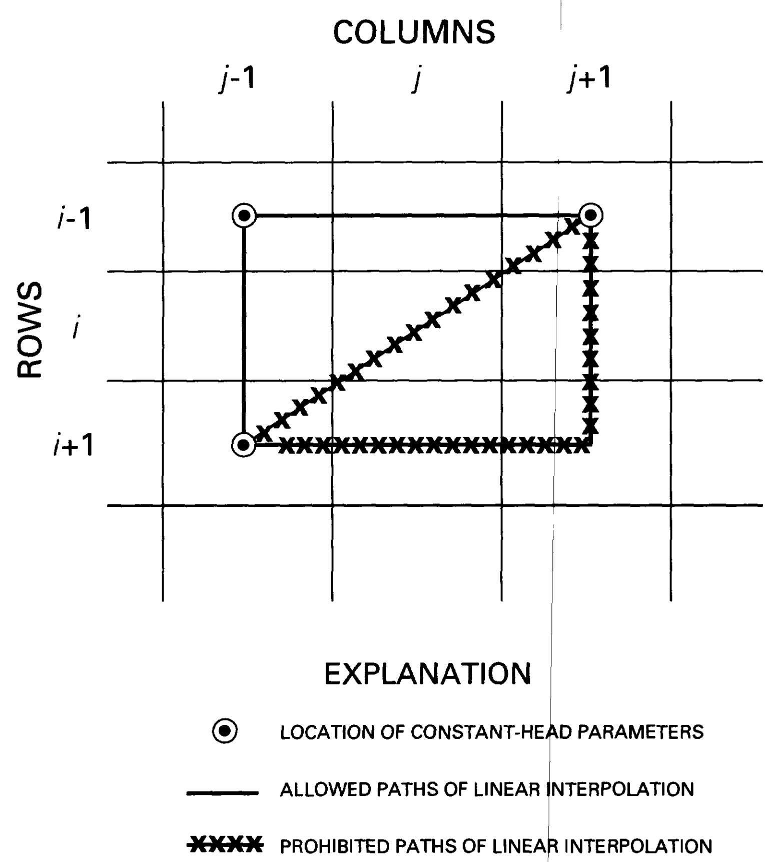


Figure A1.--Allowed and prohibited paths of linear interpolation for constant hydraulic-head parameter definition.

portrayed are: (1) A constant hydraulic-head parameter affects the constant hydraulic-head cells at rows and columns listed in DATA SET 2A; (2) as a consequence of (1), a constant hydraulic-head parameter affects cells in the same rows and columns in all layers involved; (3) affected cells can be specified as constant hydraulic head or inactive, but cannot be active (McDonald and Harbaugh, 1988, p. 4-2); and (4) vertical linear variations are not allowed.

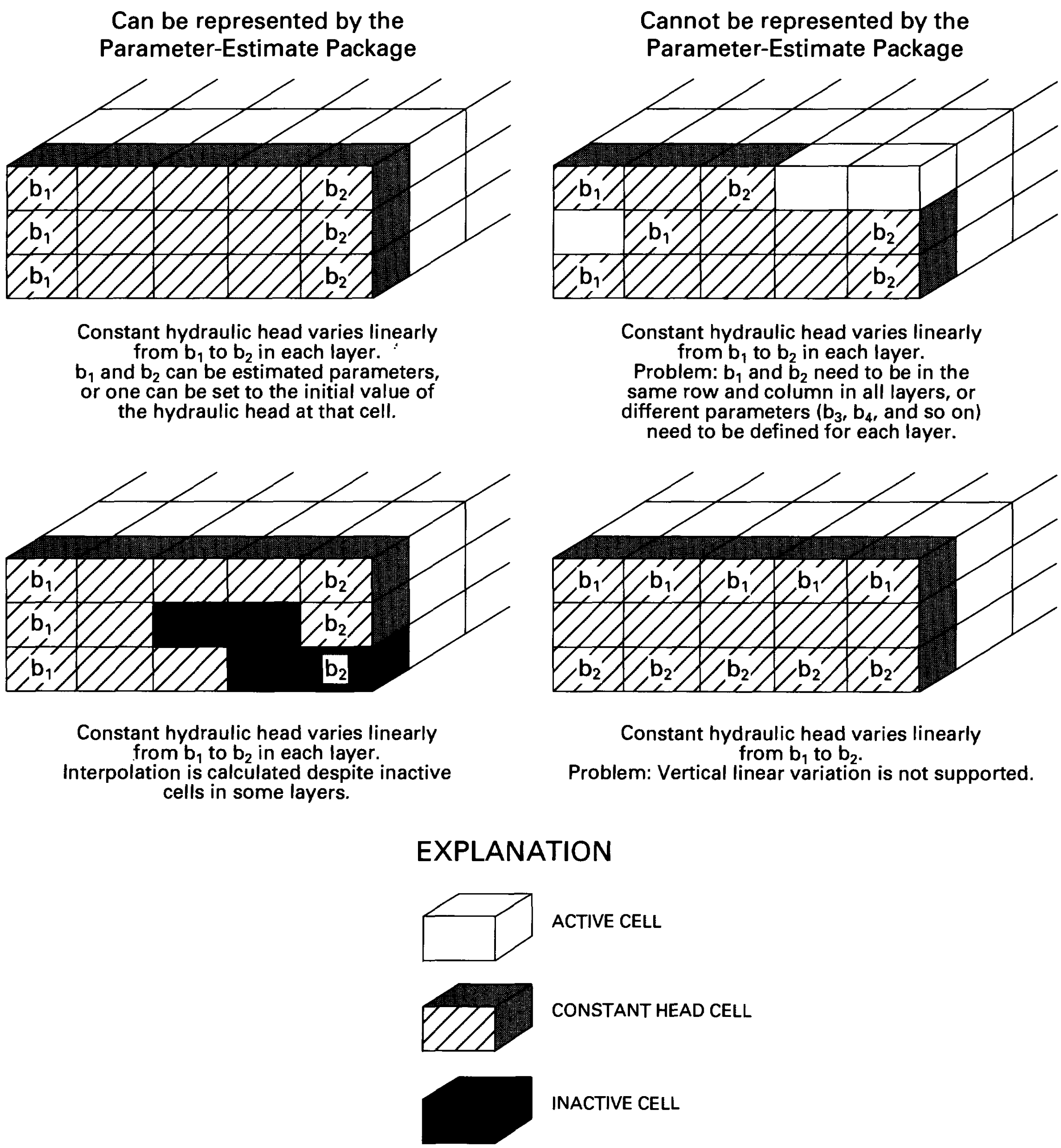


Figure A2.--Circumstances that can and cannot be represented by the Parameter-Estimation Package using PID=CH (DATA SET 2A).

If the MODFLOW input instructions specify that the model input be defined by arrays, as occurs for most aquifer properties (PID=T, KV, TKV, and S1), recharge (PID=RCH), and maximum evapotranspiration (PID=ETM), repetitions of DATA SET 2B can follow the applicable DATA SET 2. The number of repetitions of DATA SET 2B depends on how many model layers are affected by this parameter and whether defaults that can be activated using array NLL of DATA SET 2 are used. A more detailed discussion of how model inputs are calculated from parameters using DATA SET 2B follows the discussion of DATA SETS 3 and 4.

DATA SET 3 is composed of NROW × NCOL (number of rows in the finite-difference grid by number of columns) multiplication arrays. Multiplication arrays can be used to calculate model inputs defined by PID = T, KV, TKV, S1, RCH, and ETM. Multiplication array 1 is the first array in DATA SET 3, multiplication array 2 is the second, and so on. Multiplication arrays are referred to in DATA SET 2B. A mulitplication array can, for example, equal the cell-by-cell thickness of a confined model layer with homogeneous hydraulic conductivity. If the parameter equals the hydraulic conductivity, the parameter times the multiplication array would equal the transmissivity array, which is the required model input.

DATA SET 4 is composed of NROW × NCOL zone arrays. Zone arrays can be used to define the areas in which parameters are used to calculate model inputs defined by PID = T, KV, TKV, S1, RCH, and ETM. Zone array 1 is the first array in DATA SET 4, zone array 2 is the second, and so on. Zone arrays are referred to in DATA SET 2B. A zone array can, for example, be used to indicate that a parameter representing one hydraulic conductivity is to be used to calculate transmissivity in part of a confined model layer, and that a parameter representing another hydraulic conductivity is to be used to calculate transmissivity in the rest of that layer. Note that a mulitplication array with elements equal to cell-by-cell layer thickness would be used to calculate transmissivity from the hydraulic conductivities.

The calculation of model inputs from parameters with PID=T, KV, TKV, S1, RCH, and ETM is defined using array NLL of DATA SET 2, variables SFAC, LM, and LZA, and array LZ of DATA SET 2B, multiplication arrays from DATA SET 3, and zone arrays from DATA SET 4. These variables and arrays are described in the section "Instructions for the INPUT FILE," and their use is illustrated in figures A3A, A3B, and A3C for a model with five rows and five columns. Note that for subsequent parameter-estimation iterations, the parameter value changes but the other values stay the same. If the same cell is included under too many equivalent PID's, computed sensitivities will be wrong. This error is checked for by the program when ICHECK of LINE 4 is greater than zero.

EXAMPLE A

1) From DATA SET 2: PID=RCH

2) From DATA SET 8: Parameter value = 1.0

3) From DATA SET 2: NLL(1)=1
(One value > 0, so one DATA SET 2B follows this DATA SET 2)

4) From DATA SET 2B: SFAC=1×10^{-8}

5) From DATA SET 2B: LM=0 and LZA=0

6) Combining (1) through (5), the recharge rates calculated from this parameter are:

$$\begin{bmatrix} 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} \\ 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} \\ 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} \\ 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} \\ 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} & 1\times10^{-8} \end{bmatrix}$$

Figure A3.--Three examples of calculating model-input values from parameter values by using PID and NLL(I) from DATA SET 2; SFAC, LM, LZA, and LZ(I) from DATA SET 2B; multiplication and zone arrays from DATA SETS 3 and 4; and parameter values from DATA SET 8 for a model with five rows and five columns. (A zero in the final array indicates that the parameter does not affect the model-input value at that cell.)

EXAMPLE B

1) From DATA SET 2: PID=T

2) From DATA SET 8: Parameter value = 2.0

3) From DATA SET 2: NLL(1)=4
(One value > 0, so one DATA SET
2B follows this DATA SET 2)

4) From DATA SET 2B: SFAC=10.0

5) From DATA SET 2B: Use multiplication array LM=2
Multiplication array 2 from DATA SET 3:

10.	10.	10.	10.	10.
9.	9.	9.	9.	9.
8.	8.	8.	8.	8.
7.	7.	7.	7.	7.
6.	6.	6.	6.	6.

6) From DATA SET 2B: Use zone array LZA=1, zones LZ(1)=3 and LZ(2)=4
Zone array 1 from DATA SET 4:

1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5
1	2	3	4	5

7) Combining (1) through (6), the transmissivity values
calculated from this parameter for layer 4 are:

0.	0.	200.	200.	0.
0.	0.	180.	180.	0.
0.	0.	160.	160.	0.
0.	0.	140.	140.	0.
0.	0.	120.	120.	0.

Figure A3.--Three examples of calculating model-input values from parameter values by using PID and NLL(I) from DATA SET 2; SFAC, LM, LZA, and LZ(I) from DATA SET 2B; multiplication and zone arrays from DATA SETS 3 and 4; and parameter values from DATA SET 8 for a model with five rows and five columns. (A zero in the final array indicates that the parameter does not affect the model-input value at that cell.)--Continued.

EXAMPLE C

1) From DATA SET 2: PID=TKV

2) From DATA SET 8: Parameter value = 1.0

3) From DATA SET 2: NLL(1)=2, NLL(2)=-2, NLL(3)=-2
(One value > 0, so one DATA SET 2B follows this DATA SET 2)

4) From DATA SET 2B: SFAC=0.01

5) From DATA SET 2B: LM=0

6) From DATA SET 2B: Use zone matrix LZA=1, zone LZ(1)=2
Zone array 1 from DATA SET 4:

1	1	1	2	2
1	1	1	2	2
1	1	1	2	2
1	1	1	2	2
1	1	1	2	2

7) Combining (1) through (6), this parameter contributes to the model inputs as follows:

Vertical leakance between layers 1 and 2:

0.	0.	0.	.01	.01
0.	0.	0.	.01	.01
0.	0.	0.	.01	.01
0.	0.	0.	.01	.01
0.	0.	0.	.01	.01

Layer 2 transmissivity:

1.	1.	1.	1.	1.
1.	1.	1.	1.	1.
1.	1.	1.	1.	1.
1.	1.	1.	1.	1.
1.	1.	1.	1.	1.

Vertical leakance between layers 2 and 3:

1.	1.	1.	1.	1.
1.	1.	1.	1.	1.
1.	1.	1.	1.	1.
1.	1.	1.	1.	1.
1.	1.	1.	1.	1.

Figure A3.--Three examples of calculating model-input values from parameter values by using PID and NLL(I) from DATA SET 2; SFAC, LM, LZA, and LZ(I) from DATA SET 2B; multiplication and zone arrays from DATA SETS 3 and 4; and parameter values from DATA SET 8 for a model with five rows and five columns. (A zero in the final array indicates that the parameter does not affect the model-input value at that cell.)--Continued.

Parameters with PID=T, TKV, and S1 can be defined over several layers that might not have the same LAYCON value (see footnotes of table A1), but the defined parameter needs to be the same in all the layers. For example, if PID=T, and a confined and unconfined layer are included, the estimated parameter could be hydraulic conductivity. SFAC of DATA SETS 2B and multiplication arrays from DATA SET 3 are used to calculate the model input appropriate for each model layer from the parameter.

If hydrogeologic units pinch out, but the numerical layers continue into the pinched-out areas to maintain the hydraulic connection between underlying and overlying layers, the aquifer properties of the pinched-out areas generally are not included under an estimated parameter.

The model input identified by PID=TKV includes the model inputs identified by PID=T and PID=KV. This option enables the modeler to define a functional relation between the horizontal hydraulic conductivity (LAYCON=1) or transmissivity (LAYCON=0) of one or more model layers and the overlying vertical leakance, the underlying vertical leakance, the vertical leakance between layers (if there is more than one layer), or all of these. PID=ANIV is provided to enhance the applicability of the TKV option. To illustrate the use of PID=KV, TKV, and ANIV, consider the physical situations that vertical leakance is intended to represent, as discussed by McDonald and Harbaugh (1988, p. 5-11 to 5-18) and shown in figure A4.

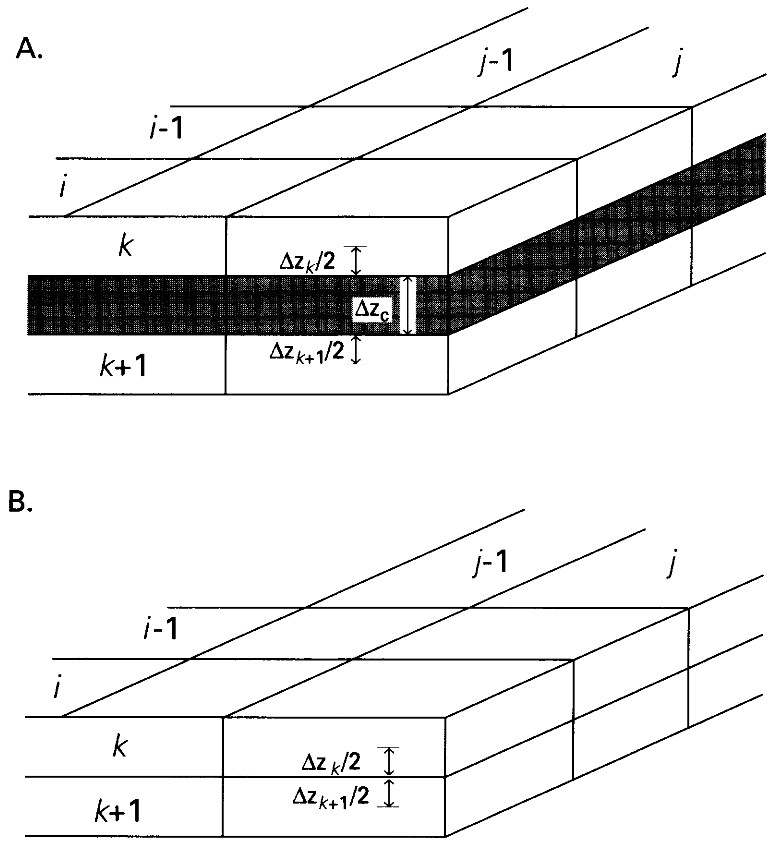
In situation (A) of figure A4, the vertical leakance between layers k and k+1 is calculated as (McDonald and Harbaugh, 1988, p. 5-17):

$$\mathrm{Vcont}_{i,j,k+\frac{1}{2}} = \left(\frac{\Delta z_k/2}{a_1 K_{i,j,k}} + \frac{\Delta z_c}{KZ_{c_{i,j,k+\frac{1}{2}}}} + \frac{\Delta z_{k+1}/2}{a_2 K_{i,j,k+1}} \right)^{-1} \qquad \text{(A1)}$$

where $K_{i,j,k}$ is the horizontal hydraulic conductivity of the cell at row i, column j, and layer k;

a_1 and a_2 are vertical to horizontal anisotropies; and

$KZ_{c_{i,j,k+\frac{1}{2}}}$ is the vertical hydraulic conductivity of the confining unit between layers k and k+1.



EXPLANATION

i, *j*, *k*	CELL ROW, COLUMN, AND LAYER NUMBER, RESPECTIVELY
Δz_k, Δz_{k+1}	THICKNESS OF MODEL LAYER *k* and *k*+1, RESPECTIVELY
Δz_c	THICKNESS OF CONFINING UNIT

Figure A4.--Subsurface layering represented by vertical leakance (A) with, and (B) without a distinct confining unit.

If a_1 and a_2 are known, and $K_{i,j,k}$, $KZ_{c_{i,j,k+\frac{1}{2}}}$, and $K_{i,j,k+1}$ are estimated independently, two TKV identifiers and one KV identifier need to be used. The two TKV identifiers are used to represent the contribution of $K_{i,j,k}$ and $K_{i,j,k+1}$ to $Vcont_{i,j,k+\frac{1}{2}}$, and the KV identifier is used to represent the contribution of $KZ_{c_{i,j,k+\frac{1}{2}}}$. For the two TKV identifiers, the product of the two parameter values and all multiplicative factors need to equal the individual leakances $(a_1K_{i,j,k})/(\Delta z_k/2)$ and $(a_2K_{i,j,k+1})/(\Delta z_{k+1}/2)$, and the appropriate position, I, of NLL(I) needs to be used (see explanation for DATA SET 2 in "Instructions for the INPUT FILE"). For the KV identifier, the product of the parameter value and all multiplicative factors need to equal $KZ_{c_{i,j,k+\frac{1}{2}}}/\Delta z_c$. Terms from the three identifiers are combined correctly by the Parameter-Estimation Package to produce equation (A1).

Alternatives to the above scenario depend on what is known about the components of equation (A1). If a_1 or a_2 or both are unknown, they cannot be estimated uniquely because then a single value of $Vcont_{i,j,k+\frac{1}{2}}$ could be produced by many different combinations of values of $KZ_{c_{i,j,k+\frac{1}{2}}}$, and a_1 or a_2 or both. ($K_{i,j,k}$ and $K_{i,j,k+1}$ are not included because they are used to calculate the transmissivities of the adjoining layers and thus independently impact the flow system if horizontal flow in the layers is significant.) Although individual components of equation (A1) cannot be estimated if a_1 or a_2 or both are unknown, an effective vertical hydraulic conductivity or leakance can be estimated. For example, if a_1 is unknown, equation (A1) can be expressed as:

$$Vcont_{i,j,k+\frac{1}{2}} = \left(\frac{\Delta z_E}{KZ_{E_{i,j,k+\frac{1}{2}}}} + \frac{\Delta z_{k+1}/2}{a_2K_{i,j,k+1}} \right)^{-1} \qquad \text{(A2)}$$

where Δz_E is an effective thickness, and $KZ_{E_{i,j,k+\frac{1}{2}}}$ is an effective vertical hydraulic conductivity. In this circumstance, one KV identifier and one TKV identifier are used to define the parameters. For the KV identifier, the product of the parameter value and all multiplicative factors needs to equal $KZ_{E_{i,j,k+\frac{1}{2}}}/\Delta z_E$. For the TKV identifier, the product of the parameter value and all multiplicative factors needs to equal $(a_2K_{i,j,k+1})/(\Delta z_{k+1}/2)$.

If a_2 also is unknown, one KV identifier is used to estimate one effective vertical hydraulic conductivity or leakance. The product of the parameter value and all multiplicative factors needs to equal $KZ_{E_{i,j,k+\frac{1}{2}}}/\Delta z_E$.

If it is known that the $KZ_{c_{i,j,k+\frac{1}{2}}}$ term dominates $Vcont_{i,j,k+\frac{1}{2}}$, the smaller terms can be neglected and one KV identifier can be used. The product of the parameter and all multiplicative factors needs to equal $KZ_{c_{i,j,k+\frac{1}{2}}}/\Delta z_c$.

In situation (B) of figure A4, the vertical leakance is calculated as (McDonald and Harbaugh, 1988, p. 5-13):

$$Vcont_{i,j,k+\frac{1}{2}} = \left(\frac{\Delta z_k/2}{a_1 K_{i,j,k}} + \frac{\Delta z_{k+1}/2}{a_2 K_{i,j,k+1}} \right)^{-1} . \tag{A3}$$

If a_1 and a_2 are known and $K_{i,j,k}$ and $K_{i,j,k+1}$ are estimated independently, two TKV identifiers are used, as discussed previously.

If a_1 and a_2 are known and one parameter is used to estimate both $K_{i,j,k}$ and $K_{i,j,k+1}$, one TKV identifier is used. For example, if $cK_{i,j,k} = K_{i,j,k+1}$, where c is a known constant and $K_{i,j,k}$ is the parameter,

$$Vcont_{i,j,k+\frac{1}{2}} = K_{i,j,k}[(\Delta z_k/2a_1) + (\Delta z_{k+1}/2a_2 c)]^{-1}. \tag{A4}$$

The product of the parameter value and the multiplicative factors need to equal equation (A4).

If a_1 and a_2 are unknown but satisfy $ca_1=a_2$,

$$Vcont_{i,j,k+\frac{1}{2}} = a_1 \left[\left(\frac{\Delta z_k/2}{K_{i,j,k}} + \frac{\Delta z_{k+1}/2}{cK_{i,j,k+1}} \right)^{-1} \right], \tag{A5}$$

and the ANIV parameter identifier can be used to define a_1 as a parameter. In the Parameter-Estimation Package, the TKV identifier(s) (one if $K_{i,j,k}$ and $K_{i,j,k+1}$ are calculated using one parameter; two if they are not) are used to calculate the term in the brackets, which is then multiplied by the parameter value of the ANIV identifier. Under the TKV identifier(s), the product of all multiplicative factors used to define vertical leakance and the relevant parameter value needs to equal $K_{i,j,k}/(\Delta z_k/2)$ or $cK_{i,j,k+1}/(\Delta z_{k+1}/2)$. Note that the multiplicative factor which relates the two anisotropies is included in a TKV identifier.

If either a_1 or a_2 is unknown in situation B of figure (A4), it could be uniquely estimated, but the Parameter-Estimation Package is not designed to solve this problem. In this situation, an effective vertical hydraulic conductivity or vertical leakance defined using a KV identifier needs to be estimated, as discussed above.

If a_1 and a_2 are unknown and are not related as above, they cannot be uniquely estimated. An effective vertical leakance defined using a single KV identifier needs to be estimated.

For PID=T, S1, RCH and ETM, the final model-input value can be calculated by summing the contributions from more than one parameter, as illustrated for a simple case in figure 3 and discussed in the section "Parameterization". Additions are performed differently for vertical leakances, as discussed above, but the following procedure still is used. To allow for additive options, the following procedure is implemented whether or not an additive option is used for a particular model input. If any PID=T, KV, TKV, S1, RCH, or ETM, the Parameter-Estimation Package sets the associated model input to zero at all cells, and then calculates values using DATA SETS 2 and 2B and initial (DATA SET 8) or estimated parameter values. Thus, if PID=T for any parameter, transmissivity is set to zero at all cells in confined layers and hydraulic conductivity is set to zero at all cells in unconfined layers. For parameters that can be time dependent (PID=RCH or ETM), the model input is set to zero at all time steps. If some of the additive contributions are assumed to be known and, therefore, are not to be estimated, they still need to be defined using DATA SETS 2 and 2B, and need to be among the last repetitions of these data sets. A negative group number in DATA SET 9 (see section "IWPG") is used to exclude the related parameter from updating by nonlinear regression. With the additive option, the same cell can appear under more than one DATA SET 2. The check initiated using ICHECK of LINE 4 of the INPUT FILE can be used to verify that cells are repeated as many times as expected.

Parameters defined using PID=Q, KRB, KDR, KST, GHB, CH, RCH, and ETM apply at time steps specified by the user. Except for PID=CH, the time steps are specified using all but the first element of array NLL of DATA SET 2 and can be continuous or discontinuous. Parameters defined using PID=CH can only apply at steady state, under transient conditions, or both, as specified by NLL(2).

Kriging can be used to parameterize spatially variable model inputs by producing multiplication matrices using a kriging algorithm. If the kriging is to be accomplished using estimated and known point values (de Marsily and others, 1984; Keidser and others, 1990), the known point values are included in the last repetitions of DATA SET 2, and the related IWPG value is negative (see section "IWPG" and the instructions for DATA SET 9). The multiplication array for this unestimated parameter can be produced by a kriging routine using a data set that includes the known point values and zeros for the estimated point values. The value of the unestimated parameter in DATA SET 8 would equal 1.0. For estimated point values that are estimated independently, each needs to be included in a repetition of DATA SET 2. The associated multiplication arrays can be produced by a kriging routine using a data set in which the point of interest has a value equal to one, and all other points have values equal to zero. After estimating point values, an experimental variogram using the measured and estimated point values needs to be produced and compared with the variogram used to do the kriging.

A second method of parameterization with kriging combines kriging with zonation (Keidser and others, 1990). In this method, the final estimated model-input values at a grid cell are calculated as the initial kriged value plus $b_{\ell}\sigma^*$, where b_{ℓ} is a zoned, estimated parameter, and σ^* is the cell variance calculated by kriging using the initial set of point values of the model input. In this situation, the multiplication array for b_{ℓ} is the variance map produced by kriging, and a zonation array is used to define its applicable area. The parameter b_{ℓ} can be negative or positive, and the associated parameter number needs to be included in DATA SET 1C if the PID=T, KV, TKV, S1, or ETM. The initial kriged model-input distribution is included by using one of the last repetitions of DATA SET 2; set the associated parameter value of DATA SET 8 to 1.0, set the associated IWPG value of DATA SET 9 to a negative number, and set the associated multiplication array equal to the kriged model-input values.

Keidser and Rosbjerg (1991) used the two kriging methods discussed above and zonation to parameterize transmissivity in a transport problem. In their examples, they found that the first kriging method, in which individual point values are estimated, produced excessive small-scale variations in the transmissivity field caused by local overfitting. Including zonation resolved this problem.

EVH, EVF, and EV

EVH, EVF, and EV of DATA SET 5 are the estimated values of σ^2, σ_h^2, and σ_f^2 (σ^2 needs to equal σ_h^2 or σ_f^2) of equations (17) and (22), and are used to calculate the weights of the dependent-variable observations as described in the explanations for DATA SETS 6 and 7A in "Instructions for the Input File." As discussed in the section "Structure and Use of the Weight Matrix", the calculated error variance printed at the end of the output of the Parameter-Estimation Package needs to approximately equal EV.

TOFF

TOFF (DATA SETS 6, 6C and 7A of the INPUT FILE) is the time-step offset that identifies a time after the end of the time step read as NDER(4) in DATA SETS 6 or 6C, or IQOB in DATA SET 7A, but before the end of the following time step. TOFF equals the fraction of the next time step that would have to occur to reach the desired time, and values of TOFF need to range from 0.0 to 1.0. As an example, consider a time 1.2 days into a simulation in which time step 3 ends at 1 day, and time step 4 ends at 1.3 days. In this example, the time step (NDER(4) or IQOB) equals 3 and TOFF=0.67. Values for the desired time are calculated by linearly interpolating between the values calculated at the end of the two time steps (fig. 5).

IWPG

The group numbers of DATA SET 9 are used to print residuals of prior parameter estimates specified using DATA SET 10, or to omit a defined parameter from parameter estimation. The latter is accomplished by making the group number negative. All group numbers larger than or equal to zero need to precede any negative group numbers, and the parameters are omitted from regression in the Parameter-Estimation Package by reducing NP, which is initially read from LINE 3.

TOL

TOL is the first parameter-estimation closure criterion discussed in the section "Convergence Criteria." Modified Gauss-Newton optimization converges if the fractional change of all parameters, as calculated by equation (50a), is less than or equal to TOL. TOL commonly equals 0.01 or 0.001 (each calculated change cannot exceed 1.0 or 0.1 percent of the parameter value), but is problem dependent. Too small a value will have little impact on estimated parameters and might substantially increase execution time. If TOL is greater than the coefficients of variation of the parameters, the calculated standard deviation might be too small.

LPRINT

If LPRINT>0, eigenvalues and eigenvectors of the variance-covariance matrix scaled by the parameter values (see section "Analysis of results for linear problems") are calculated and printed. If LPRINT=1, subroutines DE2CSF and DEPISF from the IMSL(1987) library are used to do the calculation and compute a performance index; the IMSL library needs to be included when compiling MODFLOWP (see section, "Compiling and Loading MODFLOWP" of Appendix B).

If LPRINT=2 or 3, subroutines TRED2, TQLI, and EIGSRT from Press and others (1989) are used to calculate eigenvalues and eigenvectors; these routines are included in the Parameter-Estimation Package. If LPRINT=3, the accuracy of the eigenvalues and eigenvectors from TRED2, TQLI, and EIGSRT is tested using the basic definition of eigenvalues and eigenvectors, as suggested in an example on the FORTRAN example diskette associated with Press and others (1989). Given a matrix $\underline{\underline{A}}$ of order n, n sets of eigenvalues and the associated eigenvectors need to satisfy $\underline{\underline{A}}\ \underline{x}_j=\lambda_j \underline{x}_j$, j=1,n, where λ_j is an eigenvalue, and $\underline{x}_j$ is the associated eigenvector. The accuracy of the eigenvalues and eigenvectors is tested by calculating the product of the scaled variance-covariance matrix on the parameters and the eigenvectors (equivalent to $\underline{\underline{A}}\ \underline{x}_j$ above), and comparing each element of the resulting vector to the corresponding element of $\underline{x}_j$. If the ratio equals the eigenvalue, λ_j, the calculated eigenvalue and eigenvector are accurate.

The performance index from the IMSL library is more convenient than the test conducted when TRED2, TQLI, and EIGSRT are used, so use of LPRINT=1 is suggested when the IMSL library is available.

Checking the INPUT FILE

To ensure that parameters are properly defined in the INPUT FILE, the following procedure can be used as a check and is STRONGLY RECOMMENDED FOR ALL APPLICATIONS OF THIS PACKAGE.

1. Construct MODFLOW package input files and run MODFLOWP without the Parameter-Estimation Package (IUNIT(15)=0).

2. Modify the MODFLOW input files (see section, "Changes to Input Files from Other Packages"), construct the INPUT FILE, and run MODFLOWP with the Parameter-Estimation Package (IUNIT(15)=the FORTRAN unit number of the INPUT FILE) and IPAR<0 (LINE 6 of the INPUT FILE).

3. The hydraulic heads and volumetric budgets for the entire model produced by (1) and (2) need to be IDENTICAL except for changes that are smaller in magnitude than the convergence criteria of the solver used. If they differ by more, an inconsistency exists between the two simulations and generally indicates errors in DATA SETS 2, 2A, 2B, 3, and 4 of the INPUT FILE, which are used to define the parameterization. Resolve all inconsistencies to ensure that the INPUT FILE is correct before proceeding!

To help in debugging, the model-input arrays produced by the Parameter-Estimation Pakcage using the initial parameter estimates for PID=T, KV, TKV, S1, RCH, and ETM can be printed using IPRNP of LINE 4 of the INPUT FILE. If IPRNP equals IPRN used when reading the arrays for the model run without parameter estimation (McDonald and Harbaugh, 1988, p. E-1), the arrays printed from the two runs need to be identical to four or five significant digits.

Outline of the INPUT FILE

The following outline briefly reviews the purpose and order of the lines and data sets of the INPUT FILE. The test cases at the end of this appendix provide examples of how the INPUT FILE is constructed.

LINE	Description	Number of repetitions
1	Title	1
2	Title	1
3	Data used to dimension arrays for parameterization	1
4	Data used in checking the parameterization	1
5	Data used to dimension arrays for observations	1
6	Data used to define what is to be calculated and how it is to be calculated	1
7	FORTRAN unit numbers for input or output files	1
8	Data that indicate if DATA SETS 1A, 1B, and 1C are read	1

Outline of the INPUT FILE--Continued

DATA SET	Description	Number of repetitions
1A	FORTRAN input unit numbers identified within the Recharge or Evapotranspiration Packages.	1
1B	List of time steps at which all hydraulic heads and sensitivities are set to 0.0.	1
1C	List of usually positive parameters which are allowed to be negative.	1
2; immediately followed by associated 2A or repetitions of 2B, as needed.	Parameter definitions used to calculate model inputs from parameters. A typical sequence of data sets would be: 2, 2A, 2, 2B, 2B, 2, 2, 2A.	As many as are needed to define the NP parameters; NP if there are no CH parameters.
3	Multiplication arrays used to define parameters. Referred to in DATA SET 2B.	NMM times (LINE 3). Array 1 is first, 2 is second, and so on.
4	Zone arrays used to define parameters. Referred to in DATA SET 2B.	NZM times (see LINE 3). Array 1 is first, 2 is second, and so on.
5	σ_h^2, σ_f^2, unit numbers for reading observations, and σ^2.	1
6; immediately followed by associated 6A, 6B, and 6C, as needed.	Observations of hydraulic head or temporal changes in hydraulic head. DATA SET 6 defines the obeservation location, and, unless DATA SETS 6B and 6C follow, the observation and its time and statistic. DATA SET 6A is used for multilayer observations. A typical sequence of data sets would be: 6, 6, 6, 6A, 6, 6B, 6C, 6.	DATA SET 6 needs to be repeated for each observation location, and can be repeated as many as NH times.
7; immediately followed by 7A and 7B.	Observations of head-dependent boundary gains and losses. A typical sequence of data sets would be: 7, 7A, 7B, 7, 7A, 7B. Note that each DATA SET 7 is always followed by a DATA SET 7A and 7B.	These three DATA SETS are repeated as a group NQ times (LINE 5).

Outline of the INPUT FILE--Continued

DATA SET	Description	Number of repetitions
8	Initial parameter values.	1
9	Parameter group numbers.	1
10	Allows initial parameter values to be used as prior estimates on the parameters (see eq. 28).	1
11	Flag that indicates whether the statistics of DATA SETS 9 and 12 are variances, standard deviations, or coefficients of variation.	1
12	Prior estimates that include more than one parameter (eq. 29) or that are independent of the initial parameter values.	MPR (LINE 5).
13	Data for the nonlinear regression.	1
14	Factors used to scale parameters for conjugate-direction optimization methods.	1

Instructions for the INPUT FILE

FOLLOW PROCEDURE DESCRIBED IN THE PRECEDING SECTION TO CHECK THE INPUT FILE

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION
1	1-80	A80	TITLE	Title printed when the SEN part of the Parameter-Estimation Package is first active.
2	1-80	A80	TITLE	

Data used to dimension arrays for parameterization. Usually it is best to construct the repetitions of DATA SET 2 before trying to determine NSM, NSN, NLLI1, and LZI1.

3	1-5	I5	NP	Number of parameters (includes estimated parameters and parameters that are not modified by the regression).
	6-10	I5	NSM	Total number of NLL values larger than zero for PID=T, KV, TKV, and S1, (see DATA SET 2). Add one for any PID=RCH or ETM with NLL(1)>0.
	11-15	I5	NSN	Number of cell locations listed for parameter definition in the INPUT FILE.

NSN equals the sum of the absolute values of NLL(1) for PID=GHB, Q, KRB, KDR, KST, and CH in DATA SET 2.

	16-20	I5	NLLI1	Largest number of values used for any NLL vector of DATA SET 2.
	21-25	I5	LZI1	Largest number of values used for any LZ vector of DATA SET 2B.
	26-30	I5	NMM	Number of multiplication arrays in DATA SET 3. NMM=0 if there are no multiplication arrays.
	31-35	I5	NZM	Number of zone arrays in DATA SET 4. NZM=0 if there are no zone arrays.

Data used in checking the parameterization.

4	1-5	I5	IPRNP	Output format code (see table A2) for model-input array produced by using initial parameter estimates for PID=T, KV, TKV, S1, RCH, or ETM. If IPRNP$\le$0, no arrays are printed.

Arrays are only printed if parameter estimates are used in their calculation, and in DATA SET 2, NLL>0. See section "Model-Input Arrays Produced with the Initial Parameter Estimates."

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION
4	6-10	I5	ICHECK	Flag used to indicate whether to check (ICHECK>0) or not to check (ICHECK=0) for cells that are repeated more than NUMR times (see below) in DATA SET 2 for model inputs defined by arrays.

Check involves parameters with PID=T, KV, TKV, S1, RCH, and ETM.

This check is useful in determining whether cells have been repeated under different PIDs as many times as intended. Also can be used to check that all cells in the applicable layers are included as needed (see fifth paragraph from end of section "Parameter Definition: DATA SETS 2, 2A, 2B, 3, and 4"). Set ICHECK=0 when not being used to reduce computer execution time.

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION
	11-15	I5	NUMR	Number of times a cell can be repeated to define a single model input before an error message is printed.

Suggested usage: first set NUMR=1, and make sure repeated cells are as expected; then set NUMR=2, and make sure cells repeated more than once are as expected. Continue incrementing by 1 until the maximum number of repetitions is reached.

Data used to dimension arrays for observations.

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION
5	1-5	I5	NH	Number of observations of hydraulic head or changes in hydraulic head.
	6-10	I5	MOBS	Number of the NH locations and times that are multilayer (see section "Multilayer Hydraulic Heads" and instructions for DATA SET 6A).
	11-15	I5	MAXM	Maximum number of layers used for any of the MOBS locations (see instructions for DATA SET 6A).
	16-20	I5	NQ	Number of cell groups for which there are head-dependent boundary flow observations.
	21-25	I5	NQC	Total number of individual cells in the NQ groups (sum of absolute values of all NQCL in repetitions of DATA SET 7).
	26-30	I5	NQT	Total number of head-dependent flow observations (sum of all values of NQOB in repetitions of DATA SET 7).
	31-35	I5	MPR	Number of equations of prior information on the parameters that include more than one parameter (see eq. 29).

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION

Data used to define what is calculated and how it is calculated.

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION
6	1-5	I5	IPAR	Flag identifying what to calculate. If IPAR<0, hydraulic heads are calculated; if IPAR=0, hydraulic heads and sensitivity equations (ISN<0) or adjoint states (ISN>0) are calculated; if IPAR>0, parameter-estimation is performed. See section "IPAR".

For all IPAR values, model inputs calculated using the parameter values of DATA SET 8 replace the values from the input files for other packages, where applicable.

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION
	6-10	I5	ISN	Flag identifying which optimization and numerical methods are used. See section "ISN". Generally, ISN=-1.

If ISN<0, sensitivity-equation sensitivities are calculated, and parameter estimation is performed by the modified Gauss-Newton method. ISN can equal -1 or -2.

If ISN>0, the adjoint-state method is used to calculate the gradient of the objective function, and parameter estimation is performed by the conjugate-direction method. ISN can equal 1, 2, or 4.

ISN=-1 worked most efficiently for the test problems considered by Cooley (1985) and Hill (1990).

ISN>0 cannot be used if there are any water-table layers or if NQT>0

Use ISN=-1 if there are water-table layers and if cells go dry for any parameter-estimation iteration

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION
	11-15	I5	ISCALS	If ISCALS>0 or KPRINT>0 in DATA SET 13, printed sensitivity-equation sensitivities are scaled by multiplying by the parameter value and the square root of the weight. Scaled sensitivities can be used to to compare the relative influence of different parameters.
	16-20	I5	NOPT	When NOPT=1 and ISN<0, $\underline{R}$ of equation (51) is included in equation (50a) after NFIT (DATA SET 13) modified Gauss-Newton iterations or after two iterations fail to reduce the sum of squared, weighted residuals by SOSR×100 percent (DATA SET 13).

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION

FORTRAN unit numbers for input or output files.

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION
7	1-5	I5	IUHEAD	First output/input unit for hydraulic heads or sensitivity-equation sensitivities stored in binary format.

This unit and NP (LINE 3) or NPER (as specified in the Basic Package input file) sequentially numbered additional units are opened in MODFLOWP. An error message is printed and execution stops if the units are already open.

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION
	6-10	I5	IOUB	Output unit number for estimated parameter values.

File IOUB is used to print parameter estimates and selected statistics from all parameter-estimation iterations at the end of the output file, and to provide parameter values with which to restart the parameter-estimation routine (see section "Restarting the Parameter-Estimation Routine"). If IOUB=0, the parameter estimates and statistics are not written on a separate file and are not printed at the end of the output file. Generally, IOUB≠0.

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION
	11-15	I5	IUBD	Input unit for IBOUND arrays.

See items 3 and 4 of the section "Changes to Input Files from Other Packages."

If IUBD>0, NPER>1 in the Basic Package input file, and all layers are confined, two IBOUND arrays are read from FORTRAN unit IUBD by using U2DINN, which functions like U2DINT (McDonald and Harbaugh, 1988, p. 14-30) except that nothing is printed. The first IBOUND array is for steady-state conditions; the second IBOUND array is for transient conditions. If IUBD>0, and NPER≤1 or there are water-table layers, one IBOUND array is read. If IUBD=0, the IBOUND array specified in the Basic Package input file is used, and no other input file is required.
** If there are any water-table layers and cells go dry for any parameter-estimation interation, use IUBD>0 **.

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION
	16-20	I5	IUNHEA	IUNHEA≠0 if known hydraulic heads at constant hydraulic-head boundaries change during the simulation. See text in section "IUNHEA".

LINE	COLUMN	FORMAT	VARIABLE	DEFINITION

Data that indicate if DATA SETS 1A, 1B, and 1C are read.

8	1-5	I5	NRWD	Number of FORTRAN input units identified from within the Recharge and Evapotranspiration Packages using LOCAT. If NRWD>0, DATA SET 1A is read.

Files used as multiplication arrays (DATA SET 3) cannot be read by other packages and cannot be included here and in DATA SET 1A.

	6-10	I5	NZER	Number of time steps >0 at which hydraulic head equals zero. If NZER>0, DATA SET 1B is read.
	11-15	I5	NPNG	Number of parameters with PID=T, KV, TKV, S1, or ETM that can have negative values. If NPNG>0, DATA SET 1C is read.

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION

Omit DATA SET 1A if NRWD=0 (LINE 8).

1A	1 or more	16I5	IRWD	NRWD FORTRAN input unit numbers that are identified from within the Recharge or Evapotranspiration Packages using LOCAT.

Omit DATA SET 1B if NZER=0 (LINE 8).

1B	1 or more	16I5	IZER	NZER time steps at which all hydraulic heads and sensitivities are set to 0.0.

Used when estimating parameters with multiple superposition simulations (see item 8d in section "Changes to Input Files from Other Packages").

Omit DATA SET 1C if NPNG=0 (LINE 8).

1C	1 or more	16I5	IPNG	NPNG parameter numbers for which PID=T, KV, TKV, S1, or ETM, and parameters are defined with the second kriging method discussed at the end of section "Parameter Definition: DATA SETS 2, 2A, 2B, 3, and 4."

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION

DATA SETS 2, 2A, 2B, 3, and 4 are used to calculate model inputs from the parameters. See section "Parameter Definition: DATA SETS 2, 2A, 2B, 3 and 4." Repeat DATA SET 2, and DATA SET 2A or 2B, when required, until all NP parameters are defined.

NOTE: A parameter cannot be reliably estimated if the values calculated at all observation points are insensitive to that parameter, prior estimates are missing or inaccurate, or if the parameter is highly correlated with one or more other parameters (see "Analysis of Results for Linear Problems"). To avoid these problems, MINIMIZE THE NUMBER OF ESTIMATED PARAMETERS.

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION
2	1	A4,I1, 11I5 For additional lines, Use 16I5	PID, LN, NLL(I), I=1,NLLI1	Parameter identifier, natural log indicator, and vector used as described below. NLLI1 is specified on LINE 3.

PID needs to be left-justified and in upper-case letters. PID options are listed in table A1.

LN>0 indicates that the parameter to be estimated is the natural log of the defined quantity. Valid only for PID=GHB, KRB, KDR, KST, T, KV, TKV, and S1.

NLLI1 (LINE 3) values are read for the NLL array.

DATA SET	EXPLANATION
2	PID = ANI: The NLL(I) values equal the model-layer numbers to which this horizontal anisotropy applies; no additional records follow.

PID = Q, KRB, KDR, KST, and GHB: The absolute value of NLL(1) equals the number of cell locations that follow in DATA SET 2A. NLL(1)<0 indicates that FACTOR of DATA SET 2A will be set to 1.0 for all listed cells.

NLL(I),I=2,NLLI1, are used to indicate the time steps at which the pumpage rate or head-dependent boundary condition applies. The parameter starts at the beginning of time step NLL(2), and stops at the end of time step NLL(3); the parameter starts again at the beginning of time step NLL(4), and stops at the end of NLL(5); and so forth. If applied at one time step, NLL(I)=NLL(I+1), where I is an even number. For steady state, use zero.

For Q, KRB, KDR, KST, and GHB, unused values of NLL(I) (for example, if 3 values are needed to define the parameter, but NLLI1=6, there would be 3 unused values) need to be negative because a zero or blank is interpreted as steady state.

DATA SET	EXPLANATION
2	PID = CH: NLL(1) needs to be positive and equals the number of cells that follow in DATA SET 2A. NLL(2) equals the number of parameters defined under this identifier. NLL(3) indicates when the constant hydraulic-head boundary applies and is used as follows:

NLL(3)	When CH applies
0	Steady state only.
1	Transient only.
2	Both steady state and transient.

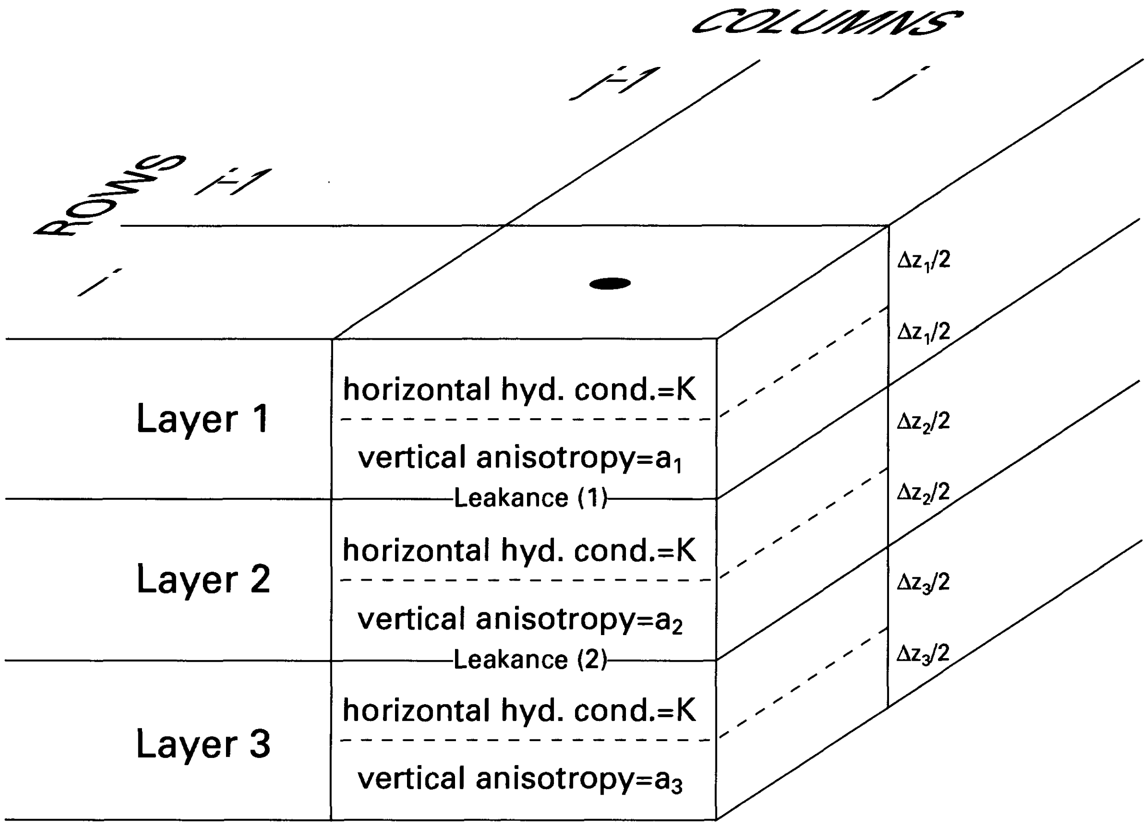
NLL(I), I=4,NLLI1, equal the layers in which this constant hydraulic-head boundary condition applies. Unused values need to be blank or zero; nonzero values need to precede blanks and zeros. See figures A1 and A2 for situations that can and cannot be represented with PID=CH.

PID = RCH and ETM: The components of NLL are not layer numbers; layer numbers are determined by using NRCHOP and array INRECH of the Recharge Package input file or by using NEVTOP and array IEVT of the Evapotranspiration Package input file (McDonald and Harbaugh, 1988, ch. 7 and 10). The components of NLL are used as follows: NLL(1)<0 indicates that the parameter equals the recharge or maximum evapotranspiration rate at all active cells where recharge or evapotranspiration apply, and no additional records follow; NLL(1)>0 indicates that the parameter is a multiplicative constant, applies over part of the modeled area, or both, and DATA SET 2B follows. NLL(I),I=2,NLLI1 are used for temporal definition as explained above for Q, KRB, KDR, KST, and GHB.

As above, unused values of NLL(I) need to be negative.

PID = T, KV, and S1: NLL is a list of layer numbers where: NLL(I)<0 indicates that the parameter equals the model-input value at all active cells of layer |NLL(I)|, and no additional records follow; NLL(I)>0 indicates that the parameter is a multiplicative constant, applies to only part of layer NLL(I), or both, and DATA SET 2B is read for each NLL(I)>0 in the order in which these positive values occur. Nonzero NLL values need to precede zeros and blanks.

DATA SET	EXPLANATION
2	PID = TKV: NLL is a list of layer numbers where positive and negative values are used as described for PID = S1, T, and KV. When I equals 1+3k, where k=0,1,2..., so that I=1,4,7..., NLL(I) is used to define vertical leakance between layer NLL(I) and the layer above (last term of eq. A1 or A2); when I equals 2+3k, so that I=2,5,8..., NLL(I) is used to define horizontal hydraulic conductivity or transmissivity; when I equals 3+3k, so that I=3,6,9..., NLL(I) is used to define vertical leakance between layer NLL(I) and the layer below (first term of eq. A1 and A2). To allow full generality, nonzero numbers can be preceded by zeros or blanks; no DATA SET 2B is read for a zero or blank. For example, NLL(I)=0,1,1,2,2,2,3,3 indicates that the horizontal hydraulic conductivity or transmissivity of layers 1, 2, and 3 are related to each other and to the vertical leakances between these three layers, and seven repetitions of DATA SET 2B would be read. If the parameter is defined as the horizontal hydraulic conductivities of the three layers, which are considered to be equal, and a_1, a_2, and a_3 are vertical anisotropies of the three layers, the seven repetitions would be defined as shown in figure A5.
	PID = ANIV: The NLL(I) values equal the layer numbers above the applicable vertical leakance. Vertical leakance is calculated using TKV identifiers, and then is multiplied by the value of the ANIV parameter. See equation A5.



Layer 1 is a water-table layer; therefore Δz_1 is approximate. K is the parameter. The eight components of array NLL are 0,1,1,2,2,2,3,3.

Seven repetitions of DATA SET 2B are required. For each repetition, K times SFAC of DATA SET 2B, and CNSTNT and the elements of the multiplication array of DATA SET 3 must equal: (description of required model inputs are in brackets)

1) K [horizontal hydraulic conductivity of layer 1]

2) $a_1K/(\Delta z_1/2)$

3) $a_2K/(\Delta z_2/2)$ $\}$ $\left[\text{Leakance (1)}=\left((\Delta z_1/2)/a_1K+(\Delta z_2/2)/a_2K\right)^{-1}\right]$

4) $K\,\Delta z_2$ [transmissivity of layer 2]

5) $a_2K/(\Delta z_2/2)$

6) $a_3K/(\Delta z_3/2)$ $\}$ $\left[\text{Leakance (2)}=\left((\Delta z_1/2)/a_2K+(\Delta z_3/2)/a_3K\right)^{-1}\right]$

7) $K\,\Delta z_3$ [transmissivity of layer 3]

Figure A5.--Use of repetitions of DATA SET 2B when PID=TKV.

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION
2A	Absolute value of NLL(1)	See below	CELS(I), I=1,3, FACTOR	List of cells where the parameter is used to calculate a model input.

Four format options are available to read DATA SET 2A to permit files developed for other packages of MODFLOW to be used more directly. The format is selected based on PID (DATA SET 2).

PID = Q: 4F10.0 is used to read the layer, row, column, and FACTOR of a cell.

PID = GHB, KRB, and KDR: 3F10.0, 10X, F10.0 is used to read the layer, row, column, and FACTOR of a cell.

PID = KST: 15X, 2F5.0, 25X, F10.0 is used to read the segment, reach, and FACTOR of a cell.

PID = CH: 10X, 3F10.0 is used to read the row, column, and FACTOR of a cell.

Note that integers and real numbers with digits to the right of the decimal point can be read using F10.0 and F5.0 formats.

FACTOR is used to calculate the model input for the PID in the preceding DATA SET 2 (see table A1) from the parameter value.

PID = Q, KRB, KDR, KST, and GHB: If NLL(1)>0, the model input equals the parameter value multiplied by the FACTOR of DATA SET 2A; if NLL(1)<0, the parameter is equivalent to the model input, and FACTOR is set to 1.0 by the program, regardless of the value specified in DATA SET 2A. The order of cells in DATA SET 2A needs to match the order of cells in the associated package input file; if cell 1 precedes cell 10 in the package input file, it needs to precede cell 10 in DATA SET 2A.

PID = CH: Values along a constant hydraulic-head boundary are linearly interpolated between cells with FACTOR>1.0 or FACTOR<0.0; at all other cells, FACTOR needs to equal zero. The parameters are the hydraulic heads at the center of the cells with FACTOR>1.0. Equal positive FACTORS indicate that the constant hydraulic heads at these and intervening cells are equal. Equal positive FACTORS need to be grouped; that is, they can be separated only by zeros. The head at a cell with FACTOR<0.0 remains at the value specified in Shead of the Basic Package input file.

All cells listed under a CH identifier need to be designated as constant head in the IBOUND array read from the Basic Package input file, or the appropriate IBOUND array(s) read from FORTRAN unit IUBD (LINE 7). See figures A1 and A2 for additional restrictions.

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION

Repeat DATA SET 2B once for each NLL(I)>0 in the preceding DATA SET 2 for PID=RCH, ETM, S1, T, KV, or TKV. The first DATA SET 2B is associated with the first NLL(I)>0, the second is associated with the second NLL(I)>0, and so on.

2B	Number of NLL(I)>0 in preceding DATA SET 2 for PID's listed above.	F10.0,I5, 14I5 After first line, use 16I5.	SFAC, LM, LZA, LZ(I), I=1, LZI1	Factor, multiplication array number (DATA SET 3), zone array number (DATA SET 4), and zone numbers (LZ(I), I=1,LZI1). LZI1 is specified on LINE 3. The F10.0 format allows the decimal point to be specified anywhere in the field.

For information about the use of SFAC, LM, LZA, and LZ(I), and a discussion of which cells need to be included in a DATA SET 2 and 2B if PID=T, KV, TKV, S1, RCH, or ETM for any parameter, see section "Parameter Definition: DATA SETS 2, 2A, 2B, 3, and 4."

If LM=0, a default multiplication array with all elements equal to 1.0 is used.

If LZA=0, zonation is not used to limit the areal extent of the parameter. The parameter will apply to all active cells with nonzero values in the multiplication array in the layer specified in DATA SET 2.

Zone numbers need to be positive, nonzero integers; in LZ(I), nonzero numbers need to precede blanks or zeros.

DATA SET	EXPLANATION
	Omit DATA SET 3 if no multiplication arrays are used (LINE 3: NMM=0).
3	NMM (LINE 4) NROW×NCOL multiplication arrays. Read by using module U2DREL of McDonald and Harbaugh (1988, p. 14-26 to 14-29; E-1). Multiplication array numbers are referenced by variable LM of DATA SET 2B. Multiplication array number 1 needs to be first, number 2 needs to be second, and so on. If the parameter is to be used to calculate the model input in only part of the area, zeros can appear in the multiplication array for uninvolved cells, or zonation (see DATA SET 4) can be used to limit the areal extent of the parameter. The product of the parameter, SFAC of DATA SET 2B, CNSTNT of U2DREL, and the nonzero entries of the multiplication array, needs to equal the model input described in table A1 (see also the section "Parameter Definition: DATA SETS 2, 2A, 2B, 3, and 4"). See figure A3 for an example of how multiplication arrays are used.
	Omit DATA SET 4 if no zone arrays are used (LINE 3:NZM=0).
4	NZM (LINE 4) NROW×NCOL zone arrays. Read by using module U2DINT of McDonald and Harbaugh (1988, p. 14-30 to 14-34; E-1). Zone array numbers are referenced by LZA of DATA SET 2B; zone numbers are referenced by LZ of DATA SET 2B. Zone array number 1 needs to be first, number 2 needs to be second, and so forth. Each zone array can include many zone numbers. For each DATA SET 2B, the specified zone array, LZA, needs to include all zone numbers referenced in the array LZ. See figures A3.A and A3.B for examples of how zone arrays are used.

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION
5	1	2F10.0, 2I5, F10.0	EVH, EVF, IUH, IUF, EV	Factors σ_h^2, σ_f^2 of equation (22), input unit numbers for hydraulic head and head-dependent boundary flow gain or loss observations (DATA SETS 6 and 7), and the estimated common error variance (σ^2 of eq. 17).

Either EVH or EVF needs to equal EV.

If equal to zero, IUH and IUF are set equal to the input unit number of the INPUT FILE.

DATA SET 6 and, if required, 6A and(or) 6B and 6C are read from input unit IUH (DATA SET 5), and needs to be repeated for each hydraulic-head observation location.

Transient data can be included by repeating DATA SET 6 or by using DATA SETS 6B and 6C.

Omit these DATA SETS if NH=0 (LINE 5).

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION
6	1	A4,1X, 4I5, 3F8.0, F10.0, F8.0,I5	DID, NDER(I), I=1,4; ROFF,COFF, TOFF, HOBS, WT, IWT	Data identifier (any four characters the user chooses), layer, row, column, time step (zero for steady-state), row and column offsets used to locate the observation point within a cell (must range from -0.5 to 0.5; see fig. 6), time-step offset (must range from 0.0 to 1.0; see fig. 5 and section "TOFF"), observed head, the value from which the weight of the observed head is calculated, and a flag to indicate whether WT is a scaled variance (IWT=0), a scaled standard deviation (IWT=1), or a scaled coefficient of variation (IWT=2).

When IPAR≤0 (LINE 6), HOBS, WT, and IWT are printed and used to calculate weighted residuals. WT needs to be greater than zero.

If NDER(1)<0, The observation is multilayer, and DATA SET 6A is read.

If NDER(4)<0, -NDER(4) time steps are involved, and DATA SETS 6B and 6C are read. DID, HOBS, and WT of DATA SET 6 are replaced when DATA SET 6C is read.

Weights are calculated as EV/(WT×EVH) for IWT=0, EV/(WT^2×EVH) for IWT=1, and EV/((WT×HOBS)^2×EVH) for IWT=2. EVH and EV are read from DATA SET 5.

In transient simulations, the steady-state IBOUND array is used to identify inactive cells for the interpretation with ROFF and COFF (fig. 6).

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION

Omit DATA SET 6A if NDER(1)>0 in the preceding repetition of DATA SET 6.

6A	1	8(I5, F5.0)	(MLAY(I), PR(I)), I=1, -NDER(1)	Layer numbers for a multilayer observation well, and the proportions assigned to each layer.

There can be as many as MAXM (LINE 5) layers; the proportions need to sum to 1.0.

The first layer is used to calculate interpolation coefficients for arbitrary locations within the finite-difference cell (see figs. 7 and 8).

Omit DATA SETS 6B and 6C if NDER(4)≥0 in the preceding DATA SET 6.

6B	1	I5	ITT	Flag for transient observations.

ITT	Description
1	Observed hydraulic heads are used for parameter estimation.
2	Observed initial hydraulic head and subsequent changes in hydraulic head are used for parameter estimation.

6C	-NDER(4) from DATA SET 6	A4,1X I5, F8.0, 3F10.0, I5	DID, NDER(4), TOFF, HOBS, WT_h, WT_{DD}, IWT	Data identifier, time step, time-step offset, observed head, values from which the weight for head or for drawdown (used when ITT=2) is calculated, and a flag that is used as described for DATA SET 6.

The first line of this DATA SET needs to be the earliest observed hydraulic head at this location used for parameter estimation. WT_h is always used for the first observation.

The time step read as NDER(4) in the first line of DATA SET 6C replaces the value of NDER(4) read in DATA SET 6.

When IPAR≤0, HOBS, WT_h, WT_{DD}, and IWT are printed and used to calculate weighted residuals. WT_h and WT_{DD} need to be greater than zero.

DATA SETS 7, 7A, and 7B are read from input unit IUF (DATA SET 5), and are repeated for each of the NQ cell groups (LINE 5) that define head-dependent boundaries with observed gains or losses. Omit these DATA SETS if NQ=0.

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION
7	1	3I5	IBT, NQOB,NQCL	Boundary type, number of times at which head-dependent boundary gains or losses are observed for this group, and number of cells in this group (fig. 9).

If NQCL<0, the FACTORS of DATA SET 7B are set equal to 1.0, and the number of cells is set equal to -NQCL.

IBT values: 1, River Package; 2, General-Head Boundary Package; 3, Stream Package; 4, Drain Package. If IBT=3, NQCL equals the number of reaches in the group.

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION
7A	NQOB	A4,1X, I5,F8.0 2F10.0, I5	DID,IQOB, TOFF, HOBS,WT, IWT	Data identifier (any four characters the user chooses), time step (zero for steady state), time-step offset (must range from 0.0 to 1.0; see fig. 5 and section "TOFF"), observed head-dependent boundary gain or loss (in L^3/T; losses indicate flow into the aquifer and are positive), the value from which the weight for observed flux is calculated, and a flag that is used as in DATA SET 6.

When IPAR≤0, HOBS, WT, and IWT are printed and used to calculate weighted residuals. WT needs to be greater than zero.

Weights are calculated as EV/(WT×EVF) for IWT=0, EV/(WT^2×EVF) for IWT=1, and EV/($(WT\times HOBS)^2$×EVF) for IWT=2. EVF and EV are read from DATA SET 5.

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION
7B	NQCL	See definition	QCLS(I), I=1,3, FACTOR	For IBT=1, 2, or 4 (DATA SET 7) 4F10.0 is used to read the layer, row, column, and FACTOR of each cell in the group. For IBT=3, 15X, 2F5.0, F10.0 is used to read segment, reach, and FACTOR.

Note that integers and real numbers with digits to the right of the decimal point can be read using F10.0 or F5.0 format.

FACTOR needs to be set to 1.0 unless only part of the gain or loss for the cell is to be included; then FACTOR is used to allocate the calculated gain or loss in this cell or reach (see eq. 4 and fig. 9).

If NQCL of DATA SET 7 is negative or FACTOR = 0.0, FACTOR is set to 1.0.

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION
8	1 or more	8F10.0	B(I), I=1,NP	Initial parameter values. B(1) is the initial value of the first parameter defined under the first repetition of DATA SET 2; B(2) is the initial value of the second parameter defined under a DATA SET 2, and so on.

Parameter values specified here are used to replace model-input values read from the input files for other packages.

Even if LN of the associated DATA SET 2 is not 0, do not log transform the value of the parameter specified in DATA SET 8; log transformations are done by the Parameter-Estimation Package.

**Convergence of parameter estimation can depend on the initial values used. If problems with convergence occur, see section "Common Problems" **

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION
9	1 or more	8I10	IWPG(I), I=I,NP	Parameter group numbers. See section "IWPG." IWPG(1) is the group number of the first parameter defined under the first repetition of DATA SET 2; IWPG(2) is the group number of the second parameter defined under a DATA SET 2, and so on.

Positive group numbers: Residuals related to prior information listed in DATA SET 8 are printed by group. Group numbers need to be less than or equal to the number of estimated parameters.

Negative group numbers: Parameter values are substituted into model arrays but are not estimated by regression.

All positive group numbers need to precede any negative group numbers.

DATA SETS 10 through 14 are read only if IPAR>0 (LINE 6).

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION
10	1 or more	8F10.0	WP(I), I=1,NP	If WP(I)≠0.0, the Ith initial value of DATA SET 8 is used as the prior estimate of the parameter (eq. 28).

Weights of the prior estimates are calculated for WP(I)≠0.0, as indicated by IWP of DATA SET 11. If IWP=0, the nonzero WP(I) are variances; if IWP=1, they are standard deviations; and if IWP=2, they are coefficients of variation.

If LN(I)≠0 (see DATA SET 2), WP(I) describes the probability distribution function of the natural log of the parameter.

Instructions for the INPUT FILE--Continued

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION
11	1	I10	IWP	Flag to indicate how the WP of DATA SETS 10 and 12 are used to calculate weights.

Weights are calculated as EV/WP(I) for IWT=0, $EV/(WP(I))^2$ for IWT=1, or $EV/(WP(I)\times B(I))^2$ for IWT=2. EV is the estimated common error variance from DATA SET 5; B(I) is the initial value of the Ith parameter from DATA SET 8. When IWT=2, a value of 1.0 is used when B(I)=0.0, and ℓn B(I) replaces B(I) when LN(I)≠0 (DATA SET 2).

DATA SET 12 needs to be repeated MPR times (LINE 5). Omit if MPR=0.

DATA SET	NUMBER OF LINES	FORMAT	VARIABLE	DEFINITION
12	1 or more	8F10.0	PRM(I), I=1,NP, PRE, WP	NP coefficients, the prior estimate, and the value from which the weight of the prior estimate is to be calculated.

See section "Restarting the Parameter-Estimation Routine" for an example of how DATA SET 12 can be used.

PRM(I) is the coefficient of the Ith parameter; if PRM(I)=0.0, the Ith parameter is not included.

WP is used to calculate a weight by using IWP of DATA SET 11 and the equations stated there, with WP replacing WP(I). For IWP=2, PRE replaces B(I).

If ISN>0 (LINE 6), either the SCL(I) (DATA SET 14) related to all PRM(I)≠0 need to be equal, or only one PRM(I)≠0.

If PRM(I)≠0.0 for more than one I, no log-transformed parameters can be included.

For log-transformed parameters, PRE needs to be untransformed, but WP describes the normal probability distribution function of the natural log of the parameter. Thus, the definition of PRE and WP of DATA SET 12 are consistent with the definition of B(I) and WP(I) of DATA SETS 8 and 10, respectively.

Instructions for the INPUT FILE--Continued

DATA SET	COLUMN	FORMAT	VARIABLE	DEFINITION
13	1-5	F5.0	DMAX	Maximum fractional change for parameter values in one iteration. Used to calculate ρ_r of equation (50b).
		If DMAX<1.0, parameters cannot change sign (see discussion after eq. 54).		
	6-10	F5.0	CSA	Search direction adjustment parameter used in Marquardt procedure.
		CSA=0.08 is the value suggested by Cooley and Naff (1990, p. 71-72, 91). Used only in modified Gauss-Newton optimization.		
	11-15	F5.0	TOL	Parameter-estimation closure criterion. See "TOL." Commonly equals 0.01 or 0.001.
	16-20	I5	ITMXP	Maximum number of parameter-estimation iterations. Commonly equal to 1.5 times the number of estimated parameters for modified Gauss-Newton optimization.
	21-25	F5.0	FCONV	If FCONV>0, coarser solver convergence criteria are used when calculating hydraulic heads and sensitivities for the earlier parameter-estimation iterations.
		Commonly, FCONV=0.0. Typical nonzero values of FCONV would be 5.0 or 10.0. See the section, "Reducing Computer Execution Time of Modified Gauss-Newton Optimization" for additional information.		
	26-30	I5	IPRIOR	If IPRIOR=0, prior information from DATA SETS 10 and 12 is disregarded.
	30-35	I5	IOUR	Output unit for unformatted data to be used by RESAN.MODP, a residual analysis program. If IOUR=0, nothing is output.
	36-40	I5	IPRC	Format code with which the variance-covariance and correlation matrices are printed when parameter estimation converges. Options are listed in table A2; IPRC=2 usually produces a satisfactory printout.

DATA SET	COLUMN	FORMAT	VARIABLE	DEFINITION
13	41-45	I5	IPRINT	Controls printing of selected items at each iteration.

The controlled items are: for ISN<0, the scaled least-squares matrix (eq. 50a), observation data and sensitivities, and prior-estimate data; for ISN>0, the observation and prior-estimate data. Runs-test statistics (see "Analysis of Results for Linear Problems") are printed with the observation and prior-estimate data.

If IPRINT>0, the items are printed at each iteration.

If IPRINT=0, the observation-point and prior-estimate data and, for ISN<0, sensitivities, are printed at the first and last iterations.

DATA SET	COLUMN	FORMAT	VARIABLE	DEFINITION
	46-50	I5	KPRINT	If KPRINT=1, hydraulic-head arrays for the parameter values for which parameter estimation converges are printed and saved, as specified in the Output Control input file. Otherwise, set KPRINT=0.
	51-55	I5	LPRINT	If LPRINT>0, the eigenvalues and eigenvectors of the variance-covariance matrix scaled with the final parameters are printed upon convergence of parameter estimation.

Used only when the covariance matrix is calculated (on LINE 6, ISN<0 or, in DATA SET 13, LASTX>0). LPRINT may equal 0, 1, 2, or 3; see section "LPRINT."

DATA SET	COLUMN	FORMAT	VARIABLE	DEFINITION
	56-60	I5	LASTX	Controls calculation of sensitivity-equation sensitivities used to calculate the variance-covariance matrix on the parameters (eq. 35) when parameter estimation converges.

When ISN<0: If LASTX=0, sensitivities from the last parameter-estimation iteration are used. If LASTX>0, sensitivities calculated by using the final parameter estimates are used. LASTX=0 saves computer time and usually produces a satisfactory variance-covariance matrix.

When ISN>0: If LASTX=0, the variance-covariance matrix is not calculated. If LASTX>0, the variance-covariance matrix (eq. 35) and the correlation matrix are calculated and printed, and -LASTX is used in the same way as ISN when ISN<0 to define the method used to calculate the sensitivities (see ISN on LINE 6).

Instructions for the INPUT FILE--Continued

DATA SET	COLUMN	FORMAT	VARIABLE	DEFINITION
13	61-65	I5	NFIT	Number of Fletcher-Reeves (ISN=2) or Gauss-Newton (ISN<0; NOPT=1) iterations. If ISN=2, NFIT needs to be equal to or greater than ITMXP. (ISN and NOPT are on LINE 6). See section "Parameter Estimation and Analysis of Results using Nonlinear Regression."
	66-70	F5.0	SOSC	The second convergence criterion discussed in the section "Convergence Criteria." Parameter estimation using the modified Gauss-Newton method converges if the percentage change in the sum of squared, weighted residuals does not exceed SOSC×100 in three parameter-estimation iterations.
	71-75	F5.0	SOSR	A criteria for using $\underline{\underline{R}}$ of equation (51) in equation (50a). $\underline{\underline{\bar{R}}}$ is used if the percentage change in the sum of squared, weighted residuals does not exceed SOSR×100 in two parameter-estimation iterations.
DATA SET 14 is read only if ISN>0.				
14	1 or more	8F10.0	SCL (I), I=1,NP	Factors used to scale parameters for conjugate-direction methods.
	Generally, SCL(I) ≅ B(I) of DATA SET 8. If LN(I)>0 (DATA SET 2), the Parameter-Estimation Package replaces SCL(I) with ℓn [SCL(I)], and the transformed value is printed.			

Table A2.--*Format code (IPRC in DATA SET 13 of the INPUT FILE) options for printing the variance-covariance and correlation matrices of the parameters*

IPRC	FORMAT	IPRC	FORMAT
1	11G10.3	7	20F5.0
2	9G13.6	8	20F5.1
3	15F7.1	9	20F5.2
4	15F7.2	10	20F5.3
5	15F7.3	11	20F5.4
6	15F7.4	12	10G11.4

Description of Output

Printing and Checking of the INPUT FILE

The input file is read and then printed on output unit IOUT (IOUT is specified as 6 in the beginning of MAIN, but can be changed if needed). If ISN<0, covergence criteria are calculated for the solution of equations (69) or (71), as described in the section "Sensitivity-Equation Method of Calculating Sensitivities" after equation (72), and are printed with the initial parameter values.

Error Messages

The Parameter-Estimation Package prints error messages to output unit IOUT. These messages indicate errors in the INPUT FILE and problems in program execution, and are followed by the name of the module or submodule (see Appendix C) from which they are printed in parentheses.

Most INPUT FILE errors will stop execution of the program. "FORMAT/DATA MISMATCH" or "END-OF-FILE" errors originate from the computer, and execution stops as soon as the error is detected. A few errors detected by checks within the Parameter-Estimation Package also will stop execution as soon as the error is detected, but for most package-detected errors, the entire INPUT FILE is read and printed before execution stops. Then "EXECUTION TERMINATED DUE TO ERRORS IN PARAMETER-ESTIMATION PACKAGE INPUT FILE--SEARCH ABOVE FOR STOP EXECUTION TO FIND ERROR MESSAGES (SEN1OT)" is printed at the end of the output file. Error messages can be located by searching the output file for the words "STOP EXECUTION."

Once the INPUT FILE has been read successfully, execution problems can occur when, for example, a negative hydraulic-conductivity value is calculated, thus activating the option of the Parameter-Estimation Package which writes an error message to IOUT and resets the value to a small, positive number. Most execution problems will not stop the program, but their presence generally indicates that the parameter-estimation problem being considered is not well-conditioned. Check scaled sensitivity-equation sensitivities or the gradient of the objective function scaled by dividing through by the relevant parameter values for parameters that are insensitive to the observed data.

Most error messages are self-explanatory and are not discussed here. However, the following need additional explanation. Underlined items are variables which are specified in the INPUT FILE.

Error message:

HEAD OBS# 4321, ID OAK5 IS DRY -- OMIT (SSEN1U)

Explanation:

The observation is omitted because of conditions represented in figure 11, and the number of observations used to calculate the estimated error variance (ND of eq. 7 and in the text after eq. 17 and 35) is reduced by one. The observation is restored and ND is reset to its original value at the beginning of the next parameter-estimation iteration. SSEN1U is the submodule from which the error message is printed (see Appendix C).

Error Message:

INTERPOLATION FOR HEAD OBS# 4321, ID OAK5 CHANGED BECAUSE AT LEAST ONE NEIGHBORING CELL IS DRY (SSEN1U)

Interpolation is recalculated because at least one of the neighboring cells used in the interpolation (see figs. 6 and 11) is dry. Although the row, column, and layer of the dry cell are not specified in the error message, dry cells are listed elsewhere in the output from a MODFLOW submodule. At the beginning of the next parameter-estimation iteration, the inactive dry cell is reactivated, and the interpolation is reset to its original form.

Model-Input Arrays Produced with the Initial Parameter Estimates

For PID = T, KV, TKV, S1, RCH, and ETM, the model-input arrays produced with the initial parameter estimates (DATA SET 8) are printed if IPRNP>0 (LINE 4 of the INPUT FILE) and the associated NLL value(s) (DATA SET 2) are positive. To check that the initial model-input arrays produced by MODFLOWP with the Parameter-Estimation Package (IUNIT (15)≠0) are the same as those read from the Block-Centered Flow, Recharge, and Evapotranspiration input files for a run without the Parameter Estimation Package (IUNIT (15)≠0), set IPRNP equal to IPRN used in those input files. The arrays printed from the two runs need to be identical to four or five significant digits.

Calculated Values and Volumetric Budgets for the Entire Model

If IPAR=0 on LINE 6 of the INPUT FILE equals zero, the following arrays are calculated using the parameter values of DATA SET 8, and can be printed, saved on an unformatted file, or both by using the specifications from the Output Control input file (McDonald and Harbaugh, 1988, p. 4-14 to 4-7): hydraulic heads and drawdowns, and, for ISN<0 (LINE 6 of the INPUT FILE), sensitivity-equation sensitivities, or for ISN>0, adjoint states and the derivative of hydraulic head with respect to Newton's iteration parameter. Arrays are printed in the order shown in figure A6. For IPAR=0, the flags of the Ouput Control input file need to be set such that drawdowns are neither printed nor saved.

[print/save time step : a time step at which Hdpr and Hdsv of Output Control (McDonald and Harbaugh, 1988, p. 4-14) indicate that results are to be printed and (or) saved and for which there is at least one dependent-variable observation. NPER : the total number of time steps.]

ISN=-1	ISN=-2	ISN>0
First print/save time step: Hydraulic heads Sensitivity-equation sensitivities for all parameters	Hydraulic heads for all print/save time steps	Hydraulic heads for all print/save time steps
Repeat for all print/ save time steps	First parameter sensitivity-equation sensitivities for all print/save time steps	Adjoint states for NPER minus the print/save time step numbers
	Repeat for all parameters	

Figure A6.--Order of arrays printed, saved, or both when IPAR of the INPUT FILE equals zero.

Volumetric budgets of the entire model (McDonald and Harbaugh, 1988, p. 4-7) are printed with solutions of hydraulic heads, if indicated by the Output Control input file. When PCG2 (Hill, 1990b) is used with MXITER>1, simplified volumetric budgets of the entire model are printed with solutions of sensitivity-equation sensitivities and adjoint states. Only the budgets related to hydraulic heads make obvious physical sense, but the others are included to check solver accuracy. Excessive errors (generally larger than 1 percent) indicate that the solution is inaccurate. Smaller convergence criteria, smaller time steps, or a smaller grid spacing might be needed to resolve the problem. Cumulative budget terms printed for solutions of steady-state hydraulic head and drawdown are very large because the time-step length is set equal to 1.0×10^{22} for steady-state simulations.

Plotting Calculated Values

Routines such as those by Leake (1990; U.S. Geological Survey, written commun., 1989) and Harbaugh (1990) have been developed to contour calculated hydraulic heads and drawdowns (Leake, 1990, also plots fluxes) from unformatted files produced by MODFLOW (McDonald and Harbaugh, 1988, p. E-3). In general, these routines or appropriate preprocessors identify the array to be contoured using values of TEXT, ILAY, KSTP, and KPER, which are written just before the array by ULASAV of MODFLOW (McDonald and Harbaugh, 1988, p. 14-9). TEXT is a label (HEAD or DRAWDOWN); ILAY is the layer number; and KSTP and KPER are the time step and stress period of the data, with KSTP measured from the beginning of the stress period.

The contouring routines also can be used to contour calculated hydraulic heads, drawdowns, and sensitivity-equation sensitivities (ISN<0) or adjoint states (ISN<0) produced by the Parameter-Estimation Package (MODFLOWP with IUNIT(15)≠0) with IPAR=0 (heads and drawdowns also can be saved when IPAR<0, and when IPAR>0 if KPRINT=1 in DATA SET 13). The only difference is that when using the Parameter-Estimation Pachage, KSTP is measured from the beginning of the simulation, with KSTP=0 indicating steady state, and KPER is used as noted on table A3 to indicate what the values represent. For sensitivity-equation sensitivities and adjoint states, TEXT=HEAD.

Table A3.--*The use of KSTP and KPER in unformatted files produced by the computer programs MODFLOW and MODFLOWP*

[--, plotted quantity indicated not calculated when IUNIT(15)=0]

Plotted quantity	MODFLOW or MODFLOWP with IUNIT(15)=0		MODFLOWP with IUNIT(15)≠0	
	KSTP	KPER	KSTP	KPER
Hydraulic heads Drawdowns Fluxes	Time step since beginning of stress period.	stress period	Time step since beginning of simulation. KSTP=0 for steady state.	0
Sensitivity-equation sensitivities.	--	--	.....do..............	Parameter number
Adjoint states	--	--	.....do..............	1

Nonlinear Regression

Nonlinear regression is performed when IPAR of LINE 6 of the INPUT FILE is greater than zero. At each parameter-estimation iteration, items are printed on output unit IOUT to permit the modeler to monitor the progress of the optimization. For modified Gauss-Newton optimization, these items are the determinant of the scaled least-squares matrix (small values indicate a poorly conditioned parameter-estimation problem), the Marquardt parameter (labeled AMP on the printout; nonzero values indicate a poorly conditioned parameter-estimation problem), the factor used to scale the parameter-change vector (ρ_r of eq. 50b; labeled AP on the printout), the maximum fractional parameter change calculated using equation (50a) (labeled DMX on the printout; modified Gauss-Newton optimization converges when the absolute value of DMX is less than TOL of DATA SET 13 of the INPUT FILE), and the number of the parameter for which the maximum fractional change occurred (j of eq. 54).

For conjugate-direction optimization, these items are the iteration parameter calculated by Newton's method, the factor used to reduce the iteration parameter (labeled AP on the printout; discussed after eq. 56), the maximum fractional parameter change prior to reduction of the iteration parameter (labeled DMX on the prinout), the parameter for which the maximum fractional change occurred, and the absolute value of the component of the gradient vector that has the largest magnitude. Conjugate-direction optimization converges when the last item and the absolute value of the maximum fractional parameter change are less than TOL of DATA SET 13 of the INPUT FILE.

For both optimization methods, the updated parameter estimates are printed at each parameter-estimation iteration and in a table at the end of the simulation (see section "Parameter Values for Restarting Parameter Estimation"). The table is printed whether or not parameter estimation converges and includes some of the items mentioned above and the sum of squared residuals for each iteration. Printing the values together in a table is intended to make it easier for the user to determine how individual parameter values are changing, and whether the changes indicate a problem. If problems occur, see the section "Common Problems." If LN(I)≠0, the value printed for parameter I equals e^b, where b is the log-transformed value estimated by the parameter-estimation routine, and e^b equals the mode of the lognormal distribution (see section "Parameterization").

The printing of other items at each parameter-estimation iteration is controlled by IPRINT of DATA SET 13 of the INPUT FILE; see the section "Instructions for the INPUT FILE" for additional information.

The following additional items are printed toward the end of the output file if parameter estimation converges.

(1) For all optimization methods, tables of information related to the dependent-variable observations and prior parameter estimates are printed, in addition to the final sum of squared weighted residuals, the calculated error variance (defined after eq. 17 and 35), R (eq. 37), the total number of parameter-estimation iterations, and R_N^2 (eq. 38). The latter three are printed excluding and including prior estimates of the parameters.

(2) If optimization is accomplished by the modified Gauss-Newton method or if LASTX>0 in DATA SET 13 of the INPUT FILE, the variance-covariance matrix (eq. 35) of the parameters and related statistics are printed, and a table containing the following is printed: The estimated parameter values, the exponential of transformed parameters (zeros are printed for parameters that are not log-transformed), the standard derivations and coefficients of variation of the estimates, and the estimates plus and minus two standard derivations. Equation (36) needs to be used to calculate accurate linear confidence intervals on the parameters, but the estimated values plus or minus two standard deviations provide the user with an estimate of a linear 95-percent confidence interval. Possible inaccuracies in linear confidence intervals in the parameters were mentioned in the section "Analysis of Results for Nonlinear Problems." Eigenvalues and eigenvectors of the variance-covariance matrix scaled with the final parameter values are calculated and printed depending on the value of LPRINT of DATA SET 13.

(3) If the quasi-Newton method is used and LASTX=0, the final approximate quasi-Newton inverse Hessian matrix is printed.

(4) Nodal hydraulic heads are printed, saved, or both, depending on the value of KPRINT of DATA SET 13 of the INPUT FILE.

Parameter Values for Restarting Parameter Estimation

Parameter values from each parameter-estimation iteration are printed on the FORTRAN unit number specified by IOUB of LINE 7 of the INPUT FILE. These parameter values are printed in a format consistent with DATA SET 8 of the INPUT FILE, so they can be used to restart the parameter-estimation routine, as discussed in the section "Restarting the Parameter-Estimation Routine" of Appendix B. File IOUB also is used to print the table of parameter values and other items for each parameter-estimation iteration discussed above.

Input for the Residual Analysis Program

A file that is to be used as input for RESAN.MODP, a slightly modified version of Cooley and Naff's (1990) residual analysis program (see Appendix B), is produced on the FORTRAN unit number specified by IOUR of DATA SET 13 of the INPUT FILE.

Sample INPUT FILES and Output

Sample INPUT FILES and output for two test cases are presented. The two test cases were presented previously by Cooley (1985), Hill (1989), and Cooley and Naff (1990).

Test Case 1

The physical system for test case 1 is shown in figure A7. The system is comprised of two confined aquifers separated by a confining unit. The system is affected by head-dependent boundaries at the stream and along the boundary farthest from the stream. The aquifer properties shown in figure A7 are the storage coefficient and transmissivity of aquifer 1 (S_1 and T_1), the vertical hydraulic conductivity of the confining unit (K_V), the storage coefficient and a multiplicative factor for transmissivity of aquifer 2 (S_2 and K_{2M}), and the hydraulic conductivity of the material bounding the aquifers on the side of the physical system farthest from the stream (K_B). Storage in the confining unit is assumed to be negligible. Stresses on the system include areal recharge in the area near the stream (RCH1), areal recharge in the area farther from the stream (RCH2), and pumpage from each of the two layers (Q_1 and Q_2). The two pumpage rates are equal, so Q_1=Q_2. The hydraulic conductivity of the riverbed is K_{RB}.

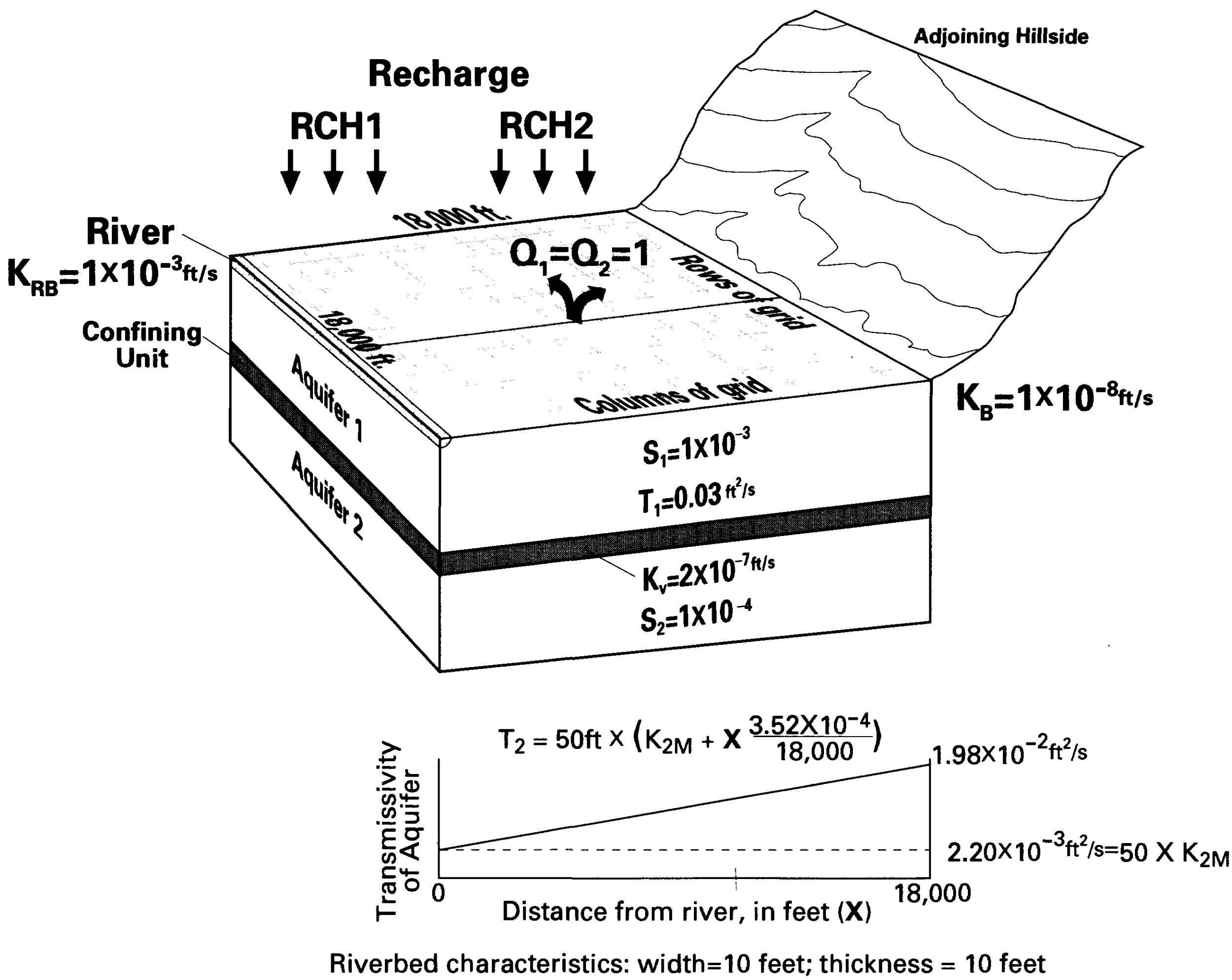


Figure A7.--Physical system for test case 1, with the true values of all model inputs except RCH1 and RCH2.

For the finite-difference method, the system is discretized into square 1,000-ft by 1,000-ft cells, so that the grid has 18 rows and 18 columns. The river is in column 1. RCH1 applies to cells in columns 1 through 9; RCH2 to cells in columns 10 through 18. The wells are located at the center of the cell at row 9, column 10. The simulation begins from steady-state initial conditions with no pumpage, and a constant rate of pumpage is maintained throughout the simulation. Four stress periods are simulated: the first three are 1, 3, and 6 days long, and each has one time step; the fourth is 272.8 days long, has 9 time steps, and each time-step length is 1.2 times the length of the previous time-step length.

The following nine parameters are estimated: Q_1 and Q_2, K_1 ($T_1 = K_1 \times 50$ ft), S_1, K_{RB}, K_z ($K_v = K_z/10$ ft), K_{2M} (see graph in figure A7 for calculation of T_2 from K_{2M}), S_2, RCH1, and RCH2. Parameters K_1, S_1, K_{RB}, K_z, and K_{2M} and S_2 are log-transformed. The true values of most of these parameters are shown in figure A7; the true values of RCH1 and RCH2 are 1.0×10^{-8} ft/s and 1.5×10^{-8} ft/s (3.78 and 5.76 in/yr), respectively. The true values are known because the test case is synthetic. Hydraulic-head data were produced by using the true parameter values, and then were corrupted by adding two realizations of normally distributed noise. The first realization had a standard deviation of 1.0 ft, and the same value was used to corrupt all hydraulic-head observations at a single location; thus ϵ_1 errors were imposed. The second realization had a standard deviation of 0.05 ft, and independent values were used to corrupt all hydraulic-head observations; thus ϵ_3 errors were superimposed. In repetitions of DATA SET 6B, IIT=2, so that initial hydraulic heads and subsequent changes in hydraulic head were used in the regression. Head-dependent boundary gain and loss data also were produced by using the true parameter values, and were corrupted by adding normally distributed noise with a standard deviation equal to 0.10 times the true gain along the stream.

INPUT FILE and Input Files from other Packages for Test Case 1

The INPUT FILE used for test case 1 is shown on the following pages. Note that although the presence of nonzero values in DATA SET 10 indicates the use of prior estimates of the parameters, the designation of IPRIOR=0 in DATA SET 13 means that these prior estimates are not used.

INPUT FILE

```
TWO-LAYER EXAMPLE - TRANSIENT                                     LINE 1
                                                                  LINE 2
    9    5   56    4    2    1    1                               LINE 3
    2    1    1                                                   LINE 4
   32    0    0    1   18    3    0                               LINE 5
    1   -1    1    0                                              LINE 6
   40   33    0    0                                              LINE 7
    1    0    0                                                   LINE 8
   30                                                             DATA SET 1A
Q   0   -2    1   15   -1                                         DATA SET 2
         1         9        10        1.0                         DATA SET 2A
         2         9        10        1.0
S1  1   -1                                                        DATA SET 2
T   1    1                                                        DATA SET 2
       50.                                                        DATA SET 2B
```

```
KST 1   18    0   15   -1                                              DATA SET 2
    1    1    1    1    1         10.0       100      1000.        90        95DATA
    1    2    1    1    2          0.0       100      1000.        90        95 SET 2A
    1    3    1    1    3          0.0       100      1000.        90        95
    1    4    1    1    4          0.0       100      1000.        90        95
    1    5    1    1    5          0.0       100      1000.        90        95
    1    6    1    1    6          0.0       100      1000.        90        95
    1    7    1    1    7          0.0       100      1000.        90        95
    1    8    1    1    8          0.0       100      1000.        90        95
    1    9    1    1    9          0.0       100      1000.        90        95
    1   10    1    1   10          0.0       100      1000.        90        95
    1   11    1    1   11          0.0       100      1000.        90        95
    1   12    1    1   12          0.0       100      1000.        90        95
    1   13    1    1   13          0.0       100      1000.        90        95
    1   14    1    1   14          0.0       100      1000.        90        95
    1   15    1    1   15          0.0       100      1000.        90        95
    1   16    1    1   16          0.0       100      1000.        90        95
    1   17    1    1   17          0.0       100      1000.        90        95
    1   18    1    1   18          0.0       100      1000.        90        95
KV  1    1                                                             DATA SET 2
        .1                                                             DATA SET 2B
S1  1   -2                                                             DATA SET 2
T   1    2                                                             DATA SET 2
       50.    1                                                        DATA SET 2B
RCH 0    1    0   15   -1                                              DATA SET 2
        1.    0    1    1                                              DATA SET 2B
RCH 0    1    0   15   -1                                              DATA SET 2
        1.    0    1    2                                              DATA SET 2B
        16         1.(18F3.0)                                          DATA SET 3
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
```

16 1(9I8) DATA SET 4

1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	2
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	3
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	4
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	5
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	6
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	7
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	8
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	9
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	2
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	11
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	12
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	13
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	14
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	15
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	16
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	17
2	2	2	2	2	2	2	2	2	
1	1	1	1	1	1	1	1	1	18
2	2	2	2	2	2	2	2	2	

```
        1.0       1.0    0    0       1.0                                 DATA SET 5
1.0       1    3    1   -3   0.00    0.00    0.00     0.000    0.00    0 DATA SET 6
    2                                                                 DATA SET 6B
1.0       0    0.00   101.804    1.0025    0.0025    0                DATA SET 6C
1.1       1    0.00   101.775    1.0025    0.0025    0
1.12     12    0.00   101.675    1.0025    0.0025    0
2.0       1    4    4   -5   0.00    0.00    0.00     0.000    0.00    0 DATA SET 6
    2                                                                 DATA SET 6B
2.0       0    0.00   128.117    1.0025    0.0025    0                DATA SET 6C
2.1       1    0.00   128.076    1.0025    0.0025    0
2.2       2    0.00   127.560    1.0025    0.0025    0
2.8       8    0.00   116.586    1.0025    0.0025    0
2.12     12    0.00   113.933    1.0025    0.0025    0
3.0       1   10    9   -3   0.00    0.00    0.00     0.000    0.00    0 DATA SET 6
    2                                                                 DATA SET 6B
3.0       0    0.00   156.678    1.0025    0.0025    0                DATA SET 6C
3.1       1    0.00   152.297    1.0025    0.0025    0
3.12     12    0.00   114.138    1.0025    0.0025    0
4.0       1   13    4   -3   0.00    0.00    0.00     0.000    0.00    0 DATA SET 6
    2                                                                 DATA SET 6B
4.0       0    0.00   124.893    1.0025    0.0025    0                DATA SET 6C
4.1       1    0.00   124.826    1.0025    0.0025    0
4.12     12    0.00   110.589    1.0025    0.0025    0
5.0       1   14    6   -3   0.00    0.00    0.00     0.000    0.00    0 DATA SET 6
    2                                                                 DATA SET 6B
5.0       0    0.00   140.961    1.0025    0.0025    0                DATA SET 6C
5.1       1    0.00   140.901    1.0025    0.0025    0
5.12     12    0.00   119.285    1.0025    0.0025    0
6.0       2    4    4   -3   0.00    0.00    0.00     0.000    0.00    0 DATA SET 6
    2                                                                 DATA SET 6B
6.0       0    0.00   126.537    1.0025    0.0025    0                DATA SET 6C
6.1       1    0.00   126.542    1.0025    0.0025    0
6.12     12    0.00   112.172    1.0025    0.0025    0
7.0       2   10    1   -3   0.00    0.00    0.00     0.000    0.00    0 DATA SET 6
    2                                                                 DATA SET 6B
7.0       0    0.00   101.112    1.0025    0.0025    0                DATA SET 6C
7.1       1    0.00   101.160    1.0025    0.0025    0
7.12     12    0.00   100.544    1.0025    0.0025    0
8.0       2   10    9   -3   0.00    0.00    0.00     0.000    0.00    0 DATA SET 6
    2                                                                 DATA SET 6B
8.0       0    0.00   158.135    1.0025    0.0025    0                DATA SET 6C
8.1       1    0.00   152.602    1.0025    0.0025    0
8.12     12    0.00   114.918    1.0025    0.0025    0
9.0       2   10   18   -3   0.00    0.00    0.00     0.000    0.00    0 DATA SET 6
    2                                                                 DATA SET 6B
9.0       0    0.00   176.374    1.0025    0.0025    0                DATA SET 6C
9.1       1    0.00   176.373    1.0025    0.0025    0
9.12     12    0.00   138.132    1.0025    0.0025    0
0.0       2   18    6   -3   0.00    0.00    0.00     0.000    0.00    0 DATA SET 6
    2                                                                 DATA SET 6B
0.0       0    0.00   142.020    1.0025    0.0025    0                DATA SET 6C
0.1       1    0.00   142.007    1.0025    0.0025    0
0.12     12    0.00   122.099    1.0025    0.0025    0
```

```
   3    3  -18                                                           DATA SET 7
SS      0     0.0       -4.4    .400000    1                             DATA SET 7B
TR3     3     0.0       -4.1    .380000    1
TR12   12     0.0       -2.2    .210000    1
   1    1    1    1    1          10.0         100      1000.        90          95 DATA
   1    2    1    1    2           0.0         100      1000.        90          95  SET
   1    3    1    1    3           0.0         100      1000.        90          95   7C
   1    4    1    1    4           0.0         100      1000.        90          95
   1    5    1    1    5           0.0         100      1000.        90          95
   1    6    1    1    6           0.0         100      1000.        90          95
   1    7    1    1    7           0.0         100      1000.        90          95
   1    8    1    1    8           0.0         100      1000.        90          95
   1    9    1    1    9           0.0         100      1000.        90          95
   1   10    1    1   10           0.0         100      1000.        90          95
   1   11    1    1   11           0.0         100      1000.        90          95
   1   12    1    1   12           0.0         100      1000.        90          95
   1   13    1    1   13           0.0         100      1000.        90          95
   1   14    1    1   14           0.0         100      1000.        90          95
   1   15    1    1   15           0.0         100      1000.        90          95
   1   16    1    1   16           0.0         100      1000.        90          95
   1   17    1    1   17           0.0         100      1000.        90          95
   1   18    1    1   18           0.0         100      1000.        90          95
     -1.1    1.3E-3   3.00E-4    1.2E-3      1.E-7      2.E-4     4.0E-5     2.E-8
   1.0E-8                                                            DATA SET 8
        2         3         4         5          6          3         4          7
        7                                                            DATA SET 9
       .1    1.0E-4   4.00E-5    1.0E-4         0.         0.        0.     1.E-9
   1.5E-9                                                            DATA SET 10
        1                                                            DATA SET 11
  2.0  .08 .010   10   0.    0    0   12    0    0    0    0    5   0.    0. DATA SET
 DMAX  CSA  TOL ITMX FCONV      IOUR     IPRINT     LPRINT     NFIT     SOSR   12
                          IPRIOR     IPRC       KPRINT     LASTX     SOSC
```

Input files from other packages used without (labeled "Steady State" and "Transient") and with the Parameter-Estimation Package are shown on the following pages. The true model input values are included in these files.

Basic Package Input Files

Without the Parameter-Estimation Package

Steady-State

```
MODULAR MODEL - TWO-LAYER EXAMPLE PROBLEM

         2         18         18          1          1
  7  8  0  0  0  0 12 31  0  0  0 11  9  0  0  0  0 13
         0          1
         0          1                             -1
         0          1                             -1
        0.
         0       200.                             -1
         0       200.                             -1
    86400.          1          1
```

Transient

```
MODULAR MODEL - TWO-LAYER EXAMPLE PROBLEM

         2         18         18          4          1
  7  8  0  0  0  0 12 31  0  0  0 11  9  0  0  0  0 13
         0          1
         0          1                             -1
         0          1                             -1
        0.
       -13         1.                             -1
       -13         1.                             -1
    87162.          1        1.2
   261486.          1        1.2
   522972.          1        1.2
 23567441.          9        1.2
```

With the Parameter-Estimation Package

```
MODULAR MODEL - TWO-LAYER EXAMPLE PROBLEM

         2         18         18         12          1
  7  8  0  0  0  0 12 31  0  0  0 11  9  0 16  0  0 13
         0          1
         0          1                             -1
         0          1                             -1
        0.
         0       200.                             -1
         0       200.                             -1
    87162.          1        1.2
   261486.          1        1.2
   522972.          1        1.2
 23567441.          9        1.2
```

Output Control Package Input Files

Steady-State

2	0	19	00
0	1	1	0
0	0	0	0

Transient

2	0	19	00
0	1	1	0
0	0	0	0
-1	1	1	0
0	1	1	0
0	0	0	0
-1	1	1	0
-1	1	1	0
0	1	1	0
0	0	0	0
-1	1	0	0
-1	1	0	0
0	1	0	0
0	0	0	0
-1	1	0	0
-1	1	0	0
0	1	0	0
0	0	0	0

With the Parameter-Estimation Package

2	0	19	00
0	1	1	0
0	0	0	0
0	1	1	0
0	0	0	0
-1	1	1	0
0	1	1	0
0	0	0	0
-1	1	1	0
-1	1	1	0
0	1	1	0
0	0	0	0
-1	1	0	0
-1	1	0	0
0	1	0	0
0	0	0	0
-1	1	0	0
-1	1	0	0
0	1	0	0
0	0	0	0

Block-Centered Flow Package Input Files

Steady-State

```
         1         0
0 0
         0        1.
         0     1000.
         0     1000.
         0    .02000
         0  2.00E-08
        18   .002200(18F3.0)
```

Transient and with the Parameter-Estimation Package

```
         0         0
0 0
         0        1.
         0     1000.
         0     1000.
         0    .00100
         0    .02000
         0  2.00E-08
         0   .000100
        18   .002200(18F3.0)
```

File 18 used by the Block-Centered Flow Package

```
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
  1  1  2  2  3  3  4  4  5  5  6  6  7  7  8  8  9  9
```

Recharge Package Input File

Steady-State

```
         1         0
         0         0
        30     1.E-8(9F8.0)                      0
```

Transient

```
         1         0
         0         0
        30     1.E-8(9F8.0)                         0
-1
-1
-1
```

With the Parameter-Estimation Package

```
         1         0
         0         0
        30     1.E-8(9F8.0)                         0
-1
-1
-1
-1
```

File 30 used by the Recharge Package

```
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0     1.0
     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5     1.5
```

Well Package Input File

Transient

```
         2         0
         2
         1         9         10        -1.0
         2         9         10        -1.0
-1
-1
-1
```

With the Parameter-Estimation Package

```
         2         0
         2
         1         9         10         0.0
         2         9         10         0.0
         2
         1         9         10        -1.0
         2         9         10        -1.0
-1
-1
-1
```

General-Head Boundary Package Input File

Steady-state

```
        36         0
        36
         1         1         18        350.   1.0E-07
         1         2         18        350.   1.0E-07
         1         3         18        350.   1.0E-07
         1         4         18        350.   1.0E-07
         1         5         18        350.   1.0E-07
         1         6         18        350.   1.0E-07
         1         7         18        350.   1.0E-07
         1         8         18        350.   1.0E-07
         1         9         18        350.   1.0E-07
         1        10         18        350.   1.0E-07
         1        11         18        350.   1.0E-07
         1        12         18        350.   1.0E-07
         1        13         18        350.   1.0E-07
         1        14         18        350.   1.0E-07
         1        15         18        350.   1.0E-07
         1        16         18        350.   1.0E-07
         1        17         18        350.   1.0E-07
         1        18         18        350.   1.0E-07
         2         1         18        350.   1.0E-07
         2         2         18        350.   1.0E-07
         2         3         18        350.   1.0E-07
         2         4         18        350.   1.0E-07
         2         5         18        350.   1.0E-07
         2         6         18        350.   1.0E-07
```

2	7	18	350.	1.0E-07
2	8	18	350.	1.0E-07
2	9	18	350.	1.0E-07
2	10	18	350.	1.0E-07
2	11	18	350.	1.0E-07
2	12	18	350.	1.0E-07
2	13	18	350.	1.0E-07
2	14	18	350.	1.0E-07
2	15	18	350.	1.0E-07
2	16	18	350.	1.0E-07
2	17	18	350.	1.0E-07
2	18	18	350.	1.0E-07

Transient

36	0			
36				
1	1	18	350.	1.0E-07
1	2	18	350.	1.0E-07
1	3	18	350.	1.0E-07
1	4	18	350.	1.0E-07
1	5	18	350.	1.0E-07
1	6	18	350.	1.0E-07
1	7	18	350.	1.0E-07
1	8	18	350.	1.0E-07
1	9	18	350.	1.0E-07
1	10	18	350.	1.0E-07
1	11	18	350.	1.0E-07
1	12	18	350.	1.0E-07
1	13	18	350.	1.0E-07
1	14	18	350.	1.0E-07
1	15	18	350.	1.0E-07
1	16	18	350.	1.0E-07
1	17	18	350.	1.0E-07
1	18	18	350.	1.0E-07
2	1	18	350.	1.0E-07
2	2	18	350.	1.0E-07
2	3	18	350.	1.0E-07
2	4	18	350.	1.0E-07
2	5	18	350.	1.0E-07
2	6	18	350.	1.0E-07
2	7	18	350.	1.0E-07
2	8	18	350.	1.0E-07
2	9	18	350.	1.0E-07
2	10	18	350.	1.0E-07
2	11	18	350.	1.0E-07
2	12	18	350.	1.0E-07
2	13	18	350.	1.0E-07
2	14	18	350.	1.0E-07
2	15	18	350.	1.0E-07
2	16	18	350.	1.0E-07
2	17	18	350.	1.0E-07
2	18	18	350.	1.0E-07
-1				
-1				
-1				

```
36        0
36
 1        1        18       350.    1.0E-07
 1        2        18       350.    1.0E-07
 1        3        18       350.    1.0E-07
 1        4        18       350.    1.0E-07
 1        5        18       350.    1.0E-07
 1        6        18       350.    1.0E-07
 1        7        18       350.    1.0E-07
 1        8        18       350.    1.0E-07
 1        9        18       350.    1.0E-07
 1       10        18       350.    1.0E-07
 1       11        18       350.    1.0E-07
 1       12        18       350.    1.0E-07
 1       13        18       350.    1.0E-07
 1       14        18       350.    1.0E-07
 1       15        18       350.    1.0E-07
 1       16        18       350.    1.0E-07
 1       17        18       350.    1.0E-07
 1       18        18       350.    1.0E-07
 2        1        18       350.    1.0E-07
 2        2        18       350.    1.0E-07
 2        3        18       350.    1.0E-07
 2        4        18       350.    1.0E-07
 2        5        18       350.    1.0E-07
 2        6        18       350.    1.0E-07
 2        7        18       350.    1.0E-07
 2        8        18       350.    1.0E-07
 2        9        18       350.    1.0E-07
 2       10        18       350.    1.0E-07
 2       11        18       350.    1.0E-07
 2       12        18       350.    1.0E-07
 2       13        18       350.    1.0E-07
 2       14        18       350.    1.0E-07
 2       15        18       350.    1.0E-07
 2       16        18       350.    1.0E-07
 2       17        18       350.    1.0E-07
 2       18        18       350.    1.0E-07
-1
-1
-1
-1
```

Streamflow-Routing Package Input File

Steady-state

```
      18         1         0         0         0         0.         0         0
      18         0         0
 1     1    1    1    1             10.0       100      1.00        90        95
 1     2    1    1    2              0.0       100      1.00        90        95
 1     3    1    1    3              0.0       100      1.00        90        95
 1     4    1    1    4              0.0       100      1.00        90        95
 1     5    1    1    5              0.0       100      1.00        90        95
 1     6    1    1    6              0.0       100      1.00        90        95
 1     7    1    1    7              0.0       100      1.00        90        95
 1     8    1    1    8              0.0       100      1.00        90        95
 1     9    1    1    9              0.0       100      1.00        90        95
 1    10    1    1   10              0.0       100      1.00        90        95
 1    11    1    1   11              0.0       100      1.00        90        95
 1    12    1    1   12              0.0       100      1.00        90        95
 1    13    1    1   13              0.0       100      1.00        90        95
 1    14    1    1   14              0.0       100      1.00        90        95
 1    15    1    1   15              0.0       100      1.00        90        95
 1    16    1    1   16              0.0       100      1.00        90        95
 1    17    1    1   17              0.0       100      1.00        90        95
 1    18    1    1   18              0.0       100      1.00        90        95
```

Transient

```
      18         1         0         0         0         0.         0         0
      18         0         0
 1     1    1    1    1             10.0       100      1.00        90        95
 1     2    1    1    2              0.0       100      1.00        90        95
 1     3    1    1    3              0.0       100      1.00        90        95
 1     4    1    1    4              0.0       100      1.00        90        95
 1     5    1    1    5              0.0       100      1.00        90        95
 1     6    1    1    6              0.0       100      1.00        90        95
 1     7    1    1    7              0.0       100      1.00        90        95
 1     8    1    1    8              0.0       100      1.00        90        95
 1     9    1    1    9              0.0       100      1.00        90        95
 1    10    1    1   10              0.0       100      1.00        90        95
 1    11    1    1   11              0.0       100      1.00        90        95
 1    12    1    1   12              0.0       100      1.00        90        95
 1    13    1    1   13              0.0       100      1.00        90        95
 1    14    1    1   14              0.0       100      1.00        90        95
 1    15    1    1   15              0.0       100      1.00        90        95
 1    16    1    1   16              0.0       100      1.00        90        95
 1    17    1    1   17              0.0       100      1.00        90        95
 1    18    1    1   18              0.0       100      1.00        90        95
      -1
      -1
      -1
```

With the Parameter-Estimation Package

```
     18          1          0          0          0          0.          0          0
     18          0          0
 1    1    1     1    1              10.0        100       1.00         90         95
 1    2    1     1    2               0.0        100       1.00         90         95
 1    3    1     1    3               0.0        100       1.00         90         95
 1    4    1     1    4               0.0        100       1.00         90         95
 1    5    1     1    5               0.0        100       1.00         90         95
 1    6    1     1    6               0.0        100       1.00         90         95
 1    7    1     1    7               0.0        100       1.00         90         95
 1    8    1     1    8               0.0        100       1.00         90         95
 1    9    1     1    9               0.0        100       1.00         90         95
 1   10    1     1   10               0.0        100       1.00         90         95
 1   11    1     1   11               0.0        100       1.00         90         95
 1   12    1     1   12               0.0        100       1.00         90         95
 1   13    1     1   13               0.0        100       1.00         90         95
 1   14    1     1   14               0.0        100       1.00         90         95
 1   15    1     1   15               0.0        100       1.00         90         95
 1   16    1     1   16               0.0        100       1.00         90         95
 1   17    1     1   17               0.0        100       1.00         90         95
 1   18    1     1   18               0.0        100       1.00         90         95
     -1
     -1
     -1
     -1
```

PCG2 Package Input File
(not changed when using the Parameter-Estimation Package)

```
      1         50          1
 .00001     .00001         1.          2          0          2
```

Output for test case 1

The ouput for test case 1 is shown in the following pages. The following points are noteworthy.

1. In the regression, observed hydraulic heads are used at steady state, and observed drawdowns are used at subsequent time steps. Larger values are used to weight the drawdowns because the subtraction eliminates the ϵ_1 errors, which are constant over time and larger than the ϵ_3 errors in this test case.

2. Interpretation of the coefficients of variation (see the "PARAMETER SUMMARY" on the output) can be confusing when there are log-transformed parameters. For example, in test case 1, a naive interpretation of the coefficients of variation would indicate that $\ell n\ K_z$ (parameter 5) is estimated more precisely than $\ell n\ K_1$ (parameter 3). However, 95-percent confidence intervals on $\ell n\ K_z$ and $\ell n\ K_1$ calculated using equation (36) and then converted into confidence intervals on K_{2M} and K_1 (the modes of the lognormal distribution) by taking the exponential are $1.70\times10^{-7} < K_{2M} < 2.71\times10^{-7}$, and $3.69\times10^{-4} < K_1 < 4.91\times10^{-4}$. The uncertainty in K_{2m} and K_1 can be compared by considering the difference between the upper and lower confidence limits divided by the exponential of the estimated values, $\hat{K}_z = 2.15\times10^{-7}$ and $\hat{K}_1 = 4.26\times10^{-4}$. The resulting value for K_z is 0.47; the value for K_1 is 0.29. Thus, the coefficients of variation for the log-transformed parameters indicate that $\ell n\ K_z$ is more precisely known, while the confidence limits indicate that K_1 is known more precisely. In general, if the coefficient of variation is the same for two log-transformed parameters, the confidence interval on the exponential of the parameter will be relatively wider, using the criteria described above, for the log-transformed parameter with an estimated value farther from 1.0.

3. The values printed below each column of the scaled sensitivities equal the square root of the sum of the squared scaled sensitivities in each column, divided by the number of values in the column. These values can be used to compare overall sensitivity of all parameters to the dependent-variable observations. For parameters that are not log-transformed, large values generally correspond to small parameter coefficients of

variation (see the 'PARAMETER SUMMARY' on the output), and small values correspond to large parameter coefficients of variation. In test case 1, this generalization can be tested by listing the parameters as in table A4. The generalization is followed by parameters that are not log-transformed. For parameters that are log-transformed, the square root of the sum of squared, scaled sensitivities divided by ND still indicates overall sensitivity to the parameter, but the coefficients of variation may not (see item 2).

4. The calculated error variance (1.39) printed near the bottom of the output is close to the estimated common error variance (1.0) printed above the observed head data, and the final weighted residuals are nearly independently, normally distributed. The latter is indicated by the fact that the correlation coefficient printed at the bottom of the output equals 0.978, which exceeds the critical values of 0.943 and 0.952 of table 3, and the runs statistic printed after the final table of "Data at Head Locations" indicates neither too few nor too many runs. These characteristics indicate that the weights are assigned appropriately, and there is little bias in the model. A more thorough analysis of the residuals, as described in the text, needs to be pursued to verify this conclusion.

Table A4.--*Parameters of test case 1 ranked in order of decreasing (sum of squared, scaled sensitivities)$^{\frac{1}{2}}$/ND and increasing coefficients of variation*

Ordered (sum of squared, scaled sensitivities)$^{\frac{1}{2}}$/ND	Related parameter number and parameter identifier	Related parameter number and parameter identifier	Ordered coefficients of variation
260	3(T)*	7(T)*	0.0075
97	7(T)*	5(KV)*	0.0076
47	1(Q)	3(T)*	0.0092
29	2(S1)*	2(S1)*	0.013
12	5(KV)*	6(S1)*	0.068
3.0	6(S1)*	1(Q)	0.072
2.9	9(RCH)	4(KST)*	0.088
1.1	8(RCH)	9(RCH)	0.098
1.0	4(KST)*	8(RCH)	0.11

*Log-transformed parameters.

```
                    U.S. GEOLOGICAL SURVEY MODULAR FINITE-DIFFERENCE GROUND-WATER MODEL - MODFLOWP VERSION
MODULAR MODEL - TWO-LAYER EXAMPLE PROBLEM
   2 LAYERS         18 ROWS        18 COLUMNS
 12 STRESS PERIOD(S) IN SIMULATION
MODEL TIME UNIT IS SECONDS
I/O UNITS:
ELEMENT OF IUNIT:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
        I/O UNIT:  7  8  0  0  0  0 12 31  0  0  0 11  9  0 16  0  0 13  0  0  0  0  0  0
BAS1 -- BASIC MODEL PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM UNIT  5
ARRAYS RHS AND BUFF WILL SHARE MEMORY.
START HEAD WILL BE SAVED
    6200 ELEMENTS IN X ARRAY ARE USED BY BAS
    6200 ELEMENTS OF X ARRAY USED OUT OF  900000
BCF1 -- BLOCK-CENTERED FLOW PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM UNIT  7
TRANSIENT SIMULATION
     LAYER  AQUIFER TYPE
     ------------------
        1        0
        2        0
     650 ELEMENTS IN X ARRAY ARE USED BY BCF
    6850 ELEMENTS OF X ARRAY USED OUT OF  900000
WEL1 -- WELL PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM  8
MAXIMUM OF    2 WELLS
       8 ELEMENTS IN X ARRAY ARE USED FOR WELLS
    6858 ELEMENTS OF X ARRAY USED OUT OF  900000
RCH1 -- RECHARGE PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM UNIT 31
OPTION 1 -- RECHARGE TO TOP LAYER
     324 ELEMENTS OF X ARRAY USED FOR RECHARGE
    7182 ELEMENTS OF X ARRAY USED OUT OF  900000
GHB1 -- GHB PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM UNIT 12
MAXIMUM OF   36 HEAD-DEPENDENT BOUNDARY NODES
     180 ELEMENTS IN X ARRAY ARE USED FOR HEAD-DEPENDENT BOUNDARIES
    7362 ELEMENTS OF X ARRAY USED OUT OF  900000
STRM -- STREAM PACKAGE, VERSION 1, 10/23/87INPUT READ FROM UNIT 13
MAXIMUM OF   18 STREAM NODES

NUMBER OF STREAM SEGMENTS IS      1

NUMBER OF STREAM TRIBUTARIES IS     0


     290 ELEMENTS IN X ARRAY ARE USED FOR STREAMS
    7652 ELEMENTS OF X ARRAY USED OUT OF 900000
PCG2 -- CONJUGATE GRADIENT SOLUTION PACKAGE, VERSION 2, 5/1/88
MAXIMUM OF    1 CALLS OF SOLUTION ROUTINE
MAXIMUM OF   50 INTERNAL ITERATIONS PER CALL TO SOLUTION ROUTINE
MATRIX PRECONDITIONING TYPE :     1
   2992 ELEMENTS IN X ARRAY ARE USED BY PCG
  10644 ELEMENTS OF X ARRAY USED OUT OF 900000
SEN1 -- SENSITIVITY PACKAGE, VERSION 1, 12/01/91 INPUT READ FROM UNIT 16
TWO-LAYER EXAMPLE - TRANSIENT

NUMBER OF PARAMETERS...............................:    9
NUMBER NLL>0 IN ALL DATA SETS 2 FOR PID=T,KV,
 TKV, AND S1, +1 FOR NLL(1)>0 FOR PID=RCH OR ETM...:    5
SUM ALL |NLL(1)| FOR PID=Q,KRB,KDR,KST,GHB,AND CH..:   56
LARGEST NLL VECTOR IN ALL DATA SETS 2..............:    4
LARGEST LZ VECTOR IN ALL DATA SETS 2B..............:    2
NUMBER OF MULTIPLICATION MATRICES IN DATA SET 3....:    1
NUMBER OF ZONE MATRICES IN DATA SET 4..............:    1

PRINT MATRICES PRODUCED BY INIT. PARAM. EST. (>0)..:    2
CHECK CELL REPETITIONS FOR MATRIX PARAMETERS (>0)..:    1
  NUMBER OF CELL REPETITIONS ALLOWED...............:    1

NUMBER OF HEADS....................................:   32
  NUMBER OF MULTILAYER HEADS.......................:    0
  MAXIMUM NUMBER OF LAYERS FOR MULTILAYER HEADS....:    0
NUMBER OF FLUX CELL GROUPS.........................:    1
  NUMBER OF CELLS IN FLUX CELL GROUPS..............:   18
  NUMBER OF FLUXES.................................:    3
NUMBER OF PRIORS WITH MULTIPLE PARAMETERS..........:    0
```

```
HEADS (<0), HEADS AND SENS. (0), OR PAR. EST. (1)..:    1
ADJOINT (+) OR SEN. EQ. (-) CALCULATIONS...........:   -1
  (GRADIENT SEARCH (+) OR GAUSS-NEWTON (-) REGRESSION)
SCALE PRINTED SENSITIVITIES (>0)...................:    1
ADJUST GAUSS-NEWTON MATRIX WITH NEWTON UPDATES.....:    0

OUTPUT/INPUT UNIT FOR HEADS OR SENSITIVITIES.......:   40
OUTPUT/INPUT UNIT NUMBER FOR INFO FROM ITERATIONS..:   33
INPUT UNIT FOR SS/TR IBOUND ARRAYS.................:    0
INPUT UNIT FOR KNOWN HEADS AT CONSTANT-HEAD BNDRYS.:    0

NUMBER OF INPUT UNITS IDENTIFIED IN RCH AND EVT....:    1
NUMBER OF TIME STEPS>0 AT WHICH HEAD=0.............:    0
NUMBER OF USUALLY POS. PARAMETERS THAT MAY BE NEG..:    0

  4233 ELEMENTS IN X ARRAY ARE USED FOR SENSITIVITIES
 14877 ELEMENTS OF X ARRAY USED OUT OF 900000
PAR1 -- PARAMETER ESTIMATION PACKAGE, VERSION 1, 12/01/91
   369 ELEMENTS IN X ARRAY ARE USED FOR PARAMETER ESTIMATION
 15246 ELEMENTS OF X ARRAY USED OUT OF 900000
MODULAR MODEL - TWO-LAYER EXAMPLE PROBLEM
                                                         BOUNDARY ARRAY =              1 FOR LAYER  1
                                                         BOUNDARY ARRAY =              1 FOR LAYER  2
AQUIFER HEAD WILL BE SET TO 0.00000     AT ALL NO-FLOW NODES (IBOUND=0).
                                                           INITIAL HEAD =  200.0000     FOR LAYER  1
                                                           INITIAL HEAD =  200.0000     FOR LAYER  2
HEAD PRINT FORMAT IS FORMAT NUMBER   2    DRAWDOWN PRINT FORMAT IS FORMAT NUMBER   0
HEADS WILL BE SAVED ON UNIT 19    DRAWDOWNS WILL BE SAVED ON UNIT  0
OUTPUT CONTROL IS SPECIFIED EVERY TIME STEP
                                                COLUMN TO ROW ANISOTROPY =  1.000000
                                                                    DELR =  1000.000
                                                                    DELC =  1000.000
                                                    PRIMARY STORAGE COEF = 0.9999999E-03 FOR LAYER  1
                                                    TRANSMIS. ALONG ROWS = 0.2000000E-01 FOR LAYER  1
                                                VERT HYD COND /THICKNESS = 0.2000000E-07 FOR LAYER  1
                                                    PRIMARY STORAGE COEF = 0.1000000E-03 FOR LAYER  2
                                                    TRANSMIS. ALONG ROWS = 0.2200000E-02 FOR LAYER  2


                                                    SOLUTION BY THE CONJUGATE-GRADIENT METHOD
                                                    ------------------------------------------
                                      MAXIMUM NUMBER OF CALLS TO PCG ROUTINE =        1
                                         MAXIMUM ITERATIONS PER CALL TO PCG =       50
                                                MATRIX PRECONDITIONING TYPE =        1
                      RELAXATION FACTOR (ONLY USED WITH PRECOND. TYPE 1) =   0.10000E+01
                PARAMETER OF POLYMOMIAL PRECOND. = 2 (2) OR IS CALCULATED :       2
                                        HEAD CHANGE CRITERION FOR CLOSURE =   0.10000E-04
                                    RESIDUAL CHANGE CRITERION FOR CLOSURE =   0.10000E-04
                             PCG HEAD AND RESIDUAL CHANGE PRINTOUT INTERVAL =      999
                  PRINTING FROM SOLVER IS LIMITED(1) OR SUPPRESSED (>1) =        2

INPUT UNIT NUMBERS LISTED IN RCH AND EVT PACKAGE INPUT FILES :   30


DEFINITION OF TEMPORAL AND SPATIAL EXTENT OF PARAMETERS (PARAMETERIZATION)

PARAMETER NUMBER:    1, PARAMETER IDENTIFIER:  Q
LN:  0, VECTOR NLL:   -2    1   15   -1
    LAYER       ROW    COLUMN    FACTOR
      1.         9.      10.    0.100E+01
      2.         9.      10.    0.100E+01

PARAMETER NUMBER:    2, PARAMETER IDENTIFIER:  S1
LN:  1, VECTOR NLL:   -1    0    0    0

PARAMETER NUMBER:    3, PARAMETER IDENTIFIER:  T
LN:  1, VECTOR NLL:    1    0    0    0
SFAC=   50.0     , MULT. MATRIX    0, ZONE MATRIX    0
ZONES =    0    0

PARAMETER NUMBER:    4, PARAMETER IDENTIFIER:  KST
LN:  1, VECTOR NLL:   18    0   15   -1
   SEGMENT     REACH    ZERO     FACTOR
      1.         1.       0.    0.100E+04
      1.         2.       0.    0.100E+04
      1.         3.       0.    0.100E+04
      1.         4.       0.    0.100E+04
```

1.	5.	0.	0.100E+04
1.	6.	0.	0.100E+04
1.	7.	0.	0.100E+04
1.	8.	0.	0.100E+04
1.	9.	0.	0.100E+04
1.	10.	0.	0.100E+04
1.	11.	0.	0.100E+04
1.	12.	0.	0.100E+04
1.	13.	0.	0.100E+04
1.	14.	0.	0.100E+04
1.	15.	0.	0.100E+04
1.	16.	0.	0.100E+04
1.	17.	0.	0.100E+04
1.	18.	0.	0.100E+04

```
PARAMETER NUMBER:    5, PARAMETER IDENTIFIER:  KV
LN:  1, VECTOR NLL:    1    0    0    0
SFAC=  0.100     , MULT. MATRIX    0, ZONE MATRIX    0
ZONES =    0    0

PARAMETER NUMBER:    6, PARAMETER IDENTIFIER:  S1
LN:  1, VECTOR NLL:   -2    0    0    0

PARAMETER NUMBER:    7, PARAMETER IDENTIFIER:  T
LN:  1, VECTOR NLL:    2    0    0    0
SFAC=   50.0     , MULT. MATRIX    1, ZONE MATRIX    0
ZONES =    0    0

PARAMETER NUMBER:    8, PARAMETER IDENTIFIER:  RCH
LN:  0, VECTOR NLL:    1    0   15   -1
SFAC=   1.00     , MULT. MATRIX    0, ZONE MATRIX    1
ZONES =    1    0

PARAMETER NUMBER:    9, PARAMETER IDENTIFIER:  RCH
LN:  0, VECTOR NLL:    1    0   15   -1
SFAC=   1.00     , MULT. MATRIX    0, ZONE MATRIX    1
ZONES =    2    0
```

MULTIPLICATION MATRIX WILL BE READ ON UNIT 16 USING FORMAT: (18F3.0)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
2	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
3	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
4	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
5	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
6	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
7	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
8	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
9	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
10	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
11	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
12	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
13	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
14	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
15	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000
16	1.000	1.000	2.000	2.000	3.000	3.000	4.000	4.000	5.000	5.000	6.000	6.000	7.000	7.000	8.000	8.000	9.000	9.000

```
 17    1.000      1.000      2.000      2.000      3.000      3.000      4.000      4.000      5.000      5.000
       6.000      6.000      7.000      7.000      8.000      8.000      9.000      9.000
 18    1.000      1.000      2.000      2.000      3.000      3.000      4.000      4.000      5.000      5.000
       6.000      6.000      7.000      7.000      8.000      8.000      9.000      9.000

                                  ZONE MATRIX WILL BE READ ON UNIT 16 USING FORMAT: (9I8)
                          ----------------------------------------------------------------------------------

           1          2          3          4          5          6          7          8          9         10
          11         12         13         14         15         16         17         18
.........................................................................................................................
  1           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
  2           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
  3           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
  4           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
  5           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
  6           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
  7           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
  8           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
  9           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
 10           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
 11           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
 12           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
 13           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
 14           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
 15           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
 16           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
 17           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2
 18           1          1          1          1          1          1          1          1          1          2
              2          2          2          2          2          2          2          2

NSN CAN BE REDUCED FROM   56 TO   20

INPUT ERROR VARIANCES FOR HEADS AND FLOWS:         1.000            1.000
INPUT UNIT NUMBERS FOR HEADS AND FLOWS :    0     0
ESTIMATED COMMON ERROR VARIANCE :       1.000

OBSERVED HEAD DATA

  OBS#  ID   LAYER ROW COLUMN   TIME STEP  ROW/COLUMN/TIME OFFSETS  OBSERVATION  VARIANCE
 TRANSIENT DATA AT THIS LOCATION, ITT =   2
    1  1.0     1     3    1        -3       0.00      0.00      0.00       0.00000      0.00000
    1  1.0                          0                           0.00       101.80       1.0025
    2  1.1                          1                           0.00      -0.29007E-01 0.25000E-02
    3  1.12                        12                           0.00      -0.12900     0.25000E-02

 TRANSIENT DATA AT THIS LOCATION, ITT =   2
    4  2.0     1     4    4        -5       0.00      0.00      0.00       0.00000      0.00000
    4  2.0                          0                           0.00       128.12       1.0025
    5  2.1                          1                           0.00      -0.41016E-01 0.25000E-02
    6  2.2                          2                           0.00      -0.55701     0.25000E-02
    7  2.8                          8                           0.00      -11.531      0.25000E-02
    8  2.12                        12                           0.00      -14.184      0.25000E-02

 TRANSIENT DATA AT THIS LOCATION, ITT =   2
    9  3.0     1    10    9        -3       0.00      0.00      0.00       0.00000      0.00000
    9  3.0                          0                           0.00       156.68       1.0025
   10  3.1                          1                           0.00      -4.3810      0.25000E-02
   11  3.12                        12                           0.00      -42.540      0.25000E-02
```

```
TRANSIENT DATA AT THIS LOCATION, ITT =    2
   12  4.0      1    13     4         -3       0.00      0.00      0.00       0.00000      0.00000
   12  4.0                             0                             0.00        124.89       1.0025
   13  4.1                             1                             0.00      -0.67001E-01 0.25000E-02
   14  4.12                           12                             0.00       -14.304      0.25000E-02

TRANSIENT DATA AT THIS LOCATION, ITT =    2
   15  5.0      1    14     6         -3       0.00      0.00      0.00       0.00000      0.00000
   15  5.0                             0                             0.00        140.96       1.0025
   16  5.1                             1                             0.00      -0.59998E-01 0.25000E-02
   17  5.12                           12                             0.00       -21.676      0.25000E-02

TRANSIENT DATA AT THIS LOCATION, ITT =    2
   18  6.0      2     4     4         -3       0.00      0.00      0.00       0.00000      0.00000
   18  6.0                             0                             0.00        126.54       1.0025
   19  6.1                             1                             0.00       0.50049E-02 0.25000E-02
   20  6.12                           12                             0.00       -14.365      0.25000E-02

TRANSIENT DATA AT THIS LOCATION, ITT =    2
   21  7.0      2    10     1         -3       0.00      0.00      0.00       0.00000      0.00000
   21  7.0                             0                             0.00        101.11       1.0025
   22  7.1                             1                             0.00       0.48004E-01 0.25000E-02
   23  7.12                           12                             0.00      -0.56799      0.25000E-02

TRANSIENT DATA AT THIS LOCATION, ITT =    2
   24  8.0      2    10     9         -3       0.00      0.00      0.00       0.00000      0.00000
   24  8.0                             0                             0.00        158.14       1.0025
   25  8.1                             1                             0.00       -5.5330      0.25000E-02
   26  8.12                           12                             0.00       -43.217      0.25000E-02

TRANSIENT DATA AT THIS LOCATION, ITT =    2
   27  9.0      2    10    18         -3       0.00      0.00      0.00       0.00000      0.00000
   27  9.0                             0                             0.00        176.37       1.0025
   28  9.1                             1                             0.00      -0.10071E-02 0.25000E-02
   29  9.12                           12                             0.00       -38.242      0.25000E-02

TRANSIENT DATA AT THIS LOCATION, ITT =    2
   30  0.0      2    18     6         -3       0.00      0.00      0.00       0.00000      0.00000
   30  0.0                             0                             0.00        142.02       1.0025
   31  0.1                             1                             0.00      -0.13000E-01 0.25000E-02
   32  0.12                           12                             0.00       -19.921      0.25000E-02


HEAD-DEPENDENT FLOW DATA

  GROUP#  1, BOUNDARY TYPE  3, # OF FLOWS    3, # OF CELLS IN GROUP  -18
                                         OBSERVED STREAMFLOW
     OBS# ID     TIME STEP AND OFFSET    GAIN(-) OR LOSS(+)  VARIANCE
       33 SS          0      0.00             -4.40           0.160
       34 TR3         3      0.00             -4.10           0.144
       35 TR12       12      0.00             -2.20           0.441E-01

    SEGMENT REACH    FACTOR
         1.    1.     1.00
         1.    2.     1.00
         1.    3.     1.00
         1.    4.     1.00
         1.    5.     1.00
         1.    6.     1.00
         1.    7.     1.00
         1.    8.     1.00
         1.    9.     1.00
         1.   10.     1.00
         1.   11.     1.00
         1.   12.     1.00
         1.   13.     1.00
         1.   14.     1.00
         1.   15.     1.00
         1.   16.     1.00
         1.   17.     1.00
         1.   18.     1.00
```

```
REGRESSION DATA:
MAX. PARAMETER CORRECTION (DMAX)  =  2.0000
SEARCH DIR. ADJUSTMENT PAR. (CSA) = 0.80000E-01
CLOSURE CRITERION (TOL) --------- = 0.10000E-01
MAXIMUM NO. OF ITERATIONS (ITMXP) =       10
MODIFY CONV. CRITERIA (>0) (FCONV)= 0.00000
USE PRIOR PAR EST (>0) (IPRIOR)   =        0
OUTPUT UNIT FOR RESAN (IOUR) ---- =        0
FORMAT CODE FOR COV AND COR (IPRC)=       12
PRINT SENS., ETC. (>0)     (IPRINT)=       0
PRINT LAST HEADS (1) (KPRINT) --- =        0
PRINT EIGEN V&V OF COV(>0)(LPRINT)=        0
CALC. SEN. W/ LAST PAR (>0)(LASTX)=        0
NO. FLETCHER-REEVES ITERS    (NFIT)=       0
SUM OF SQUARES CLOSURE CRIT.(SOSC)= 0.00000

PARAMETER INFORMATION:
 (CONVERGENCE CRITERIA LISTED HERE ARE USED TO SOLVE FOR SENSITIVITY-EQUATION SENSITIVITIES)
    #  ID   INITIAL VALUE  LN      CONVERGENCE CRITERIA    GROUP#  WEIGHT OF PRIOR EST.
    1  Q        -1.100     0     0.909E-07    0.909E-07     2
    2  S1       0.1300E-02 1     0.769E-04    0.769E-04     3
    3  T        0.3000E-03 1     0.333E-03    0.333E-03     4
    4  KST      0.1200E-02 1     0.833E-04    0.833E-04     5
    5  KV       0.1000E-06 1      1.00         1.00         6
    6  S1       0.2000E-03 1     0.500E-03    0.500E-03     3
    7  T        0.4000E-04 1     0.250E-02    0.250E-02     4
    8  RCH      0.2000E-07 0      5.00         5.00         7
    9  RCH      0.1000E-07 0      10.0         10.0         7
          2 WELLS
                                                LAYER    ROW    COL    STRESS RATE    WELL NO.
                                                -------------------------------------------
                                                  1       9     10     0.00000           1
                                                  2       9     10     0.00000           2
```

```
                                                RECHARGE WILL BE READ ON UNIT 30 USING FORMAT: (9F8.0)
                         -------------------------------------------------------------------------------------
          1          2          3          4          5          6          7          8          9          10
         11         12         13         14         15         16         17         18
.....................................................................................................................
  1  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
  2  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
  3  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
  4  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
  5  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
  6  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
  7  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
  8  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
  9  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
 10  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
 11  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
 12  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
 13  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
 14  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
 15  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
 16  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
 17  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
 18  1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.0000E-08 1.5000E-08
     1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08 1.5000E-08
```

36 HEAD-DEPENDENT BOUNDARY NODES

LAYER	ROW	COL	ELEVATION	CONDUCTANCE	BOUND NO.
1	1	18	350.0	0.1000E-06	1
1	2	18	350.0	0.1000E-06	2
1	3	18	350.0	0.1000E-06	3
1	4	18	350.0	0.1000E-06	4
1	5	18	350.0	0.1000E-06	5
1	6	18	350.0	0.1000E-06	6
1	7	18	350.0	0.1000E-06	7
1	8	18	350.0	0.1000E-06	8
1	9	18	350.0	0.1000E-06	9
1	10	18	350.0	0.1000E-06	10
1	11	18	350.0	0.1000E-06	11
1	12	18	350.0	0.1000E-06	12
1	13	18	350.0	0.1000E-06	13
1	14	18	350.0	0.1000E-06	14
1	15	18	350.0	0.1000E-06	15
1	16	18	350.0	0.1000E-06	16
1	17	18	350.0	0.1000E-06	17
1	18	18	350.0	0.1000E-06	18
2	1	18	350.0	0.1000E-06	19
2	2	18	350.0	0.1000E-06	20
2	3	18	350.0	0.1000E-06	21
2	4	18	350.0	0.1000E-06	22
2	5	18	350.0	0.1000E-06	23
2	6	18	350.0	0.1000E-06	24
2	7	18	350.0	0.1000E-06	25
2	8	18	350.0	0.1000E-06	26
2	9	18	350.0	0.1000E-06	27
2	10	18	350.0	0.1000E-06	28
2	11	18	350.0	0.1000E-06	29
2	12	18	350.0	0.1000E-06	30
2	13	18	350.0	0.1000E-06	31
2	14	18	350.0	0.1000E-06	32
2	15	18	350.0	0.1000E-06	33
2	16	18	350.0	0.1000E-06	34
2	17	18	350.0	0.1000E-06	35
2	18	18	350.0	0.1000E-06	36

18 STREAM NODES

LAYER	ROW	COL	SEGMENT NUMBER	REACH NUMBER	STREAMFLOW	STREAM STAGE	STREAMBED CONDUCTANCE	STREAMBED BOT ELEVATION	STREAMBED TOP ELEVATION
1	1	1	1	1	10.00	100.0	1.000	90.00	95.00
1	2	1	1	2	0.0000	100.0	1.000	90.00	95.00
1	3	1	1	3	0.0000	100.0	1.000	90.00	95.00
1	4	1	1	4	0.0000	100.0	1.000	90.00	95.00
1	5	1	1	5	0.0000	100.0	1.000	90.00	95.00
1	6	1	1	6	0.0000	100.0	1.000	90.00	95.00
1	7	1	1	7	0.0000	100.0	1.000	90.00	95.00
1	8	1	1	8	0.0000	100.0	1.000	90.00	95.00
1	9	1	1	9	0.0000	100.0	1.000	90.00	95.00
1	10	1	1	10	0.0000	100.0	1.000	90.00	95.00
1	11	1	1	11	0.0000	100.0	1.000	90.00	95.00
1	12	1	1	12	0.0000	100.0	1.000	90.00	95.00
1	13	1	1	13	0.0000	100.0	1.000	90.00	95.00
1	14	1	1	14	0.0000	100.0	1.000	90.00	95.00
1	15	1	1	15	0.0000	100.0	1.000	90.00	95.00
1	16	1	1	16	0.0000	100.0	1.000	90.00	95.00
1	17	1	1	17	0.0000	100.0	1.000	90.00	95.00
1	18	1	1	18	0.0000	100.0	1.000	90.00	95.00

MODEL INPUT MATRICES CALCULATED WITH INITIAL PARAMETER VALUES

TRANSMIS. ALONG ROWS LAYER 1

	1 10	2 11	3 12	4 13	5 14	6 15	7 16	8 17	9 18
1	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
2	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
3	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
4	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
5	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
6	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
7	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
8	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
9	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
10	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
11	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
12	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
13	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
14	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
15	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
16	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
17	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
18	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02
	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02	1.500000E-02

TRANSMIS. ALONG ROWS LAYER 2

	1 10	2 11	3 12	4 13	5 14	6 15	7 16	8 17	9 18
1	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
2	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
3	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
4	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
5	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
6	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
7	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
8	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
9	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
10	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02

11	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
12	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
13	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
14	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
15	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
16	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
17	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02
18	2.000000E-03	2.000000E-03	4.000000E-03	4.000000E-03	5.999999E-03	5.999999E-03	7.999999E-03	7.999999E-03	9.999998E-03
	9.999998E-03	1.200000E-02	1.200000E-02	1.400000E-02	1.400000E-02	1.600000E-02	1.600000E-02	1.800000E-02	1.800000E-02

VERTICAL LEAKANCE LAYER 1

	1 10	2 11	3 12	4 13	5 14	6 15	7 16	8 17	9 18
1	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
2	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
3	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
4	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
5	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
6	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
7	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
8	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
9	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
10	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
11	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
12	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
13	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
14	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
15	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
16	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
17	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
18	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09
	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09	9.999995E-09

RECHARGE FOR TIME STEP 0

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
2	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
3	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
4	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
5	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
6	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
7	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
8	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
9	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
10	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
11	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
12	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
13	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
14	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
15	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
16	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
17	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09
18	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	2.000000E-08	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09	9.999999E-09

2 WELLS

LAYER	ROW	COL	STRESS RATE	WELL NO.
1	9	10	-1.0000	1
2	9	10	-1.0000	2

REUSING RECH FROM LAST STRESS PERIOD
REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS PERIOD
REUSING STREAM NODES FROM LAST STRESS PERIOD
REUSING WELLS FROM LAST STRESS PERIOD
REUSING RECH FROM LAST STRESS PERIOD
REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS PERIOD
REUSING STREAM NODES FROM LAST STRESS PERIOD
REUSING WELLS FROM LAST STRESS PERIOD
REUSING RECH FROM LAST STRESS PERIOD
REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS PERIOD
REUSING STREAM NODES FROM LAST STRESS PERIOD
REUSING WELLS FROM LAST STRESS PERIOD
REUSING RECH FROM LAST STRESS PERIOD
REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS PERIOD
REUSING STREAM NODES FROM LAST STRESS PERIOD
REUSING WELLS FROM LAST STRESS PERIOD

REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS PERIOD
REUSING STREAM NODES FROM LAST STRESS PERIOD
REUSING WELLS FROM LAST STRESS PERIOD
REUSING RECH FROM LAST STRESS PERIOD
REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS PERIOD
REUSING STREAM NODES FROM LAST STRESS PERIOD
REUSING WELLS FROM LAST STRESS PERIOD
REUSING RECH FROM LAST STRESS PERIOD
REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS PERIOD
REUSING STREAM NODES FROM LAST STRESS PERIOD
REUSING WELLS FROM LAST STRESS PERIOD
REUSING RECH FROM LAST STRESS PERIOD
REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS PERIOD
REUSING STREAM NODES FROM LAST STRESS PERIOD
REUSING WELLS FROM LAST STRESS PERIOD
REUSING RECH FROM LAST STRESS PERIOD
REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS PERIOD
REUSING STREAM NODES FROM LAST STRESS PERIOD
REUSING WELLS FROM LAST STRESS PERIOD
REUSING RECH FROM LAST STRESS PERIOD
REUSING HEAD-DEPENDENT BOUNDS FROM LAST STRESS PERIOD
REUSING STREAM NODES FROM LAST STRESS PERIOD

DATA AT HEAD LOCATIONS

OBS#	ID	LAYER,ROW,COL			TIME STEP	ROW/COL/TIME OFFSETS			MEAS. HEAD	CALC. HEAD	RESIDUAL	WEIGHT**.5	WEIGHTED RESIDUAL
1	1.0	1	3	1	0	0.00	0.00	0.00	101.804	100.225	1.58	0.999	1.58
2	1.1	1	3	1	1	0.00	0.00	0.00	-0.029	0.000	-0.290E-01	20.0	-0.580
3	1.12	1	3	1	12	0.00	0.00	0.00	-0.129	-0.091	-0.384E-01	20.0	-0.767
4	2.0	1	4	4	0	0.00	0.00	0.00	128.117	139.329	-11.2	0.999	-11.2
5	2.1	1	4	4	1	0.00	0.00	0.00	-0.041	-0.010	-0.315E-01	20.0	-0.630
6	2.2	1	4	4	2	0.00	0.00	0.00	-0.557	-0.276	-0.281	20.0	-5.62
7	2.8	1	4	4	8	0.00	0.00	0.00	-11.531	-12.963	1.43	20.0	28.6
8	2.12	1	4	4	12	0.00	0.00	0.00	-14.184	-18.771	4.59	20.0	91.7
9	3.0	1	10	9	0	0.00	0.00	0.00	156.678	174.360	-17.7	0.999	-17.7
10	3.1	1	10	9	1	0.00	0.00	0.00	-4.381	-3.666	-0.715	20.0	-14.3
11	3.12	1	10	9	12	0.00	0.00	0.00	-42.540	-56.237	13.7	20.0	274.
12	4.0	1	13	4	0	0.00	0.00	0.00	124.893	139.329	-14.4	0.999	-14.4
13	4.1	1	13	4	1	0.00	0.00	0.00	-0.067	-0.016	-0.507E-01	20.0	-1.01
14	4.12	1	13	4	12	0.00	0.00	0.00	-14.304	-18.848	4.54	20.0	90.9
15	5.0	1	14	6	0	0.00	0.00	0.00	140.961	157.130	-16.2	0.999	-16.1
16	5.1	1	14	6	1	0.00	0.00	0.00	-0.060	-0.037	-0.232E-01	20.0	-0.464
17	5.12	1	14	6	12	0.00	0.00	0.00	-21.676	-28.461	6.78	20.0	136.
18	6.0	2	4	4	0	0.00	0.00	0.00	126.537	139.629	-13.1	0.999	-13.1
19	6.1	2	4	4	1	0.00	0.00	0.00	0.005	-0.013	0.175E-01	20.0	0.350
20	6.12	2	4	4	12	0.00	0.00	0.00	-14.365	-19.183	4.82	20.0	96.4
21	7.0	2	10	1	0	0.00	0.00	0.00	101.112	102.851	-1.74	0.999	-1.74
22	7.1	2	10	1	1	0.00	0.00	0.00	0.048	-0.001	0.491E-01	20.0	0.982
23	7.12	2	10	1	12	0.00	0.00	0.00	-0.568	-1.372	0.804	20.0	16.1
24	8.0	2	10	9	0	0.00	0.00	0.00	158.135	173.954	-15.8	0.999	-15.8
25	8.1	2	10	9	1	0.00	0.00	0.00	-5.533	-5.807	0.274	20.0	5.47
26	8.12	2	10	9	12	0.00	0.00	0.00	-43.217	-57.252	14.0	20.0	281.
27	9.0	2	10	18	0	0.00	0.00	0.00	176.374	190.298	-13.9	0.999	-13.9
28	9.1	2	10	18	1	0.00	0.00	0.00	-0.001	-0.050	0.494E-01	20.0	0.989
29	9.12	2	10	18	12	0.00	0.00	0.00	-38.242	-49.510	11.3	20.0	225.
30	0.0	2	18	6	0	0.00	0.00	0.00	142.020	157.039	-15.0	0.999	-15.0
31	0.1	2	18	6	1	0.00	0.00	0.00	-0.013	-0.004	-0.873E-02	20.0	-0.175
32	0.12	2	18	6	12	0.00	0.00	0.00	-19.921	-26.127	6.21	20.0	124.

STATISTICS FOR THESE RESIDUALS :
MAXIMUM WEIGHTED RESIDUAL : 0.281E+03 OBS# 26
MINIMUM WEIGHTED RESIDUAL :-0.177E+02 OBS# 9
AVERAGE WEIGHTED RESIDUAL : 0.385E+02
RESIDUALS >= 0. : 15
RESIDUALS < 0. : 17
NUMBER OF RUNS : 19 IN 32 OBSERVATIONS

SUM OF SQUARED WEIGHTED RESIDUALS (HEADS) 0.26751E+06

DATA FOR FLOWS

OBS#	ID	TIME STEP	MEAS. FLOW	CALC. FLOW	RESIDUAL	WEIGHT**.5	WEIGHTED RESIDUAL
33	SS	0	-4.40	-4.86	0.460	2.50	1.15
34	TR3	3	-4.10	-4.72	0.618	2.63	1.63
35	TR12	12	-2.20	-2.86	0.663	4.76	3.16

```
STATISTICS FOR THESE RESIDUALS :
MAXIMUM WEIGHTED RESIDUAL  :-0.115E+01 OBS#     33
MINIMUM WEIGHTED RESIDUAL  :-0.316E+01 OBS#     35
AVERAGE WEIGHTED RESIDUAL  :-0.198E+01
# RESIDUALS >= 0. :        0
# RESIDUALS < 0.  :        3
NUMBER OF RUNS  :   19  IN   35 OBSERVATIONS

SUM OF SQUARED WEIGHTED RESIDUALS (ALL DEP. VAR.)  0.26752E+06
STATISTICS FOR ALL RESIDUALS :
AVERAGE WEIGHTED RESIDUAL  :-0.353E+02
# RESIDUALS >= 0. :       17
# RESIDUALS < 0.  :       18
NUMBER OF RUNS  :   19  IN   35 OBSERVATIONS
RUNS STATISTIC (TOO FEW RUNS):     0.348
 (IF #NEG>10 AND #POS>10, P(STAT < -1.28) = 0.10,
                          P(STAT < -1.645) = 0.05,
                          P(STAT < -1.96) = 0.025)
RUNS STATISTIC (TOO MANY RUNS):     0.491E-02
 (IF #NEG>10 AND #POS>10, P(STAT > 1.28) = 0.10,
                          P(STAT > 1.645) = 0.05,
                          P(STAT > 1.96) = 0.025)
```

SCALED SENSITIVITIES (SCALED BY B*(WT**.5))

PARAMETER # :		1	2	3	4	5	6	7	8	9
PARAMETER ID :		Q	S1	T	KST	KV	S1	T	RCH	RCH
OBS# & ID										
1	1.0	0.000	0.000	-0.902E-04	1.51	-0.712E-06	0.000	-0.392E-04	0.150	0.749E-01
2	1.1	-0.262E-03	-0.573E-02	0.608E-02	-0.204E-02	-0.903E-03	-0.198E-02	0.404E-02	0.000	0.000
3	1.12	-1.81	-2.22	2.30	-12.2	0.198E-01	-0.439	1.02	0.000	0.000
4	2.0	0.000	0.000	270.	1.51	4.58	0.000	55.4	24.0	15.3
5	2.1	-0.190	-3.31	2.15	-0.237E-03	-0.489	-1.11	2.08	0.000	0.000
6	2.2	-5.52	-66.5	33.1	-0.275E-01	-2.57	-15.5	24.2	0.000	0.512E-04
7	2.8	-259.	-784.	-930.	-6.44	-8.78	-155.	-70.7	0.102E-03	0.512E-04
8	2.12	-375.	-447.	-0.210E+04	-11.4	-22.9	-88.2	-362.	0.102E-03	0.512E-04
9	3.0	0.000	0.000	470.	1.51	7.95	0.000	159.	38.3	35.9
10	3.1	-73.3	-394.	-14.9	0.401E-05	112.	-83.7	-94.5	0.102E-03	0.000
11	3.12	-0.112E+04	-975.	-0.569E+04	-10.1	70.7	-192.	-0.260E+04	0.307E-03	0.307E-03
12	4.0	0.000	0.000	270.	1.51	4.58	0.000	55.4	24.0	15.3
13	4.1	-0.325	-5.29	3.33	-0.329E-03	-0.690	-1.74	3.11	0.000	0.000
14	4.12	-377.	-447.	-0.211E+04	-11.2	-21.3	-88.2	-370.	0.102E-03	0.512E-04
15	5.0	0.000	0.000	377.	1.51	6.35	0.000	101.	32.9	24.1
16	5.1	-0.736	-10.5	5.94	-0.321E-04	-1.24	-3.55	6.19	0.000	0.000
17	5.12	-569.	-690.	-0.302E+04	-10.6	-28.0	-136.	-745.	0.205E-03	0.205E-03
18	6.0	0.000	0.000	271.	1.51	10.2	0.000	54.2	24.0	15.6
19	6.1	-0.249	-3.60	2.14	-0.245E-03	-1.95	-1.84	3.71	0.000	0.000
20	6.12	-384.	-456.	-0.213E+04	-11.3	-165.	-90.6	-306.	0.205E-03	0.102E-03
21	7.0	0.000	0.000	18.9	1.51	38.4	0.000	-21.1	1.81	1.04
22	7.1	-0.227E-01	-0.345	0.229	-0.892E-02	-0.368	-0.192	0.482	0.000	0.000
23	7.12	-27.4	-30.7	-148.	-13.2	-378.	-6.15	218.	0.640E-05	0.320E-05
24	8.0	0.000	0.000	467.	1.51	2.15	0.000	162.	37.8	36.1
25	8.1	-116.	-275.	-73.3	0.000	-456.	-238.	-94.9	0.102E-03	0.000
26	8.12	-0.115E+04	-975.	-0.544E+04	-10.1	-283.	-194.	-0.289E+04	0.307E-03	0.205E-03
27	9.0	0.000	0.000	540.	1.51	0.938	0.000	236.	38.1	52.1
28	9.1	-1.01	-9.93	3.70	0.120E-04	-11.8	-6.73	15.7	0.102E-03	0.000
29	9.12	-990.	-0.132E+04	-0.456E+04	-9.24	-76.5	-262.	-0.195E+04	0.410E-03	0.410E-03
30	0.0	0.000	0.000	375.	1.51	5.33	0.000	103.	32.6	24.4
31	0.1	-0.858E-01	-1.44	0.858	-0.160E-04	-0.914	-0.763	1.74	0.000	0.000
32	0.12	-523.	-695.	-0.276E+04	-10.4	-128.	-138.	-528.	0.205E-03	0.102E-03
33	SS	0.000	0.000	0.488E-02	0.135E-04	0.417E-04	0.000	0.211E-02	-8.10	-4.05
34	TR3	0.375	3.25	-3.62	-0.161E-01	0.592E-02	0.709	-1.26	-8.53	-4.26
35	TR12	9.51	11.4	-12.1	-0.311E-01	-0.180	2.26	-4.83	-15.4	-7.71
((SUM OF THE SQUARED VALUES)**.5)/ND		0.617E+02	0.703E+02	0.306E+03	0.102E+01	0.202E+02	0.154E+02	0.129E+03	0.265E+01	0.244E+01

```
UNSCALED RIGHT-HAND SIDE VECTOR (= -0.5*GRADIENT)
 0.10071E+07  0.17486E+06  0.68179E+06   2097.1        7937.7        27104.      0.22105E+06  -.19133E+12  -.33156E+12

ITERATION NO. =     1
VALUES FROM LEAST-SQUARES REGRESSION PROCEDURE :
  DET OF SCALED LEAST-SQUARES MATRIX = 0.60737E-09
  MARQUARDT PARAMETER (AMP)--------- = 0.00000
  FACTOR FOR SCALING PAR. CHANGE (AP)=  1.0000
  MAX. FRACTIONAL PAR. CHANGE (DMX)  = 0.85868
  MAX. FRAC. CHANGE OCCURRED FOR PAR.#      5

UPDATED ESTIMATES OF REGRESSION PARAMETERS :
    Q            S1           T           KST          KV           S1           T           RCH          RCH
 -1.0009      0.12205E-02  0.39340E-03  0.20973E-03  0.18587E-06  0.87396E-04  0.42774E-04  0.12443E-07  0.13861E-07

ITERATION NO. =     2
VALUES FROM LEAST-SQUARES REGRESSION PROCEDURE :
  DET OF SCALED LEAST-SQUARES MATRIX = 0.86775E-09
  MARQUARDT PARAMETER (AMP)--------- = 0.00000
  FACTOR FOR SCALING PAR. CHANGE (AP)=  1.0000
  MAX. FRACTIONAL PAR. CHANGE (DMX)  =  1.4328
  MAX. FRAC. CHANGE OCCURRED FOR PAR.#      4

UPDATED ESTIMATES OF REGRESSION PARAMETERS :
    Q            S1           T           KST          KV           S1           T           RCH          RCH
 -1.0594      0.11481E-02  0.42276E-03  0.51023E-03  0.21470E-06  0.54922E-04  0.47328E-04  0.11028E-07  0.15692E-07

ITERATION NO. =     3
VALUES FROM LEAST-SQUARES REGRESSION PROCEDURE :
  DET OF SCALED LEAST-SQUARES MATRIX = 0.11453E-08
  MARQUARDT PARAMETER (AMP)--------- = 0.00000
  FACTOR FOR SCALING PAR. CHANGE (AP)=  1.0000
  MAX. FRACTIONAL PAR. CHANGE (DMX)  = 0.88538
  MAX. FRAC. CHANGE OCCURRED FOR PAR.#      4

UPDATED ESTIMATES OF REGRESSION PARAMETERS :
    Q            S1           T           KST          KV           S1           T           RCH          RCH
 -1.0736      0.11373E-02  0.42541E-03  0.96197E-03  0.21483E-06  0.63302E-04  0.48216E-04  0.10828E-07  0.15995E-07

ITERATION NO. =     4
VALUES FROM LEAST-SQUARES REGRESSION PROCEDURE :
  DET OF SCALED LEAST-SQUARES MATRIX = 0.12056E-08
  MARQUARDT PARAMETER (AMP)--------- = 0.00000
  FACTOR FOR SCALING PAR. CHANGE (AP)=  1.0000
  MAX. FRACTIONAL PAR. CHANGE (DMX)  = 0.33946
  MAX. FRAC. CHANGE OCCURRED FOR PAR.#      4

UPDATED ESTIMATES OF REGRESSION PARAMETERS :
    Q            S1           T           KST          KV           S1           T           RCH          RCH
 -1.0742      0.11371E-02  0.42563E-03  0.12885E-02  0.21508E-06  0.63091E-04  0.48228E-04  0.10821E-07  0.16009E-07

ITERATION NO. =     5
VALUES FROM LEAST-SQUARES REGRESSION PROCEDURE :
  DET OF SCALED LEAST-SQUARES MATRIX = 0.12128E-08
  MARQUARDT PARAMETER (AMP)--------- = 0.00000
  FACTOR FOR SCALING PAR. CHANGE (AP)=  1.0000
  MAX. FRACTIONAL PAR. CHANGE (DMX)  = 0.54045E-01
  MAX. FRAC. CHANGE OCCURRED FOR PAR.#      4

UPDATED ESTIMATES OF REGRESSION PARAMETERS :
    Q            S1           T           KST          KV           S1           T           RCH          RCH
 -1.0742      0.11370E-02  0.42566E-03  0.13582E-02  0.21507E-06  0.63170E-04  0.48232E-04  0.10820E-07  0.16011E-07

ITERATION NO. =     6
VALUES FROM LEAST-SQUARES REGRESSION PROCEDURE :
  DET OF SCALED LEAST-SQUARES MATRIX = 0.24034E-07
  MARQUARDT PARAMETER (AMP)--------- = 0.10000E-02
  FACTOR FOR SCALING PAR. CHANGE (AP)=  1.0000
  MAX. FRACTIONAL PAR. CHANGE (DMX)  = 0.17538E-02
  MAX. FRAC. CHANGE OCCURRED FOR PAR.#      4

UPDATED ESTIMATES OF REGRESSION PARAMETERS :
    Q            S1           T           KST          KV           S1           T           RCH          RCH
 -1.0742      0.11370E-02  0.42565E-03  0.13605E-02  0.21506E-06  0.63185E-04  0.48232E-04  0.10819E-07  0.16011E-07

PARAMETER ESTIMATION CONVERGED BY SATISFYING TOL CRITERIA
```

DATA AT HEAD LOCATIONS

OBS#	ID	LAYER,ROW,COL			TIME STEP	ROW/COL/TIME OFFSETS			MEAS. HEAD	CALC. HEAD	RESIDUAL	WEIGHT**.5	WEIGHTED RESIDUAL
1	1.0	1	3	1	0	0.00	0.00	0.00	101.804	100.178	1.63	0.999	1.62
2	1.1	1	3	1	1	0.00	0.00	0.00	-0.029	0.000	-0.289E-01	20.0	-0.578
3	1.12	1	3	1	12	0.00	0.00	0.00	-0.129	-0.084	-0.450E-01	20.0	-0.900
4	2.0	1	4	4	0	0.00	0.00	0.00	128.117	126.993	1.12	0.999	1.12
5	2.1	1	4	4	1	0.00	0.00	0.00	-0.041	-0.034	-0.728E-02	20.0	-0.146
6	2.2	1	4	4	2	0.00	0.00	0.00	-0.557	-0.544	-0.131E-01	20.0	-0.261
7	2.8	1	4	4	8	0.00	0.00	0.00	-11.531	-11.554	0.233E-01	20.0	0.466
8	2.12	1	4	4	12	0.00	0.00	0.00	-14.184	-14.192	0.845E-02	20.0	0.169
9	3.0	1	10	9	0	0.00	0.00	0.00	156.678	157.131	-0.453	0.999	-0.452
10	3.1	1	10	9	1	0.00	0.00	0.00	-4.381	-4.319	-0.620E-01	20.0	-1.24
11	3.12	1	10	9	12	0.00	0.00	0.00	-42.540	-42.593	0.530E-01	20.0	1.06
12	4.0	1	13	4	0	0.00	0.00	0.00	124.893	126.993	-2.10	0.999	-2.10
13	4.1	1	13	4	1	0.00	0.00	0.00	-0.067	-0.051	-0.155E-01	20.0	-0.310
14	4.12	1	13	4	12	0.00	0.00	0.00	-14.304	-14.251	-0.530E-01	20.0	-1.06
15	5.0	1	14	6	0	0.00	0.00	0.00	140.961	140.914	0.471E-01	0.999	0.471E-01
16	5.1	1	14	6	1	0.00	0.00	0.00	-0.060	-0.099	0.387E-01	20.0	0.773
17	5.12	1	14	6	12	0.00	0.00	0.00	-21.676	-21.657	-0.185E-01	20.0	-0.370
18	6.0	2	4	4	0	0.00	0.00	0.00	126.537	127.204	-0.667	0.999	-0.666
19	6.1	2	4	4	1	0.00	0.00	0.00	0.005	-0.038	0.433E-01	20.0	0.867
20	6.12	2	4	4	12	0.00	0.00	0.00	-14.365	-14.367	0.233E-02	20.0	0.467E-01
21	7.0	2	10	1	0	0.00	0.00	0.00	101.112	101.200	-0.876E-01	0.999	-0.875E-01
22	7.1	2	10	1	1	0.00	0.00	0.00	0.048	-0.003	0.508E-01	20.0	1.02
23	7.12	2	10	1	12	0.00	0.00	0.00	-0.568	-0.648	0.799E-01	20.0	1.60
24	8.0	2	10	9	0	0.00	0.00	0.00	158.135	157.114	1.02	0.999	1.02
25	8.1	2	10	9	1	0.00	0.00	0.00	-5.533	-5.535	0.153E-02	20.0	0.305E-01
26	8.12	2	10	9	12	0.00	0.00	0.00	-43.217	-43.108	-0.109	20.0	-2.18
27	9.0	2	10	18	0	0.00	0.00	0.00	176.374	176.750	-0.376	0.999	-0.376
28	9.1	2	10	18	1	0.00	0.00	0.00	-0.001	-0.100	0.993E-01	20.0	1.99
29	9.12	2	10	18	12	0.00	0.00	0.00	-38.242	-38.355	0.113	20.0	2.25
30	0.0	2	18	6	0	0.00	0.00	0.00	142.020	141.022	0.998	0.999	0.997
31	0.1	2	18	6	1	0.00	0.00	0.00	-0.013	-0.017	0.378E-02	20.0	0.757E-01
32	0.12	2	18	6	12	0.00	0.00	0.00	-19.921	-19.876	-0.447E-01	20.0	-0.894

STATISTICS FOR THESE RESIDUALS :
MAXIMUM WEIGHTED RESIDUAL : 0.225E+01 OBS# 29
MINIMUM WEIGHTED RESIDUAL :-0.218E+01 OBS# 26
AVERAGE WEIGHTED RESIDUAL : 0.110E+00
RESIDUALS >= 0. : 17
RESIDUALS < 0. : 15
NUMBER OF RUNS : 16 IN 32 OBSERVATIONS

SUM OF SQUARED WEIGHTED RESIDUALS (HEADS) 36.127

DATA FOR FLOWS

OBS#	ID	TIME STEP	MEAS. FLOW	CALC. FLOW	RESIDUAL	WEIGHT**.5	WEIGHTED RESIDUAL
33	SS	0	-4.40	-4.35	-0.528E-01	2.50	-0.132
34	TR3	3	-4.10	-4.06	-0.417E-01	2.63	-0.110
35	TR12	12	-2.20	-2.25	0.495E-01	4.76	0.236

STATISTICS FOR THESE RESIDUALS :
MAXIMUM WEIGHTED RESIDUAL : 0.236E+00 OBS# 35
MINIMUM WEIGHTED RESIDUAL :-0.132E+00 OBS# 33
AVERAGE WEIGHTED RESIDUAL :-0.199E-02
RESIDUALS >= 0. : 1
RESIDUALS < 0. : 2
NUMBER OF RUNS : 17 IN 35 OBSERVATIONS

```
SUM OF SQUARED WEIGHTED RESIDUALS (ALL DEP. VAR.)   36.220
STATISTICS FOR ALL RESIDUALS :
AVERAGE WEIGHTED RESIDUAL  :-0.100E+00
# RESIDUALS >= 0. :       17
# RESIDUALS < 0.  :       18
NUMBER OF RUNS  :   17  IN   35 OBSERVATIONS
RUNS STATISTIC (TOO FEW RUNS):   -0.339
 (IF #NEG>10 AND #POS>10, P(STAT < -1.28) = 0.10,
                          P(STAT < -1.645) = 0.05,
                          P(STAT < -1.96) = 0.025)
RUNS STATISTIC (TOO MANY RUNS):   -0.682
 (IF #NEG>10 AND #POS>10, P(STAT > 1.28) = 0.10,
                          P(STAT > 1.645) = 0.05,
                          P(STAT > 1.96) = 0.025)
```

SCALED SENSITIVITIES (SCALED BY B*(WT**.5))

PARAMETER # :		1	2	3	4	5	6	7	8	9
PARAMETER ID :		Q	S1	T	KST	KV	S1	T	RCH	RCH
OBS# & ID										
1	1.0	0.000	0.000	-0.652E-04	1.17	-0.186E-06	0.000	-0.289E-04	0.716E-01	0.106
2	1.1	-0.171E-02	-0.360E-01	0.334E-01	-0.119E-01	-0.340E-02	-0.386E-02	0.159E-01	-0.216E-06	-0.320E-06
3	1.12	-1.68	-0.812	0.718	-11.1	-0.167E-02	-0.643E-01	0.341	0.000	-0.320E-06
4	2.0	0.000	0.000	180.	1.17	1.34	0.000	34.5	9.34	17.6
5	2.1	-0.674	-11.5	6.12	-0.211E-02	-1.06	-1.21	4.32	0.277E-04	0.000
6	2.2	-10.9	-118.	39.5	-0.108	-2.88	-10.2	26.5	0.554E-04	0.820E-04
7	2.8	-231.	-520.	-0.105E+04	-7.79	-5.95	-41.3	-128.	0.554E-04	0.164E-03
8	2.12	-284.	-133.	-0.177E+04	-11.0	-10.5	-10.6	-322.	0.554E-04	0.820E-04
9	3.0	0.000	0.000	351.	1.17	2.47	0.000	115.	15.0	42.0
10	3.1	-86.4	-407.	-104.	-0.122E-03	58.5	-36.6	-129.	0.000	0.164E-03
11	3.12	-852.	-295.	-0.474E+04	-10.6	42.6	-23.3	-0.195E+04	0.111E-03	0.164E-03
12	4.0	0.000	0.000	180.	1.17	1.34	0.000	34.5	9.34	17.6
13	4.1	-1.03	-16.2	8.22	-0.271E-02	-1.42	-1.70	5.71	0.277E-04	0.000
14	4.12	-285.	-133.	-0.178E+04	-10.9	-9.85	-10.6	-327.	0.554E-04	0.820E-04
15	5.0	0.000	0.000	263.	1.17	1.76	0.000	66.2	12.9	28.0
16	5.1	-1.97	-27.8	12.5	-0.523E-03	-2.46	-2.94	9.74	0.000	0.000
17	5.12	-433.	-207.	-0.260E+04	-10.7	-13.1	-16.4	-625.	0.554E-04	0.000
18	6.0	0.000	0.000	181.	1.17	4.89	0.000	33.3	9.33	17.8
19	6.1	-0.767	-12.4	6.30	-0.223E-02	-2.89	-1.59	5.94	0.277E-04	0.000
20	6.12	-287.	-135.	-0.179E+04	-11.0	-67.8	-10.7	-298.	0.554E-04	0.000
21	7.0	0.000	0.000	7.13	1.17	14.8	0.000	-8.56	0.451	0.747
22	7.1	-0.545E-01	-0.861	0.496	-0.447E-01	-0.791	-0.122	0.790	0.000	-0.256E-05
23	7.12	-13.0	-5.73	-71.4	-12.1	-162.	-0.456	94.4	0.000	-0.256E-05
24	8.0	0.000	0.000	350.	1.17	2.01	0.000	116.	14.9	42.1
25	8.1	-111.	-356.	-141.	-0.142E-03	-329.	-66.9	-116.	0.000	0.000
26	8.12	-862.	-295.	-0.465E+04	-10.6	-158.	-23.4	-0.203E+04	0.111E-03	0.164E-03
27	9.0	0.000	0.000	441.	1.17	-0.968	0.000	196.	15.0	61.7
28	9.1	-2.01	-26.7	9.33	0.118E-04	-12.2	-3.87	19.2	0.000	0.164E-03
29	9.12	-767.	-404.	-0.417E+04	-10.3	-33.7	-32.0	-0.162E+04	0.111E-03	0.164E-03
30	0.0	0.000	0.000	263.	1.17	3.57	0.000	65.9	12.8	28.2
31	0.1	-0.334	-6.50	3.48	-0.244E-03	-1.64	-0.833	3.67	0.000	0.000
32	0.12	-398.	-208.	-0.241E+04	-10.5	-54.7	-16.5	-492.	0.554E-04	0.000
33	SS	0.000	0.000	0.399E-02	0.109E-04	0.122E-04	0.000	0.177E-02	-4.38	-6.48
34	TR3	0.759	5.70	-5.61	-0.279E-01	0.268E-01	0.473	-1.62	-4.61	-6.83
35	TR12	9.99	4.73	-4.40	-0.131E-01	-0.331E-01	0.374	-1.63	-8.35	-12.4
((SUM OF THE SQUARED VALUES)**.5)/ND		0.471E+02	0.293E+02	0.264E+03	0.101E+01	0.119E+02	0.295E+01	0.974E+02	0.106E+01	0.287E+01

```
PARAMETER VALUES AND STATISTICS FOR ALL ITERATIONS
          PAR. ID.:
        Q          S1         T          KST        KV         S1
        T          RCH        RCH
                                                                            SUM OF SQUARED WEIGHTED RESIDUALS                    AMP OR
ITER                                                                         HEADS      W/FLOWS     W/PARAMS     DMX         PAR#  AGMX
     -0.110E+01 0.130E-02 0.300E-03 0.120E-02 0.100E-06 0.200E-03
      0.400E-04 0.200E-07 0.100E-07
  1                                                                         0.268E+06  0.268E+06  0.268E+06   0.859        5   0.000
     -0.100E+01 0.122E-02 0.393E-03 0.210E-03 0.186E-06 0.874E-04
      0.428E-04 0.124E-07 0.139E-07
  2                                                                         0.113E+04  0.113E+04  0.113E+04    1.43        4   0.000
     -0.106E+01 0.115E-02 0.423E-03 0.510E-03 0.215E-06 0.549E-04
      0.473E-04 0.110E-07 0.157E-07
  3                                                                          62.7       62.8       62.8       0.885        4   0.000
     -0.107E+01 0.114E-02 0.425E-03 0.962E-03 0.215E-06 0.633E-04
      0.482E-04 0.108E-07 0.160E-07
  4                                                                          40.3       40.4       40.4       0.339        4   0.000
     -0.107E+01 0.114E-02 0.426E-03 0.129E-02 0.215E-06 0.631E-04
      0.482E-04 0.108E-07 0.160E-07
  5                                                                          36.2       36.3       36.3       0.540E-01    4   0.000
     -0.107E+01 0.114E-02 0.426E-03 0.136E-02 0.215E-06 0.632E-04
      0.482E-04 0.108E-07 0.160E-07
  6                                                                          36.1       36.2       36.2       0.175E-02    4   0.100E-02
     -0.107E+01 0.114E-02 0.426E-03 0.136E-02 0.215E-06 0.632E-04
      0.482E-04 0.108E-07 0.160E-07
  7                                                                          36.1       36.2       36.2
```

COVARIANCE MAT.

	1	2	3	4	5	6	7	8	9
1	5.9319E-03	-5.5862E-03	-5.5080E-03	-5.3527E-03	-5.5753E-03	-4.8110E-03	-5.5623E-03	-1.3943E-11	-1.0984E-10
2	-5.5862E-03	7.4539E-03	5.1664E-03	-5.5827E-03	8.5119E-03	-2.5022E-02	5.2008E-03	1.2420E-11	1.0267E-10
3	-5.5080E-03	5.1664E-03	5.1564E-03	2.2975E-03	5.2591E-03	4.4799E-03	5.0570E-03	1.3355E-11	1.0182E-10
4	-5.3527E-03	-5.5827E-03	2.2975E-03	0.3361	-2.0936E-02	0.1599	1.1507E-02	7.6163E-12	1.0854E-10
5	-5.5753E-03	8.5119E-03	5.2591E-03	-2.0936E-02	1.3486E-02	-4.5061E-02	4.9276E-03	1.3502E-11	1.0156E-10
6	-4.8110E-03	-2.5022E-02	4.4799E-03	0.1599	-4.5061E-02	0.4300	5.4141E-03	1.7087E-11	9.9397E-11
7	-5.5623E-03	5.2008E-03	5.0570E-03	1.1507E-02	4.9276E-03	5.4141E-03	5.5212E-03	1.1806E-11	1.0366E-10
8	-1.3943E-11	1.2420E-11	1.3355E-11	7.6163E-12	1.3502E-11	1.7087E-11	1.1806E-11	1.4337E-18	-4.7146E-19
9	-1.0984E-10	1.0267E-10	1.0182E-10	1.0854E-10	1.0156E-10	9.9397E-11	1.0366E-10	-4.7146E-19	2.4504E-18

PARAMETER SUMMARY

```
PARAMETER # :       1         2         3         4         5         6         7         8         9
PARAMETER ID :      Q         S1        T         KST       KV        S1        T         RCH       RCH
FINAL VALUES
                 -0.107E+01-0.678E+01-0.776E+01-0.660E+01-0.154E+02-0.967E+01-0.994E+01 0.108E-07 0.160E-07
  EXPONENTIAL OF LN PARAMETERS
  (0.0 FOR UNTRANSFORMED PARAMETERS)
                  0.000E+00 0.114E-02 0.426E-03 0.136E-02 0.215E-06 0.632E-04 0.482E-04 0.000E+00 0.000E+00
STD. DEV.
                  0.770E-01 0.863E-01 0.718E-01 0.580E+00 0.116E+00 0.656E+00 0.743E-01 0.120E-08 0.157E-08
COEF. OF VAR.
                  0.717E-01 0.127E-01 0.925E-02 0.878E-01 0.756E-02 0.678E-01 0.748E-02 0.111E+00 0.978E-01

+ 2 STD. DEV.
                 -0.920E+00-0.661E+01-0.762E+01-0.544E+01-0.151E+02-0.836E+01-0.979E+01 0.132E-07 0.191E-07
  EXPONENTIAL OF LN PARAMETERS
  (0.0 FOR UNTRANSFORMED PARAMETERS)
                  0.000E+00 0.135E-02 0.491E-03 0.434E-02 0.271E-06 0.235E-03 0.560E-04 0.000E+00 0.000E+00

- 2 STD. DEV.
                 -0.123E+01-0.695E+01-0.791E+01-0.776E+01-0.156E+02-0.110E+02-0.101E+02 0.842E-08 0.129E-07
  EXPONENTIAL OF LN PARAMETERS
  (0.0 FOR UNTRANSFORMED PARAMETERS)
                  0.000E+00 0.957E-03 0.369E-03 0.427E-03 0.170E-06 0.170E-04 0.416E-04 0.000E+00 0.000E+00
```

CORRELATION MAT.

	1	2	3	4	5	6	7	8	9
1	1.000	-0.8401	-0.9959	-0.1199	-0.6233	-9.5264E-02	-0.9719	-0.1512	-0.9111
2	-0.8401	1.000	0.8333	-0.1115	0.8490	-0.4420	0.8107	0.1201	0.7597
3	-0.9959	0.8333	1.000	5.5186E-02	0.6307	9.5145E-02	0.9478	0.1553	0.9058
4	-0.1199	-0.1115	5.5186E-02	1.000	-0.3110	0.4207	0.2671	1.0972E-02	0.1196
5	-0.6233	0.8490	0.6307	-0.3110	1.000	-0.5918	0.5711	9.7099E-02	0.5587
6	-9.5264E-02	-0.4420	9.5145E-02	0.4207	-0.5918	1.000	0.1111	2.1763E-02	9.6838E-02
7	-0.9719	0.8107	0.9478	0.2671	0.5711	0.1111	1.000	0.1327	0.8912
8	-0.1512	0.1201	0.1553	1.0972E-02	9.7099E-02	2.1763E-02	0.1327	1.000	-0.2515
9	-0.9111	0.7597	0.9058	0.1196	0.5587	9.6838E-02	0.8912	-0.2515	1.000

THE CORRELATION OF THE FOLLOWING PARAMETER PAIRS >= .95

PARAMETER #	ID	#	ID	CORRELATION
1	Q	3	T	-1.00
1	Q	7	T	-0.97

THE CORRELATION OF THE FOLLOWING PARAMETER PAIRS IS BETWEEN .90 AND .95

PARAMETER #	ID	#	ID	CORRELATION
1	Q	9	RCH	-0.91
3	T	7	T	0.95
3	T	9	RCH	0.91

THE CORRELATION OF THE FOLLOWING PARAMETER PAIRS IS BETWEEN .85 AND .90

PARAMETER #	ID	#	ID	CORRELATION
7	T	9	RCH	0.89

```
RSQ (DEP.VAR. ONLY)------ =  36.220
RSQ (W/PARAMETERS)------- =  36.220
CALCULATED ERROR VARIANCE =  1.3931
CORRELATION COEFFICIENT-- = 0.99999
     W/PARAMETERS-------- = 0.99999
ITERATIONS--------------- =       6

MAX LIKE OBJ FUNC = -38.131
AIC STATISTIC---- = -20.131
BIC STATISTIC---- = -6.1326

ORDERED DEPENDENT-VARIABLE WEIGHTED RESIDUALS
NUMBER OF RESIDUALS INCLUDED:        35
 -2.25     -1.99     -1.62     -1.60     -1.12     -1.06     -1.02     -1.01
-0.997    -0.867    -0.774    -0.463    -0.237    -0.166    -0.745E-01-0.474E-01
-0.430E-01-0.323E-01 0.874E-01 0.109     0.132     0.146     0.262     0.312
 0.372     0.376     0.452     0.578     0.666     0.896     0.900     1.06
 1.24      2.10      2.18
CORRELATION BETWEEN ORDERED WEIGHTED RESIDUALS AND
INDEPENDENT NORMAL DEVIATES =    0.978
```

Test Case 2

The physical system for test case 2 is shown in figure A8. This test case represents the same ground-water system discussed by Cooley and Naff (1990, p. 79-81), and is repeated here partly for the benefit of users to whom it is familiar. The system is comprised of one confined aquifer that is divided into three zones. The system is bounded by no-flow, defined-flow (q_{b1} and q_{B2}, in units of L/T), and constant hydraulic-head (h_B) boundaries, and is affected by a river with a riverbed hydraulic conductance equal to R_2. The aquifer properties shown in figure A8 are the transmissivity values in zones 1, 2, and 3 (T_1, T_2, and T_3). Stresses on the system include areal recharge in zones 1, 2, and 3 (W_1, W_2, and W_3), and pumpage from two wells (Q_1 and Q_2). The system is at steady state.

For the finite-difference method, most of the system is discretized into square 1,000-ft^2 cells, with the following exceptions: along the river, the cells are 200 ft wide across the river; along the defined-flux and constant-head boundaries, the cells are 50 ft wide across the boundary. The two wells are located at the center of cells at row 7, column 11, and at row 9, column 12. The following 14 model parameters are estimated: q_{B1}, q_{B2}, Q_1, Q_2, h_{B1}, h_{B2}, and h_{B3} (located in fig. A8; the constant hydraulic head at intermediate points is calculated by linear interpolation), T_1, W_1, T_2, W_2, T_3, W_3, and R_2; the true values are shown in figure A8. The hydraulic-head data were produced by using the true parameter values, and then were corrupted by adding normally distributed noise with a standard deviation of 1.0 ft.

INPUT FILE and Input Files from Other Packages for Test Case 2

The INPUT FILE used for test case 2 is shown on the following pages. Note that two repetitions of DATA SET 12 are used to represent two of the nine prior parameter estimates. Because each of these prior estimates are associated only with one parameter, they also could have been represented with DATA SET 10. Use of DATA SET 12, however, enables the user to change the initial parameter values, as might be required to produce convergence of parameter estimation, without changing the prior estimate of the parameter (see section "Restarting the Parameter-Estimation Routine" of Appendix B).

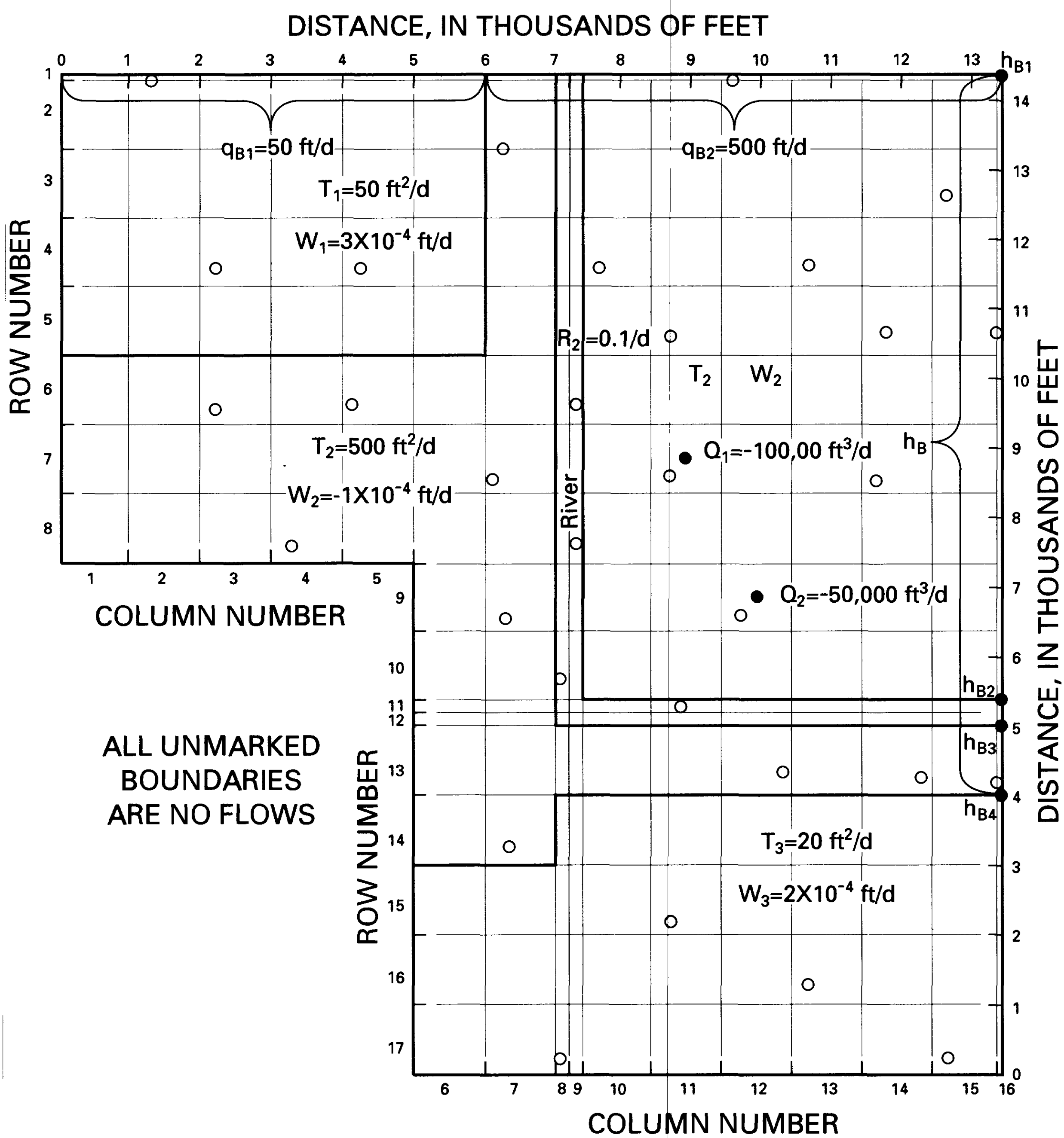


Figure A8.--Physical system for test case 2.

INPUT FILE

TEST CASE 2								LINE 1
								LINE 2
14	10	68	4	3	0	1		LINE 3
2	1	1						LINE 4
32	0	0	0	0	0	2		LINE 5
1	-1	0	0					LINE 6
36	13	0	0					LINE 7
0	0	0						LINE 8
Q	6	0	0	-1				DATA SET 2
	1		1		1		10.	DATA SET 2A
	1		1		2		10.	
	1		1		3		10.	
	1		1		4		10.	
	1		1		5		10.	
	1		1		6		10.	
Q	9	0	0	-1				DATA SET 2
	1		1		7		.56	DATA SET 2A
	1		1		8		.112	
	1		1		9		.112	
	1		1		10		.56	
	1		1		11		.56	
	1		1		12		.56	
	1		1		13		.56	
	1		1		14		.56	
	1		1		15		.56	
Q	1	0	0	-1				DATA SET 2
	1		7		11		1.0	DATA SET 2A
Q	1	0	0	-1				DATA SET 2
	1		9		12		1.0	DATA SET 2A
CH	13	3	0	1				DATA SET 2
	1		1		16		2.	DATA SET 2A
	1		2		16		0.	
	1		3		16		0.	
	1		4		16		0.	
	1		5		16		0.	
	1		6		16		0.	
	1		7		16		0.	
	1		8		16		0.	
	1		9		16		0.	
	1		10		16		0.	
	1		11		16		3.	
	1		12		16		3.	
	1		13		16		4.	

```
T   0     1                                                                  DATA SET 2
        1.    0     1     1     4                                            DATA SET 2B
RCH       1    0     0    -1                                                 DATA SET 2
     1.E-0    0     1     1                                                  DATA SET 2B
T   0     1                                                                  DATA SET 2
        1.    0     1     2     5                                            DATA SET 2B
RCH 0     2    0     0    -1                                                 DATA SET 2
     1.E-0    0     1     2                                                  DATA SET 2B
T   0     1                                                                  DATA SET 2
        1.    0     1     3     6                                            DATA SET 2B
RCH       3    0     0    -1                                                 DATA SET 2
     1.E-0    0     1     3                                                  DATA SET 2B
KRB      38    0     0    -1                                                 DATA SET 2
          1          1          8        4.5     10000.           -4.5       DATA SET 2A
          1          2          8        4.5    200000.           -4.5
          1          3          8        4.5    200000.           -4.5
          1          4          8        4.5    200000.           -4.5
          1          5          8        4.5    200000.           -4.5
          1          6          8        4.5    200000.           -4.5
          1          7          8        4.5    200000.           -4.5
          1          8          8        4.5    200000.           -4.5
          1          9          8        4.5    200000.           -4.5
          1         10          8        4.5    200000.           -4.5
          1         11          8        4.5     40000.           -4.5
          1         12          8        4.5     40000.           -4.5
          1         12          9        4.5     40000.           -4.5
          1         12         10        4.5    200000.           -4.5
          1         12         11        4.5    200000.           -4.5
          1         12         12        4.5    200000.           -4.5
          1         12         13        4.5    200000.           -4.5
          1         12         14        4.5    200000.           -4.5
          1         12         15        4.5    200000.           -4.5
          1         12         16        4.5     10000.           -4.5
          1          1          9        4.5     10000.           -4.5
          1          2          9        4.5    200000.           -4.5
          1          3          9        4.5    200000.           -4.5
          1          4          9        4.5    200000.           -4.5
          1          5          9        4.5    200000.           -4.5
          1          6          9        4.5    200000.           -4.5
          1          7          9        4.5    200000.           -4.5
          1          8          9        4.5    200000.           -4.5
          1          9          9        4.5    200000.           -4.5
          1         10          9        4.5    200000.           -4.5
          1         11          9        4.5     40000.           -4.5
          1         11         10        4.5    200000.           -4.5
          1         11         11        4.5    200000.           -4.5
          1         11         12        4.5    200000.           -4.5
          1         11         13        4.5    200000.           -4.5
          1         11         14        4.5    200000.           -4.5
          1         11         15        4.5    200000.           -4.5
          1         11         16        4.5     10000.           -4.5
```

```
   16        1(8I8)                                        DATA SET 4
  4       4       4       4       4       4       5       5
  5       5       5       5       5       5       5       5
  1       1       1       1       1       1       2       5
  5       2       2       2       2       2       2       5
  1       1       1       1       1       1       2       5
  5       2       2       2       2       2       2       5
  1       1       1       1       1       1       2       5
  5       2       2       2       2       2       2       5
  1       1       1       1       1       1       2       5
  5       2       2       2       2       2       2       5
  2       2       2       2       2       2       2       5   5
  5       2       2       2       2       2       2       5
  2       2       2       2       2       2       2       5
  5       2       2       2       2       2       2       5
  2       2       2       2       2       2       2       5
  5       2       2       2       2       2       2       5
  0       0       0       0       0       2       2       5
  5       2       2       2       2       2       2       5
  0       0       0       0       0       2       2       5
  5       2       2       2       2       2       2       5
  0       0       0       0       0       2       2       5  10
  5       5       5       5       5       5       5       5
  0       0       0       0       0       2       2       5
  5       5       5       5       5       5       5       5
  0       0       0       0       0       2       2       2
  2       2       2       2       2       2       2       5
  0       0       0       0       0       2       2       3
  3       3       3       3       3       3       3       6
  0       0       0       0       0       3       3       3
  3       3       3       3       3       3       3       6
  0       0       0       0       0       3       3       3
  3       3       3       3       3       3       3       6
  0       0       0       0       0       3       3       3   30
  3       3       3       3       3       3       3       6
  1.0       0.0   0   0    1.0                                  DATA SET 5
 1   1  17   8   0  0.20  -0.30  0.00   69.22  1.00  0  DATA SET 6
 2   1  17  15   0  0.20  -0.30  0.00   80.58  1.00  0  DATA SET 6
 3   1  16  13   0  0.20  -0.30  0.00   72.66  1.00  0  DATA SET 6
 4   1  15  11   0  0.20  -0.30  0.00   49.55  1.00  0  DATA SET 6
 5   1  14   7   0  0.20  -0.20  0.00   10.04  1.00  0  DATA SET 6
 6   1  13  12   0  0.20   0.40  0.00    7.66  1.00  0  DATA SET 6
 7   1  13  14   0  0.30   0.40  0.00   11.44  1.00  0  DATA SET 6
 8   1  13  16   0  0.40   0.30  0.00   13.50  1.00  0  DATA SET 6
 9   1  11  11   0  0.20  -0.30  0.00    4.60  1.00  0  DATA SET 6
10   1  10   8   0  0.20  -0.30  0.00    5.30  1.00  0  DATA SET 6
11   1   9  12   0  0.20  -0.30  0.00  -38.73  1.00  0  DATA SET 6
12   1  10  16   0  0.20  -0.30  0.00    6.62  1.00  0  DATA SET 6
13   1   9   7   0  0.20  -0.30  0.00    6.54  1.00  0  DATA SET 6
14   1   8   4   0  0.20  -0.30  0.00   12.19  1.00  0  DATA SET 6
15   1   8   9   0  0.20  -0.30  0.00    4.40  1.00  0  DATA SET 6
16   1   7  11   0  0.20  -0.30  0.00  -61.07  1.00  0  DATA SET 6
17   1   7   7   0  0.20  -0.30  0.00    6.44  1.00  0  DATA SET 6
```

```
18      1    7   14    0    0.20   -0.30    0.00    -14.03    1.00    0  DATA SET 6
19      1    6    3    0    0.20   -0.30    0.00     15.73    1.00    0  DATA SET 6
20      1    6    5    0    0.20   -0.30    0.00     14.89    1.00    0  DATA SET 6
21      1    6    9    0    0.20   -0.30    0.00      4.30    1.00    0  DATA SET 6
22      1    5   11    0    0.20   -0.30    0.00    -16.15    1.00    0  DATA SET 6
23      1    5   14    0    0.20   -0.30    0.00     -5.37    1.00    0  DATA SET 6
24      1    5   16    0    0.20   -0.30    0.00      7.59    1.00    0  DATA SET 6
25      1    4    3    0    0.20   -0.30    0.00     42.42    1.00    0  DATA SET 6
26      1    4    5    0    0.20   -0.30    0.00     29.08    1.00    0  DATA SET 6
27      1    4   10    0    0.20   -0.30    0.00      1.47    1.00    0  DATA SET 6
28      1    4   13    0    0.20   -0.30    0.00     -7.42    1.00    0  DATA SET 6
29      1    3    7    0   -0.50    0.20    0.00      6.58    1.00    0  DATA SET 6
30      1    3   15    0    0.20   -0.30    0.00      4.02    1.00    0  DATA SET 6
31      1    1    2    0    0.20   -0.30    0.00     85.49    1.00    0  DATA SET 6
32      1    1   12    0    0.20   -0.30    0.00      0.98    1.00    0  DATA SET 6
     45.      420.   -97000.   -51000.      10.4       4.8       5.1       70.
  .00040      420.   -.00020       15.    .00017      .080                 DATA SET 8
       1         1         2         2         3         3         3         4
       5         4         5         4         5         6                 DATA SET 9
      0.        0.        0.        0.      1.04       .48       .51        0.
  .00012       84.        0.        0.   .000051      .008                 DATA SET 10
       1                                                                   DATA SET 11
      0.        0.        1.        0.        0.        0.        0.        0. DATA
      0.        0.        0.        0.        0.        0.   -97000.     1940. SET 12
      0.        0.        0.        1.        0.        0.        0.        0.
      0.        0.        0.        0.        0.        0.   -51000.     1020.
2.0  .08 .001  100   0.    1   35    2    0    0    2    0    0  .00  .00 DATA SET 13
```

Input files from other packages used without and with the Parameter-Estimation Package are shown on the following pages. The true model input values are included in these files.

Basic Package Input Files

Without the Parameter-Estimation Package

CLASS PROBLEM

```
         1        17        16         1         4
  7  8  0 15  0  0  0 31  0  0  0 11  9  0  0
         0         0
        10         1(16I3)                      -1
   99999.
        12        1.(10F8.0)                    -1
   86400.          1         1
```

With the Parameter-Estimation Package

CLASS PROBLEM - PARAMETER ESTIMATION

```
         1        17        16         0         4
  7  8  0 15  0  0  0 31  0  0  0 11  9  0 16
         0         0
        10         1(16I3)                      -1
   99999.
        12        1.(10F8.0)                    -1
   86400.          1         1
```

File 10 used by the Basic Package

```
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1
  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1 -1
  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1 -1
  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1 -1
  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1 -1
  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1 -1
  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1 -1
  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1
  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1
  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1
  0  0  0  0  0  1  1  1  1  1  1  1  1  1  1  1
```

File 12 used by the Basic Package

```
5.      5.      5.      5.      5.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5.    10.0
5.      5.      5.      5.      5.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5. 9.71233
5.      5.      5.      5.      5.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5. 9.16438
5.      5.      5.      5.      5.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5. 8.61644
5.      5.      5.      5.      5.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5. 8.06849
5.      5.      5.      5.      5.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5. 7.52055
5.      5.      5.      5.      5.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5. 6.97260
5.      5.      5.      5.      5.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5. 6.42466
0.      0.      0.      0.      0.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5. 5.87671
0.      0.      0.      0.      0.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5. 5.32877
0.      0.      0.      0.      0.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5.     5.0
0.      0.      0.      0.      0.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5.     5.0
0.      0.      0.      0.      0.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5.     5.5
0.      0.      0.      0.      0.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5.      5.
0.      0.      0.      0.      0.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5.      5.
0.      0.      0.      0.      0.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5.      5.
0.      0.      0.      0.      0.      5.      5.      5.      5.      5.
5.      5.      5.      5.      5.      5.
```

Output Control Package Input File
(not changed when using the Parameter-Estimation Package)

```
2    2    33    0
0    1     1    0
1    0     1    0
```

Block-Centered Flow Package Input File
(not changed when using the Parameter-Estimation Package)

```
         1         0
0 0
         0         1.
         7         1.(8F8.0)
   1000.   1000.   1000.   1000.   1000.   1000.   1000.    200.
    200.   1000.   1000.   1000.   1000.   1000.   1000.     50.
         7         1.(8F8.0)
     50.   1000.   1000.   1000.   1000.   1000.   1000.   1000.
   1000.   1000.    200.    200.   1000.   1000.   1000.   1000.
   1000.
         7         1.(8F8.0)
     50.     50.     50.     50.     50.     50.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
     50.     50.     50.     50.     50.     50.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
     50.     50.     50.     50.     50.     50.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
     50.     50.     50.     50.     50.     50.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
     50.     50.     50.     50.     50.     50.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
      0.      0.      0.      0.      0.    500.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
      0.      0.      0.      0.      0.    500.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
      0.      0.      0.      0.      0.    500.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
      0.      0.      0.      0.      0.    500.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
      0.      0.      0.      0.      0.    500.    500.    500.
    500.    500.    500.    500.    500.    500.    500.    500.
      0.      0.      0.      0.      0.    500.    500.     20.
     20.     20.     20.     20.     20.     20.     20.     20.
      0.      0.      0.      0.      0.     20.     20.     20.
     20.     20.     20.     20.     20.     20.     20.     20.
      0.      0.      0.      0.      0.     20.     20.     20.
     20.     20.     20.     20.     20.     20.     20.     20.
      0.      0.      0.      0.      0.     20.     20.     20.
     20.     20.     20.     20.     20.     20.     20.     20.
```

River Package Input File
(not changed when using the Parameter-Estimation Package)

38	0				
38					
1	1	8	4.5	1000.	-4.5
1	2	8	4.5	20000.	-4.5
1	3	8	4.5	20000.	-4.5
1	4	8	4.5	20000.	-4.5
1	5	8	4.5	20000.	-4.5
1	6	8	4.5	20000.	-4.5
1	7	8	4.5	20000.	-4.5
1	8	8	4.5	20000.	-4.5
1	9	8	4.5	20000.	-4.5
1	10	8	4.5	20000.	-4.5
1	11	8	4.5	4000.	-4.5
1	12	8	4.5	4000.	-4.5
1	12	9	4.5	4000.	-4.5
1	12	10	4.5	20000.	-4.5
1	12	11	4.5	20000.	-4.5
1	12	12	4.5	20000.	-4.5
1	12	13	4.5	20000.	-4.5
1	12	14	4.5	20000.	-4.5
1	12	15	4.5	20000.	-4.5
1	12	16	4.5	1000.	-4.5
1	1	9	4.5	1000.	-4.5
1	2	9	4.5	20000.	-4.5
1	3	9	4.5	20000.	-4.5
1	4	9	4.5	20000.	-4.5
1	5	9	4.5	20000.	-4.5
1	6	9	4.5	20000.	-4.5
1	7	9	4.5	20000.	-4.5
1	8	9	4.5	20000.	-4.5
1	9	9	4.5	20000.	-4.5
1	10	9	4.5	20000.	-4.5
1	11	9	4.5	4000.	-4.5
1	11	10	4.5	20000.	-4.5
1	11	11	4.5	20000.	-4.5
1	11	12	4.5	20000.	-4.5
1	11	13	4.5	20000.	-4.5
1	11	14	4.5	20000.	-4.5
1	11	15	4.5	20000.	-4.5
1	11	16	4.5	1000.	-4.5

Recharge Package Input File
(not changed when using the Parameter-Estimation Package)

```
         1         0     NRCHOP   IRCHBD
         0         0
        31      1.E-4(8F8.0)                        2
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
     3.0     3.0     3.0     3.0     3.0     3.0    -1.0     0.0
    -0.0    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0     0.0
     3.0     3.0     3.0     3.0     3.0     3.0    -1.0     0.0
    -0.0    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0     0.0
     3.0     3.0     3.0     3.0     3.0     3.0    -1.0     0.0
    -0.0    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0     0.0
     3.0     3.0     3.0     3.0     3.0     3.0    -1.0    -0.0
    -0.0    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0     0.0
    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0    -0.0    5
    -0.0    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0     0.0
    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0    -0.0
    -0.0    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0     0.0
    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0    -0.0
    -0.0    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0     0.0
      0.      0.      0.      0.      0.    -1.0    -1.0    -0.0
    -0.0    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0     0.0
      0.      0.      0.      0.      0.    -1.0    -1.0    -0.0
    -0.0    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0     0.0
      0.      0.      0.      0.      0.    -1.0    -1.0    -0.0   10
    -0.0    -0.0    -0.0    -0.0    -0.0    -0.0    -0.0     0.0
      0.      0.      0.      0.      0.    -1.0    -1.0    -0.0
    -0.0    -0.0    -0.0    -0.0    -0.0    -0.0    -0.0     0.0
      0.      0.      0.      0.      0.    -1.0    -1.0    -1.0
    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0    -1.0     0.0
      0.      0.      0.      0.      0.    -1.0    -1.0     2.0
     2.0     2.0     2.0     2.0     2.0     2.0     2.0     0.0
      0.      0.      0.      0.      0.     2.0     2.0     2.0
     2.0     2.0     2.0     2.0     2.0     2.0     2.0     0.0
      0.      0.      0.      0.      0.     2.0     2.0     2.0
     2.0     2.0     2.0     2.0     2.0     2.0     2.0     0.0
      0.      0.      0.      0.      0.     2.0     2.0     2.0   16
     2.0     2.0     2.0     2.0     2.0     2.0     2.0     0.0
```

Well Package Input File

(not changed when using the Parameter-Estimation Package)

17	0		
17			
1	1	1	500.
1	1	2	500.
1	1	3	500.
1	1	4	500.
1	1	5	500.
1	1	6	500.
1	1	7	280.
1	1	8	56.
1	1	9	56.
1	1	10	280.
1	1	11	280.
1	1	12	280.
1	1	13	280.
1	1	14	280.
1	1	15	280.
1	7	11	-100000.
1	9	12	-50000.

PCG2 Package Input File

(not changed when using the Parameter-Estimation Package)

1	40	1			
.00001	.00001	1.	2	1	3

Output for Test Case 2

```
1                    U.S. GEOLOGICAL SURVEY MODULAR FINITE-DIFFERENCE GROUND-WATER MODEL - MODFLOWP VERSION
0CLASS PROBLEM - PARAMETER ESTIMATION
    1 LAYERS         17 ROWS         16 COLUMNS
   0 STRESS PERIOD(S) IN SIMULATION
 MODEL TIME UNIT IS DAYS
0I/O UNITS:
 ELEMENT OF IUNIT:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
        I/O UNIT:  7  8  0 15  0  0  0 31  0  0  0 11  9  0 16  0  0  0  0  0  0  0  0  0
0BAS1 -- BASIC MODEL PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM UNIT  5
 ARRAYS RHS AND BUFF WILL SHARE MEMORY.
 START HEAD WILL NOT BE SAVED -- DRAWDOWN CANNOT BE CALCULATED
      2213 ELEMENTS IN X ARRAY ARE USED BY BAS
      2213 ELEMENTS OF X ARRAY USED OUT OF  900000
0BCF1 -- BLOCK-CENTERED FLOW PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM UNIT  7
 STEADY-STATE SIMULATION
       LAYER  AQUIFER TYPE
       ------------------
          1         0
         1 ELEMENTS IN X ARRAY ARE USED BY BCF
      2214 ELEMENTS OF X ARRAY USED OUT OF  900000
0WEL1 -- WELL PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM  8
 MAXIMUM OF    17 WELLS
        68 ELEMENTS IN X ARRAY ARE USED FOR WELLS
      2282 ELEMENTS OF X ARRAY USED OUT OF  900000
0RCH1 -- RECHARGE PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM UNIT 31
 OPTION 1 -- RECHARGE TO TOP LAYER
       272 ELEMENTS OF X ARRAY USED FOR RECHARGE
      2554 ELEMENTS OF X ARRAY USED OUT OF  900000
0RIV1 -- RIVER PACKAGE, VERSION 1, 9/1/87 INPUT READ FROM UNIT 15
 MAXIMUM OF    38 RIVER NODES
       228 ELEMENTS IN X ARRAY ARE USED FOR RIVERS
      2782 ELEMENTS OF X ARRAY USED OUT OF  900000
0PCG2 -- CONJUGATE GRADIENT SOLUTION PACKAGE, VERSION 2, 5/1/88
 MAXIMUM OF    1 CALLS OF SOLUTION ROUTINE
 MAXIMUM OF   40 INTERNAL ITERATIONS PER CALL TO SOLUTION ROUTINE
 MATRIX PRECONDITIONING TYPE :     1
     1408 ELEMENTS IN X ARRAY ARE USED BY PCG
     4190 ELEMENTS OF X ARRAY USED OUT OF 900000
0SEN1 -- SENSITIVITY PACKAGE, VERSION 1, 12/01/91 INPUT READ FROM UNIT 16
0CLASS PROBLEM
0
 NUMBER OF PARAMETERS...............................:    14
 NUMBER NLL>0 IN ALL DATA SETS 2 FOR PID=T,KV,
  TKV, AND S1, +1 FOR NLL(1)>0 FOR PID=RCH OR ETM...:    10
 SUM ALL |NLL(1)| FOR PID=Q,KRB,KDR,KST,GHB,AND CH..:    68
 LARGEST NLL VECTOR IN ALL DATA SETS 2..............:     4
 LARGEST LZ VECTOR IN ALL DATA SETS 2B..............:     3
 NUMBER OF MULTIPLICATION MATRICES IN DATA SET 3....:     0
 NUMBER OF ZONE MATRICES IN DATA SET 4..............:     1

 PRINT MATRICES PRODUCED BY INIT. PARAM. EST. (>0)..:     2
 CHECK CELL REPETITIONS FOR MATRIX PARAMETERS (>0)..:     0
   NUMBER OF CELL REPETITIONS ALLOWED...............:     0

 NUMBER OF HEADS....................................:    32
   NUMBER OF MULTILAYER HEADS.......................:     0
   MAXIMUM NUMBER OF LAYERS FOR MULTILAYER HEADS....:     0
 NUMBER OF FLUX CELL GROUPS.........................:     0
   NUMBER OF CELLS IN FLUX CELL GROUPS..............:     0
   NUMBER OF FLUXES.................................:     0
 NUMBER OF PRIORS WITH MULTIPLE PARAMETERS..........:     2

 HEADS (<0), HEADS AND SENS. (0), OR PAR. EST. (1)..:     1
 ADJOINT (+) OR SEN. EQ. (-) CALCULATIONS...........:    -1
   (GRADIENT SEARCH (+) OR GAUSS-NEWTON (-) REGRESSION)
 SCALE PRINTED SENSITIVITIES (>0)...................:     0
 ADJUST GAUSS-NEWTON MATRIX WITH NEWTON UPDATES.....:     0

 OUTPUT/INPUT UNIT FOR HEADS OR SENSITIVITIES.......:    36
 OUTPUT/INPUT UNIT NUMBER FOR INFO FROM ITERATIONS..:    13
 INPUT UNIT FOR SS/TR IBOUND ARRAYS.................:     0
 INPUT UNIT FOR KNOWN HEADS AT CONSTANT-HEAD BNDRYS.:     0

 NUMBER OF INPUT UNITS IDENTIFIED IN RCH AND EVT....:     0
 NUMBER OF TIME STEPS>0 AT WHICH HEAD=0.............:     0
 NUMBER OF USUALLY POS. PARAMETERS THAT MAY BE NEG..:     0
```

2718 ELEMENTS IN X ARRAY ARE USED FOR SENSITIVITIES
6908 ELEMENTS OF X ARRAY USED OUT OF 900000
PAR1 -- PARAMETER ESTIMATION PACKAGE, VERSION 1, 12/01/91
816 ELEMENTS IN X ARRAY ARE USED FOR PARAMETER ESTIMATION
7724 ELEMENTS OF X ARRAY USED OUT OF 900000
CLASS PROBLEM - PARAMETER ESTIMATION

BOUNDARY ARRAY FOR LAYER 1 WILL BE READ ON UNIT 5 USING FORMAT: (16I3)

AQUIFER HEAD WILL BE SET TO 99999. AT ALL NO-FLOW NODES (IBOUND=0).

INITIAL HEAD FOR LAYER 1 WILL BE READ ON UNIT 5 USING FORMAT: (10F8.0)

HEAD PRINT FORMAT IS FORMAT NUMBER 2 DRAWDOWN PRINT FORMAT IS FORMAT NUMBER 2
HEADS WILL BE SAVED ON UNIT 33 DRAWDOWNS WILL BE SAVED ON UNIT 0
OUTPUT CONTROL IS SPECIFIED EVERY TIME STEP

COLUMN TO ROW ANISOTROPY = 1.000000

DELR WILL BE READ ON UNIT 7 USING FORMAT: (8F8.0)

1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	200.00	200.00	1000.0
1000.0	1000.0	1000.0	1000.0	1000.0	50.000				

DELC WILL BE READ ON UNIT 7 USING FORMAT: (8F8.0)

50.000	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0	1000.0
200.00	200.00	1000.0	1000.0	1000.0	1000.0	1000.0			

TRANSMIS. ALONG ROWS FOR LAYER 1 WILL BE READ ON UNIT 7 USING FORMAT: (8F8.0)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	50.00	50.00	50.00	50.00	50.00	50.00	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
2	50.00	50.00	50.00	50.00	50.00	50.00	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
3	50.00	50.00	50.00	50.00	50.00	50.00	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
4	50.00	50.00	50.00	50.00	50.00	50.00	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
5	50.00	50.00	50.00	50.00	50.00	50.00	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
6	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
7	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
8	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
9	0.0000	0.0000	0.0000	0.0000	0.0000	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
10	0.0000	0.0000	0.0000	0.0000	0.0000	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
11	0.0000	0.0000	0.0000	0.0000	0.0000	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
12	0.0000	0.0000	0.0000	0.0000	0.0000	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
13	0.0000	0.0000	0.0000	0.0000	0.0000	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0	500.0
14	0.0000	0.0000	0.0000	0.0000	0.0000	500.0	500.0	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00
15	0.0000	0.0000	0.0000	0.0000	0.0000	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

16	0.0000	0.0000	0.0000	0.0000	0.0000	20.00	20.00	20.00	20.00	20.00
	20.00	20.00	20.00	20.00	20.00	20.00				
17	0.0000	0.0000	0.0000	0.0000	0.0000	20.00	20.00	20.00	20.00	20.00
	20.00	20.00	20.00	20.00	20.00	20.00				

SOLUTION BY THE CONJUGATE-GRADIENT METHOD

```
                               MAXIMUM NUMBER OF CALLS TO PCG ROUTINE =        1
                                 MAXIMUM ITERATIONS PER CALL TO PCG =       40
                                        MATRIX PRECONDITIONING TYPE =        1
               RELAXATION FACTOR (ONLY USED WITH PRECOND. TYPE 1) =   0.10000E+01
     PARAMETER OF POLYNOMIAL PRECOND. = 2 (2) OR IS CALCULATED :        2
                                 HEAD CHANGE CRITERION FOR CLOSURE =   0.10000E-04
                             RESIDUAL CHANGE CRITERION FOR CLOSURE =   0.10000E-04
                      PCG HEAD AND RESIDUAL CHANGE PRINTOUT INTERVAL =        1
          PRINTING FROM SOLVER IS LIMITED(1) OR SUPPRESSED (>1) =        3
```

DEFINITION OF TEMPORAL AND SPATIAL EXTENT OF PARAMETERS (PARAMETERIZATION)

```
PARAMETER NUMBER:    1, PARAMETER IDENTIFIER:  Q
LN:  0, VECTOR NLL:    6    0    0   -1
    LAYER        ROW    COLUMN    FACTOR
      1.          1.        1.    0.100E+02
      1.          1.        2.    0.100E+02
      1.          1.        3.    0.100E+02
      1.          1.        4.    0.100E+02
      1.          1.        5.    0.100E+02
      1.          1.        6.    0.100E+02

PARAMETER NUMBER:    2, PARAMETER IDENTIFIER:  Q
LN:  0, VECTOR NLL:    9    0    0   -1
    LAYER        ROW    COLUMN    FACTOR
      1.          1.        7.    0.560E+00
      1.          1.        8.    0.112E+00
      1.          1.        9.    0.112E+00
      1.          1.       10.    0.560E+00
      1.          1.       11.    0.560E+00
      1.          1.       12.    0.560E+00
      1.          1.       13.    0.560E+00
      1.          1.       14.    0.560E+00
      1.          1.       15.    0.560E+00

PARAMETER NUMBER:    3, PARAMETER IDENTIFIER:  Q
LN:  0, VECTOR NLL:    1    0    0   -1
    LAYER        ROW    COLUMN    FACTOR
      1.          7.       11.    0.100E+01

PARAMETER NUMBER:    4, PARAMETER IDENTIFIER:  Q
LN:  0, VECTOR NLL:    1    0    0   -1
    LAYER        ROW    COLUMN    FACTOR
      1.          9.       12.    0.100E+01

PARAMETER NUMBER:    5, PARAMETER IDENTIFIER:  CH
LN:  0, VECTOR NLL:   13    3    0    1
    LAYER        ROW    COLUMN    FACTOR
      0.          1.       16.    0.200E+01
      0.          2.       16.    0.000E+00
      0.          3.       16.    0.000E+00
      0.          4.       16.    0.000E+00
      0.          5.       16.    0.000E+00
      0.          6.       16.    0.000E+00
      0.          7.       16.    0.000E+00
      0.          8.       16.    0.000E+00
      0.          9.       16.    0.000E+00
      0.         10.       16.    0.000E+00
      0.         11.       16.    0.300E+01
      0.         12.       16.    0.300E+01
      0.         13.       16.    0.400E+01

PARAMETER NUMBER:    8, PARAMETER IDENTIFIER:  T
LN:  0, VECTOR NLL:    1    0    0    0
SFAC=   1.00     , MULT. MATRIX    0, ZONE MATRIX    1
ZONES =    1    4    0
```

```
PARAMETER NUMBER:    9, PARAMETER IDENTIFIER:  RCH
LN:  0, VECTOR NLL:    1    0    0   -1
SFAC=   1.00     , MULT. MATRIX    0, ZONE MATRIX    1
ZONES =    1    0    0

PARAMETER NUMBER:   10, PARAMETER IDENTIFIER:  T
LN:  0, VECTOR NLL:    1    0    0    0
SFAC=   1.00     , MULT. MATRIX    0, ZONE MATRIX    1
ZONES =    2    5    0

PARAMETER NUMBER:   11, PARAMETER IDENTIFIER:  RCH
LN:  0, VECTOR NLL:    2    0    0   -1
SFAC=   1.00     , MULT. MATRIX    0, ZONE MATRIX    1
ZONES =    2    0    0

PARAMETER NUMBER:   12, PARAMETER IDENTIFIER:  T
LN:  0, VECTOR NLL:    1    0    0    0
SFAC=   1.00     , MULT. MATRIX    0, ZONE MATRIX    1
ZONES =    3    6    0

PARAMETER NUMBER:   13, PARAMETER IDENTIFIER:  RCH
LN:  0, VECTOR NLL:    3    0    0   -1
SFAC=   1.00     , MULT. MATRIX    0, ZONE MATRIX    1
ZONES =    3    0    0

PARAMETER NUMBER:   14, PARAMETER IDENTIFIER:  KRB
LN:  0, VECTOR NLL:   38    0    0   -1
    LAYER       ROW     COLUMN     FACTOR
      1.         1.       8.     0.100E+05
      1.         2.       8.     0.200E+06
      1.         3.       8.     0.200E+06
      1.         4.       8.     0.200E+06
      1.         5.       8.     0.200E+06
      1.         6.       8.     0.200E+06
      1.         7.       8.     0.200E+06
      1.         8.       8.     0.200E+06
      1.         9.       8.     0.200E+06
      1.        10.       8.     0.200E+06
      1.        11.       8.     0.400E+05
      1.        12.       8.     0.400E+05
      1.        12.       9.     0.400E+05
      1.        12.      10.     0.200E+06
      1.        12.      11.     0.200E+06
      1.        12.      12.     0.200E+06
      1.        12.      13.     0.200E+06
      1.        12.      14.     0.200E+06
      1.        12.      15.     0.200E+06
      1.        12.      16.     0.100E+05
      1.         1.       9.     0.100E+05
      1.         2.       9.     0.200E+06
      1.         3.       9.     0.200E+06
      1.         4.       9.     0.200E+06
      1.         5.       9.     0.200E+06
      1.         6.       9.     0.200E+06
      1.         7.       9.     0.200E+06
      1.         8.       9.     0.200E+06
      1.         9.       9.     0.200E+06
      1.        10.       9.     0.200E+06
      1.        11.       9.     0.400E+05
      1.        11.      10.     0.200E+06
      1.        11.      11.     0.200E+06
      1.        11.      12.     0.200E+06
      1.        11.      13.     0.200E+06
      1.        11.      14.     0.200E+06
      1.        11.      15.     0.200E+06
      1.        11.      16.     0.100E+05
```

ZONE MATRIX WILL BE READ ON UNIT 16 USING FORMAT: (8I8)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	4	4	4	4	4	4	5	5	5	5	5	5	5	5	5	5
2	1	1	1	1	1	1	2	5	5	2	2	2	2	2	2	5
3	1	1	1	1	1	1	2	5	5	2	2	2	2	2	2	5
4	1	1	1	1	1	1	2	5	5	2	2	2	2	2	2	5
5	1	1	1	1	1	1	2	5	5	2	2	2	2	2	2	5
6	2	2	2	2	2	2	2	5	5	2	2	2	2	2	2	5
7	2	2	2	2	2	2	2	5	5	2	2	2	2	2	2	5
8	2	2	2	2	2	2	2	5	5	2	2	2	2	2	2	5
9	0	0	0	0	0	2	2	5	5	2	2	2	2	2	2	5
10	0	0	0	0	0	2	2	5	5	2	2	2	2	2	2	5
11	0	0	0	0	0	2	2	5	5	5	5	5	5	5	5	5
12	0	0	0	0	0	2	2	5	5	5	5	5	5	5	5	5
13	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	5
14	0	0	0	0	0	2	2	3	3	3	3	3	3	3	3	6
15	0	0	0	0	0	3	3	3	3	3	3	3	3	3	3	6
16	0	0	0	0	0	3	3	3	3	3	3	3	3	3	3	6
17	0	0	0	0	0	3	3	3	3	3	3	3	3	3	3	6

NSM CAN BE REDUCED FROM 10 TO 6

INPUT ERROR VARIANCES FOR HEADS AND FLOWS: 1.000 0.0000
INPUT UNIT NUMBERS FOR HEADS AND FLOWS : 0 0
ESTIMATED COMMON ERROR VARIANCE : 1.000

OBSERVED HEAD DATA

OBS#	ID	LAYER	ROW	COLUMN	TIME STEP	ROW/COLUMN/TIME OFFSETS			OBSERVATION	VARIANCE
1	1	1	17	8	0	0.20	-0.30	0.00	69.220	1.0000
2	2	1	17	15	0	0.20	-0.30	0.00	80.580	1.0000
3	3	1	16	13	0	0.20	-0.30	0.00	72.660	1.0000
4	4	1	15	11	0	0.20	-0.30	0.00	49.550	1.0000
5	5	1	14	7	0	0.20	-0.20	0.00	10.040	1.0000
6	6	1	13	12	0	0.20	0.40	0.00	7.6600	1.0000
7	7	1	13	14	0	0.30	0.40	0.00	11.440	1.0000
8	8	1	13	16	0	0.40	0.30	0.00	13.500	1.0000
9	9	1	11	11	0	0.20	-0.30	0.00	4.6000	1.0000
10	10	1	10	8	0	0.20	-0.30	0.00	5.3000	1.0000
11	11	1	9	12	0	0.20	-0.30	0.00	-38.730	1.0000
12	12	1	10	16	0	0.20	-0.30	0.00	6.6200	1.0000
13	13	1	9	7	0	0.20	-0.30	0.00	6.5400	1.0000
14	14	1	8	4	0	0.20	-0.30	0.00	12.190	1.0000
15	15	1	8	9	0	0.20	-0.30	0.00	4.4000	1.0000
16	16	1	7	11	0	0.20	-0.30	0.00	-61.070	1.0000
17	17	1	7	7	0	0.20	-0.30	0.00	6.4400	1.0000
18	18	1	7	14	0	0.20	-0.30	0.00	-14.030	1.0000
19	19	1	6	3	0	0.20	-0.30	0.00	15.730	1.0000
20	20	1	6	5	0	0.20	-0.30	0.00	14.890	1.0000
21	21	1	6	9	0	0.20	-0.30	0.00	4.3000	1.0000
22	22	1	5	11	0	0.20	-0.30	0.00	-16.150	1.0000
23	23	1	5	14	0	0.20	-0.30	0.00	-5.3700	1.0000
24	24	1	5	16	0	0.20	-0.30	0.00	7.5900	1.0000
25	25	1	4	3	0	0.20	-0.30	0.00	42.420	1.0000
26	26	1	4	5	0	0.20	-0.30	0.00	29.080	1.0000
27	27	1	4	10	0	0.20	-0.30	0.00	1.4700	1.0000
28	28	1	4	13	0	0.20	-0.30	0.00	-7.4200	1.0000
29	29	1	3	7	0	-0.50	0.20	0.00	6.5800	1.0000
30	30	1	3	15	0	0.20	-0.30	0.00	4.0200	1.0000
31	31	1	1	2	0	0.20	-0.30	0.00	85.490	1.0000
32	32	1	1	12	0	0.20	-0.30	0.00	0.98000	1.0000

REGRESSION DATA:
MAX. PARAMETER CORRECTION (DMAX) = 2.0000
SEARCH DIR. ADJUSTMENT PAR. (CSA) = 0.80000E-01
CLOSURE CRITERION (TOL) --------- = 0.10000E-02
MAXIMUM NO. OF ITERATIONS (ITMXP) = 100
MODIFY CONV. CRITERIA (>0) (FCONV)= 0.00000
USE PRIOR PAR EST (>0) (IPRIOR) = 1
OUTPUT UNIT FOR RESAN (IOUR) ---- = 35
FORMAT CODE FOR COV AND COR (IPRC)= 2
PRINT SENS., ETC. (>0) (IPRINT)= 0
PRINT LAST HEADS (1) (KPRINT) --- = 0
PRINT EIGEN V&V OF COV(>0)(LPRINT)= 2
CALC. SEN. W/ LAST PAR (>0)(LASTX)= 0
NO. FLETCHER-REEVES ITERS (NFIT)= 0
SUM OF SQUARES CLOSURE CRIT.(SOSC)= 0.00000

PARAMETER INFORMATION:
(CONVERGENCE CRITERIA LISTED HERE ARE USED TO SOLVE FOR SENSITIVITY-EQUATION SENSITIVITIES)

#	ID	INITIAL VALUE	LN	CONVERGENCE	CRITERIA	GROUP#	WEIGHT OF PRIOR EST.
1	Q	45.00	0	0.222E-08	0.222E-08	1	
2	Q	420.0	0	0.238E-09	0.238E-09	1	
3	Q	-0.9700E+05	0	0.103E-11	0.103E-11	2	
4	Q	-0.5100E+05	0	0.196E-11	0.196E-11	2	
5	CH	10.40	0	0.962E-08	0.962E-08	3	0.9246
6	CH	4.800	0	0.208E-07	0.208E-07	3	4.340
7	CH	5.100	0	0.196E-07	0.196E-07	3	3.845
8	T	70.00	0	0.143E-08	0.143E-08	4	
9	RCH	0.4000E-03	0	0.250E-03	0.250E-03	5	0.6944E+08
10	T	420.0	0	0.238E-09	0.238E-09	4	0.1417E-03
11	RCH	-0.2000E-03	0	0.500E-03	0.500E-03	5	
12	T	15.00	0	0.667E-08	0.667E-08	4	
13	RCH	0.1700E-03	0	0.588E-03	0.588E-03	5	0.3845E+09
14	KRB	0.8000E-01	0	0.125E-05	0.125E-05	6	0.1563E+05

PRIOR ESTIMATES OF PARAMETER SUMS

EQ.# :		1	2
PAR#	PID		
1	Q	0.000	0.000
2	Q	0.000	0.000
3	Q	1.00	0.000
4	Q	0.000	1.00
5	CH	0.000	0.000
6	CH	0.000	0.000
7	CH	0.000	0.000
8	T	0.000	0.000
9	RCH	0.000	0.000
10	T	0.000	0.000
11	RCH	0.000	0.000
12	T	0.000	0.000
13	RCH	0.000	0.000
14	KRB	0.000	0.000
ESTIMATE		-0.970E+05	-0.510E+05
WEIGHT**.5		0.515E-03	0.980E-03

WARNING -- NP > ND/3 : YOU MAY BE TRYING TO ESTIMATE TOO MANY PARAMETERS FOR THE DATA (PAR1RP)

17 WELLS

LAYER	ROW	COL	STRESS RATE	WELL NO.
1	1	1	500.00	1
1	1	2	500.00	2
1	1	3	500.00	3
1	1	4	500.00	4
1	1	5	500.00	5
1	1	6	500.00	6
1	1	7	280.00	7
1	1	8	56.000	8
1	1	9	56.000	9
1	1	10	280.00	10
1	1	11	280.00	11
1	1	12	280.00	12
1	1	13	280.00	13
1	1	14	280.00	14
1	1	15	280.00	15
1	7	11	-0.10000E+06	16
1	9	12	-50000.	17

RECHARGE WILL BE READ ON UNIT 31 USING FORMAT: (8F8.0)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	3.000000E-04	3.000000E-04	3.000000E-04	3.000000E-04	3.000000E-04	3.000000E-04	-1.000000E-04	0.000000	0.000000	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	0.000000
3	3.000000E-04	3.000000E-04	3.000000E-04	3.000000E-04	3.000000E-04	3.000000E-04	-1.000000E-04	0.000000	0.000000	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	0.000000
4	3.000000E-04	3.000000E-04	3.000000E-04	3.000000E-04	3.000000E-04	3.000000E-04	-1.000000E-04	0.000000	0.000000	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	0.000000
5	3.000000E-04	3.000000E-04	3.000000E-04	3.000000E-04	3.000000E-04	3.000000E-04	-1.000000E-04	0.000000	0.000000	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	0.000000
6	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	0.000000	0.000000	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	0.000000
7	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	0.000000	0.000000	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	0.000000
8	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	0.000000	0.000000	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	0.000000
9	0.000000	0.000000	0.000000	0.000000	0.000000	-1.000000E-04	-1.000000E-04	0.000000	0.000000	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	0.000000
10	0.000000	0.000000	0.000000	0.000000	0.000000	-1.000000E-04	-1.000000E-04	0.000000	0.000000	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	0.000000
11	0.000000	0.000000	0.000000	0.000000	0.000000	-1.000000E-04	-1.000000E-04	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
12	0.000000	0.000000	0.000000	0.000000	0.000000	-1.000000E-04	-1.000000E-04	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
13	0.000000	0.000000	0.000000	0.000000	0.000000	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	-1.000000E-04	0.000000
14	0.000000	0.000000	0.000000	0.000000	0.000000	-1.000000E-04	-1.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	0.000000
15	0.000000	0.000000	0.000000	0.000000	0.000000	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	0.000000
16	0.000000	0.000000	0.000000	0.000000	0.000000	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	0.000000
17	0.000000	0.000000	0.000000	0.000000	0.000000	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	2.000000E-04	0.000000

38 RIVER REACHES

LAYER	ROW	COL	STAGE	CONDUCTANCE	BOTTOM ELEVATION	RIVER REACH
1	1	8	4.500	1000.	-4.500	1
1	2	8	4.500	0.2000E+05	-4.500	2
1	3	8	4.500	0.2000E+05	-4.500	3
1	4	8	4.500	0.2000E+05	-4.500	4
1	5	8	4.500	0.2000E+05	-4.500	5
1	6	8	4.500	0.2000E+05	-4.500	6
1	7	8	4.500	0.2000E+05	-4.500	7
1	8	8	4.500	0.2000E+05	-4.500	8
1	9	8	4.500	0.2000E+05	-4.500	9
1	10	8	4.500	0.2000E+05	-4.500	10
1	11	8	4.500	4000.	-4.500	11
1	12	8	4.500	4000.	-4.500	12
1	12	9	4.500	4000.	-4.500	13
1	12	10	4.500	0.2000E+05	-4.500	14
1	12	11	4.500	0.2000E+05	-4.500	15
1	12	12	4.500	0.2000E+05	-4.500	16
1	12	13	4.500	0.2000E+05	-4.500	17
1	12	14	4.500	0.2000E+05	-4.500	18
1	12	15	4.500	0.2000E+05	-4.500	19
1	12	16	4.500	1000.	-4.500	20
1	1	9	4.500	1000.	-4.500	21
1	2	9	4.500	0.2000E+05	-4.500	22
1	3	9	4.500	0.2000E+05	-4.500	23
1	4	9	4.500	0.2000E+05	-4.500	24
1	5	9	4.500	0.2000E+05	-4.500	25
1	6	9	4.500	0.2000E+05	-4.500	26
1	7	9	4.500	0.2000E+05	-4.500	27
1	8	9	4.500	0.2000E+05	-4.500	28
1	9	9	4.500	0.2000E+05	-4.500	29
1	10	9	4.500	0.2000E+05	-4.500	30
1	11	9	4.500	4000.	-4.500	31

1	11	10	4.500	0.2000E+05	-4.500	32
1	11	11	4.500	0.2000E+05	-4.500	33
1	11	12	4.500	0.2000E+05	-4.500	34
1	11	13	4.500	0.2000E+05	-4.500	35
1	11	14	4.500	0.2000E+05	-4.500	36
1	11	15	4.500	0.2000E+05	-4.500	37
1	11	16	4.500	1000.	-4.500	38

MODEL INPUT MATRICES CALCULATED WITH INITIAL PARAMETER VALUES

TRANSMIS. ALONG ROWS LAYER 1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	70.0000	70.0000	70.0000	70.0000	70.0000	70.0000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000
2	70.0000	70.0000	70.0000	70.0000	70.0000	70.0000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000
3	70.0000	70.0000	70.0000	70.0000	70.0000	70.0000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000
4	70.0000	70.0000	70.0000	70.0000	70.0000	70.0000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000
5	70.0000	70.0000	70.0000	70.0000	70.0000	70.0000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000
6	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000
7	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000
8	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000
9	0.000000	0.000000	0.000000	0.000000	0.000000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000
10	0.000000	0.000000	0.000000	0.000000	0.000000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000
11	0.000000	0.000000	0.000000	0.000000	0.000000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000
12	0.000000	0.000000	0.000000	0.000000	0.000000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000
13	0.000000	0.000000	0.000000	0.000000	0.000000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000	420.000
14	0.000000	0.000000	0.000000	0.000000	0.000000	420.000	420.000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000
15	0.000000	0.000000	0.000000	0.000000	0.000000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000
16	0.000000	0.000000	0.000000	0.000000	0.000000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000
17	0.000000	0.000000	0.000000	0.000000	0.000000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000	15.0000

RECHARGE FOR TIME STEP 0

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	4.000000E-04	4.000000E-04	4.000000E-04	4.000000E-04	4.000000E-04	4.000000E-04	-2.000000E-04	0.000000	0.000000	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	0.000000
3	4.000000E-04	4.000000E-04	4.000000E-04	4.000000E-04	4.000000E-04	4.000000E-04	-2.000000E-04	0.000000	0.000000	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	0.000000
4	4.000000E-04	4.000000E-04	4.000000E-04	4.000000E-04	4.000000E-04	4.000000E-04	-2.000000E-04	0.000000	0.000000	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	0.000000
5	4.000000E-04	4.000000E-04	4.000000E-04	4.000000E-04	4.000000E-04	4.000000E-04	-2.000000E-04	0.000000	0.000000	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	0.000000
6	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	0.000000	0.000000	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	0.000000
7	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	0.000000	0.000000	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	0.000000
8	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	0.000000	0.000000	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	0.000000
9	0.000000	0.000000	0.000000	0.000000	0.000000	-2.000000E-04	-2.000000E-04	0.000000	0.000000	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	0.000000
10	0.000000	0.000000	0.000000	0.000000	0.000000	-2.000000E-04	-2.000000E-04	0.000000	0.000000	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	0.000000

11	0.000000	0.000000	0.000000	0.000000	0.000000	-2.000000E-04	-2.000000E-04	0.000000	0.000000
	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000		
12	0.000000	0.000000	0.000000	0.000000	0.000000	-2.000000E-04	-2.000000E-04	0.000000	0.000000
	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000		
13	0.000000	0.000000	0.000000	0.000000	0.000000	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04
	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	-2.000000E-04	0.000000		
14	0.000000	0.000000	0.000000	0.000000	0.000000	-2.000000E-04	-2.000000E-04	1.700000E-04	1.700000E-04
	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04	0.000000		
15	0.000000	0.000000	0.000000	0.000000	0.000000	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04
	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04	0.000000		
16	0.000000	0.000000	0.000000	0.000000	0.000000	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04
	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04	0.000000		
17	0.000000	0.000000	0.000000	0.000000	0.000000	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04
	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04	1.700000E-04	0.000000		

DATA AT HEAD LOCATIONS

OBS#	ID	LAYER	ROW	COL	TIME STEP	ROW/COL/TIME OFFSETS	MEAS. HEAD	CALC. HEAD	RESIDUAL	WEIGHT**.5	WEIGHTED RESIDUAL
1	1	1	17	8	0	0.00-0.30 0.00	69.220	75.801	-6.58	1.00	-6.58
2	2	1	17	15	0	0.00-0.30 0.00	80.580	90.568	-9.99	1.00	-9.99
3	3	1	16	13	0	0.20-0.30 0.00	72.660	80.344	-7.68	1.00	-7.68
4	4	1	15	11	0	0.20-0.30 0.00	49.550	56.003	-6.45	1.00	-6.45
5	5	1	14	7	0	0.20-0.20 0.00	10.040	12.171	-2.13	1.00	-2.13
6	6	1	13	12	0	0.20 0.40 0.00	7.660	9.334	-1.67	1.00	-1.67
7	7	1	13	14	0	0.30 0.40 0.00	11.440	11.742	-0.302	1.00	-0.302
8	8	1	13	16	0	0.40 0.00 0.00	13.500	13.831	-0.331	1.00	-0.331
9	9	1	11	11	0	0.20-0.30 0.00	4.600	4.151	0.449	1.00	0.449
10	10	1	10	8	0	0.20-0.30 0.00	5.300	4.519	0.781	1.00	0.781
11	11	1	9	12	0	0.20-0.30 0.00	-38.730	-45.718	6.99	1.00	6.99
12	12	1	10	16	0	0.20-0.30 0.00	6.620	4.994	1.63	1.00	1.63
13	13	1	9	7	0	0.20-0.30 0.00	6.540	4.830	1.71	1.00	1.71
14	14	1	8	4	0	0.20-0.30 0.00	12.190	11.969	0.221	1.00	0.221
15	15	1	8	9	0	0.20-0.30 0.00	4.400	3.934	0.466	1.00	0.466
16	16	1	7	11	0	0.20-0.30 0.00	-61.070	-72.046	11.0	1.00	11.0
17	17	1	7	7	0	0.20-0.30 0.00	6.440	5.878	0.562	1.00	0.562
18	18	1	7	14	0	0.20-0.30 0.00	-14.030	-19.139	5.11	1.00	5.11
19	19	1	6	3	0	0.20-0.30 0.00	15.730	16.378	-0.648	1.00	-0.648
20	20	1	6	5	0	0.20-0.30 0.00	14.890	11.882	3.01	1.00	3.01
21	21	1	6	9	0	0.20-0.30 0.00	4.300	3.894	0.406	1.00	0.406
22	22	1	5	11	0	0.20-0.30 0.00	-16.150	-19.851	3.70	1.00	3.70
23	23	1	5	14	0	0.20-0.30 0.00	-5.370	-9.861	4.49	1.00	4.49
24	24	1	5	16	0	0.20-0.30 0.00	7.590	7.959	-0.369	1.00	-0.369
25	25	1	4	3	0	0.20-0.30 0.00	42.420	40.661	1.76	1.00	1.76
26	26	1	4	5	0	0.20-0.30 0.00	29.080	28.663	0.417	1.00	0.417
27	27	1	4	10	0	0.20-0.30 0.00	1.470	0.682	0.788	1.00	0.788
28	28	1	4	13	0	0.20-0.30 0.00	-7.420	-10.410	2.99	1.00	2.99
29	29	1	3	7	0	-0.50 0.20 0.00	6.580	6.021	0.559	1.00	0.559
30	30	1	3	15	0	0.20-0.30 0.00	4.020	3.962	0.582E-01	1.00	0.582E-01
31	31	1	1	2	0	0.20-0.30 0.00	85.490	72.029	13.5	1.00	13.5
32	32	1	1	12	0	0.20-0.30 0.00	0.980	-2.157	3.14	1.00	3.14

STATISTICS FOR THESE RESIDUALS :
MAXIMUM WEIGHTED RESIDUAL : 0.135E+02 OBS# 31
MINIMUM WEIGHTED RESIDUAL :-0.999E+01 OBS# 2
AVERAGE WEIGHTED RESIDUAL : 0.859E+00
RESIDUALS >= 0. : 22
RESIDUALS < 0. : 10
NUMBER OF RUNS : 6 IN 32 OBSERVATIONS

SUM OF SQUARED WEIGHTED RESIDUALS (HEADS) 701.51

PARAMETERS WITH PRIOR INFORMATION, BY GROUP

PARAMETER #	PID	VALUES MEAS.	CALC.	RESIDUAL	WEIGHT **.5	WEIGHTED RESIDUAL
5	CH	0.104E+02	0.104E+02	0.000E+00	0.962	0.000
6	CH	0.480E+01	0.480E+01	0.000E+00	2.08	0.000
7	CH	0.510E+01	0.510E+01	0.000E+00	1.96	0.000
10	T	0.420E+03	0.420E+03	0.000E+00	0.119E-01	0.000
9	RCH	0.400E-03	0.400E-03	0.000E+00	0.833E+04	0.000
13	RCH	0.170E-03	0.170E-03	0.000E+00	0.196E+05	0.000
14	KRB	0.800E-01	0.800E-01	0.000E+00	125.	0.000

```
STATISTICS FOR THESE RESIDUALS :
MAXIMUM WEIGHTED RESIDUAL  : 0.000E+00
MINIMUM WEIGHTED RESIDUAL  : 0.000E+00
AVERAGE WEIGHTED RESIDUAL  : 0.000E+00
# RESIDUALS >= 0. :       7
# RESIDUALS < 0.  :       0
NUMBER OF RUNS  :    6  IN   40 OBSERVATIONS

PARAMETER SUMS WITH PRIOR INFORMATION

                  VALUES                           WEIGHT  WEIGHTED
   EQ.#        MEAS.    CALC.      RESIDUAL         **.5    RESIDUAL
     1    -0.970E+05 -0.970E+05  0.000E+00  0.515E-03  0.000
     2    -0.510E+05 -0.510E+05  0.000E+00  0.980E-03  0.000
STATISTICS FOR THESE RESIDUALS :
MAXIMUM WEIGHTED RESIDUAL  : 0.000E+00
MINIMUM WEIGHTED RESIDUAL  : 0.000E+00
AVERAGE WEIGHTED RESIDUAL  : 0.000E+00
# RESIDUALS >= 0. :       2
# RESIDUALS < 0.  :       0
NUMBER OF RUNS  :    6  IN   42 OBSERVATIONS

SUM OF SQUARED WEIGHTED RESIDUALS (ALL DEP. VAR.)    701.51
SUM OF SQUARED WEIGHTED RESIDUALS (WITH PARAMETERS)  701.51
STATISTICS FOR ALL RESIDUALS :
AVERAGE WEIGHTED RESIDUAL  :-0.671E+00
# RESIDUALS >= 0. :      19
# RESIDUALS < 0.  :      22
NUMBER OF RUNS  :    6  IN   42 OBSERVATIONS
RUNS STATISTIC (TOO FEW RUNS):     -4.74
 (IF #NEG>10 AND #POS>10, P(STAT < -1.28) = 0.10,
                          P(STAT < -1.645) = 0.05,
                          P(STAT < -1.96) = 0.025)
RUNS STATISTIC (TOO MANY RUNS):     -5.05
 (IF #NEG>10 AND #POS>10, P(STAT > 1.28) = 0.10,
                          P(STAT > 1.645) = 0.05,
                          P(STAT > 1.96) = 0.025)

SENSITIVITIES

PARAMETER # :          1          2          3          4          5          6          7          8          9         10
                      11         12         13         14
PARAMETER ID :         Q          Q          Q          Q         CH         CH         CH          T        RCH          T
                     RCH          T        RCH        KRB
OBS# & ID
   1     1       0.147E-02 0.818E-06 0.178E-06 0.361E-06 0.695E-04 0.436E-03 0.325E-01-0.401E-03  711.       -0.587E-02
                 0.761E+04 -4.59     0.426E+06 0.577
   2     2       0.380E-03 0.251E-06 0.140E-06 0.422E-06 0.121E-03 0.169E-02 0.154     -0.104E-03  184.       -0.368E-02
                 0.285E+04 -5.63     0.509E+06 0.576
   3     3       0.527E-03 0.331E-06 0.159E-06 0.466E-06 0.111E-03 0.120E-02 0.102     -0.144E-03  255.       -0.399E-02
                 0.354E+04 -4.94     0.449E+06 0.662
   4     4       0.893E-03 0.521E-06 0.179E-06 0.452E-06 0.818E-04 0.538E-03 0.396E-01-0.244E-03  431.       -0.461E-02
                 0.516E+04 -3.31     0.308E+06 0.683
   5     5       0.229E-02 0.124E-05 0.184E-06 0.248E-06 0.455E-04 0.101E-03 0.380E-02-0.623E-03 0.110E+04-0.647E-02
                 0.112E+05-0.329     0.553E+05 0.456
   6     6       0.652E-04 0.112E-06 0.204E-06 0.782E-06 0.134E-03 0.632E-03 0.314E-01-0.178E-04  31.5      -0.189E-02
                 0.170E+04-0.274     0.306E+05  1.11
   7     7       0.313E-04 0.610E-07 0.912E-07 0.295E-06 0.155E-03 0.287E-02 0.256     -0.855E-05  15.1      -0.173E-02
                 0.124E+04-0.426     0.432E+05 0.335
   8     8       0.322E-04 0.306E-07 0.304E-07 0.960E-07 0.510E-04 0.125E-02 0.746     -0.878E-05  15.6      -0.116E-02
                  512.     -0.560    0.529E+05 0.106
   9     9       0.249E-05 0.415E-06 0.152E-05 0.385E-05 0.437E-03 0.709E-03 0.419E-04-0.669E-06  1.20       0.658E-04
                  95.7      0.869E-05  60.7      4.05
  10    10       0.332E-03 0.199E-06 0.101E-06 0.170E-06 0.238E-04 0.348E-04 0.598E-04-0.902E-04  160.       -0.801E-04
                  765.     -0.168E-03  651.     0.302
  11    11       0.963E-05 0.490E-04 0.151E-03 0.684E-03 0.566E-01 0.934E-01 0.837E-04-0.159E-05  4.50       0.119
                 0.538E+04 0.723E-06  14.3       5.59
  12    12       0.102E-07 0.112E-06 0.179E-06 0.456E-06 0.438E-01 0.938     0.309E-04-0.102E-08 0.459E-02 0.105E-03
                  25.9      0.714E-08 0.262     0.232E-01
  13    13       0.583E-02 0.315E-05 0.398E-06 0.343E-06 0.803E-04 0.947E-04 0.230E-03-0.158E-02 0.282E+04-0.701E-03
                 0.727E+04-0.700E-03 0.263E+04  1.08
  14    14       0.631E-01 0.284E-04 0.649E-06 0.267E-06 0.162E-03 0.133E-03 0.839E-04-0.182E-01 0.300E+05-0.150E-01
                 0.373E+05-0.260E-03  975.       1.34
  15    15       0.103E-03 0.998E-06 0.429E-05 0.283E-05 0.705E-03 0.848E-03 0.166E-05-0.274E-04  50.1       0.920E-04
                  185.     -0.367E-05  14.0       6.67
  16    16       0.274E-04 0.115E-03 0.700E-03 0.156E-03 0.758E-01 0.786E-01 0.283E-04-0.396E-05  12.9       0.182
                 0.632E+04-0.181E-06  5.25       7.30
```

```
  17    17        0.143E-01 0.858E-05 0.818E-06 0.298E-06 0.154E-03 0.140E-03 0.549E-04-0.367E-02 0.697E+04-0.300E-02
                  0.104E+05-0.169E-03  635.      1.77
  18    18        0.895E-05 0.147E-03 0.164E-03 0.154E-03 0.256     0.345     0.733E-04-0.382E-06  3.89     0.597E-01
                  0.774E+04 0.134E-06  4.53      2.44
  19    19        0.862E-01 0.365E-04 0.639E-06 0.254E-06 0.170E-03 0.134E-03 0.745E-04-0.253E-01 0.407E+05-0.243E-01
                  0.418E+05-0.231E-03  865.      1.28
  20    20        0.552E-01 0.281E-04 0.661E-06 0.254E-06 0.170E-03 0.136E-03 0.702E-04-0.153E-01 0.267E+05-0.153E-01
                  0.294E+05-0.218E-03  815.      1.33
  21    21        0.146E-03 0.292E-05 0.559E-05 0.133E-05 0.125E-02 0.102E-02 0.609E-06-0.251E-04  72.3     0.111E-03
                   209.     -0.106E-05  4.00      7.11
  22    22        0.328E-04 0.325E-03 0.214E-03 0.582E-04 0.115     0.752E-01 0.136E-04 0.233E-05  14.3     0.588E-01
                  0.713E+04-0.101E-06  1.99      4.66
  23    23        0.125E-04 0.412E-03 0.122E-03 0.663E-04 0.398     0.269     0.234E-04 0.569E-06  4.98     0.396E-01
                  0.877E+04 0.176E-07  1.88      1.81
  24    24        0.102E-06 0.409E-05 0.983E-06 0.602E-06 0.590     0.407     0.262E-06 0.459E-08 0.401E-01 0.342E-03
                   102.     0.265E-09 0.177E-01 0.150E-01
  25    25        0.205     0.755E-04 0.568E-06 0.220E-06 0.191E-03 0.134E-03 0.601E-04-0.371     0.838E+05-0.245E-01
                  0.333E+05-0.187E-03  698.      1.01
  26    26        0.129     0.886E-04 0.509E-06 0.188E-06 0.211E-03 0.135E-03 0.443E-04-0.245     0.565E+05-0.168E-01
                  0.216E+05-0.137E-03  514.     0.794
  27    27        0.411E-04 0.149E-03 0.325E-04 0.967E-05 0.318E-01 0.166E-01 0.239E-05 0.900E-05  17.2     0.894E-02
                  0.219E+04-0.762E-07 0.556      3.20
  28    28        0.221E-04 0.832E-03 0.128E-03 0.550E-04 0.344     0.170     0.162E-04 0.186E-05  8.26     0.400E-01
                  0.107E+05-0.138E-07  1.59      2.24
  29    29        0.114E-01 0.236E-03 0.159E-06 0.539E-07 0.297E-03 0.122E-03 0.166E-05 0.121E-02 0.316E+04-0.372E-02
                  0.170E+04-0.510E-05  19.1     -0.515
  30    30        0.693E-05 0.613E-03 0.315E-04 0.146E-04 0.671     0.187     0.477E-05 0.621E-06  2.45     0.107E-01
                  0.509E+04-0.235E-08 0.430     0.581
  31    31        0.547     0.106E-03 0.526E-06 0.202E-06 0.202E-03 0.133E-03 0.534E-04-0.827     0.122E+06-0.230E-01
                  0.301E+05-0.166E-03  620.     0.855
  32    32        0.386E-04 0.292E-02 0.643E-04 0.249E-04 0.275     0.910E-01 0.709E-05 0.398E-05  12.0     0.194E-01
                  0.101E+05-0.277E-07 0.801      1.74

UNSCALED RIGHT-HAND SIDE VECTOR (= -0.5*GRADIENT)
  7.9058      0.19006E-01  0.11534E-01  0.80376E-02   6.5618       6.9291      -3.1775      -11.922      0.18719E+07  3.4751
0.67349E+06   147.23      -.13510E+08   177.43

ITERATION NO. =     1
VALUES FROM LEAST-SQUARES REGRESSION PROCEDURE :
  DET OF SCALED LEAST-SQUARES MATRIX = 0.31565E-07
  MARQUARDT PARAMETER (AMP)--------- = 0.00000
  FACTOR FOR SCALING PAR. CHANGE (AP)=  1.0000
  MAX. FRACTIONAL PAR. CHANGE (DMX)  =  1.0844
  MAX. FRAC. CHANGE OCCURRED FOR PAR.#      2

UPDATED ESTIMATES OF REGRESSION PARAMETERS :
    Q            Q            Q            Q            CH           CH           CH           T            RCH          T
    RCH          T            RCH          KRB
  87.694       875.46       -96921.      -51064.      10.161       5.1424       5.2235       67.610       0.29818E-03  478.66
-.14004E-03   13.310       0.13339E-03  0.80495E-01

SUM OF SQUARED WEIGHTED RESIDUALS (ALL DEP. VAR.)   31.053
SUM OF SQUARED WEIGHTED RESIDUALS (WITH PARAMETERS)  33.405

ITERATION NO. =     2
VALUES FROM LEAST-SQUARES REGRESSION PROCEDURE :
  DET OF SCALED LEAST-SQUARES MATRIX = 0.15610E-07
  MARQUARDT PARAMETER (AMP)--------- = 0.00000
  FACTOR FOR SCALING PAR. CHANGE (AP)=  1.0000
  MAX. FRACTIONAL PAR. CHANGE (DMX)  = 0.90701E-01
  MAX. FRAC. CHANGE OCCURRED FOR PAR.#     11

UPDATED ESTIMATES OF REGRESSION PARAMETERS :
    Q            Q            Q            Q            CH           CH           CH           T            RCH          T
    RCH          T            RCH          KRB
  84.059       937.99       -96870.      -51047.      10.155       5.1421       5.2228       66.264       0.29818E-03  487.46
-.12734E-03   12.854       0.13067E-03  0.80461E-01

SUM OF SQUARED WEIGHTED RESIDUALS (ALL DEP. VAR.)   24.131
SUM OF SQUARED WEIGHTED RESIDUALS (WITH PARAMETERS)  26.722

ITERATION NO. =     3
VALUES FROM LEAST-SQUARES REGRESSION PROCEDURE :
  DET OF SCALED LEAST-SQUARES MATRIX = 0.87981E-07
  MARQUARDT PARAMETER (AMP)--------- = 0.10000E-02
  FACTOR FOR SCALING PAR. CHANGE (AP)=  1.0000
  MAX. FRACTIONAL PAR. CHANGE (DMX)  = 0.35499E-02
  MAX. FRAC. CHANGE OCCURRED FOR PAR.#      2
```

```
UPDATED ESTIMATES OF REGRESSION PARAMETERS :
    Q           Q            Q            Q           CH          CH          CH          T          RCH          T
   RCH          T           RCH          KRB
  84.288      941.32      -96866.      -51045.      10.157      5.1424      5.2227      66.382    0.29830E-03   487.63
-.12767E-03   12.870    0.13077E-03  0.80459E-01

SUM OF SQUARED WEIGHTED RESIDUALS (ALL DEP. VAR.)   24.128
SUM OF SQUARED WEIGHTED RESIDUALS (WITH PARAMETERS)  26.717

ITERATION NO. =     4
VALUES FROM LEAST-SQUARES REGRESSION PROCEDURE :
  DET OF SCALED LEAST-SQUARES MATRIX = 0.14678E-07
  MARQUARDT PARAMETER (AMP)--------- = 0.00000
  FACTOR FOR SCALING PAR. CHANGE (AP)=  1.0000
  MAX. FRACTIONAL PAR. CHANGE (DMX)  = -.52083E-02
  MAX. FRAC. CHANGE OCCURRED FOR PAR.#      11

UPDATED ESTIMATES OF REGRESSION PARAMETERS :
    Q           Q            Q            Q           CH          CH          CH          T          RCH          T
   RCH          T           RCH          KRB
  84.497      943.73      -96867.      -51045.      10.158      5.1426      5.2226      66.523    0.29873E-03   487.67
-.12833E-03   12.912    0.13119E-03  0.80459E-01

SUM OF SQUARED WEIGHTED RESIDUALS (ALL DEP. VAR.)   24.145
SUM OF SQUARED WEIGHTED RESIDUALS (WITH PARAMETERS)  26.717

ITERATION NO. =     5
VALUES FROM LEAST-SQUARES REGRESSION PROCEDURE :
  DET OF SCALED LEAST-SQUARES MATRIX = 0.14787E-07
  MARQUARDT PARAMETER (AMP)--------- = 0.00000
  FACTOR FOR SCALING PAR. CHANGE (AP)=  1.0000
  MAX. FRACTIONAL PAR. CHANGE (DMX)  = -.51786E-04
  MAX. FRAC. CHANGE OCCURRED FOR PAR.#       1

UPDATED ESTIMATES OF REGRESSION PARAMETERS :
    Q           Q            Q            Q           CH          CH          CH          T          RCH          T
   RCH          T           RCH          KRB
  84.492      943.72      -96867.      -51045.      10.158      5.1426      5.2226      66.522    0.29873E-03   487.67
-.12833E-03   12.912    0.13119E-03  0.80459E-01

PARAMETER ESTIMATION CONVERGED BY SATISFYING TOL CRITERIA
```

DATA AT HEAD LOCATIONS

OBS#	ID	LAYER	ROW	COL	TIME STEP	ROW/COL/TIME OFFSETS	MEAS. HEAD	CALC. HEAD	RESIDUAL	WEIGHT**.5	WEIGHTED RESIDUAL
1	1	1	17	8	0	0.00-0.30 0.00	69.220	68.077	1.14	1.00	1.14
2	2	1	17	15	0	0.00-0.30 0.00	80.580	81.404	-0.824	1.00	-0.824
3	3	1	16	13	0	0.20-0.30 0.00	72.660	72.219	0.441	1.00	0.441
4	4	1	15	11	0	0.20-0.30 0.00	49.550	50.375	-0.825	1.00	-0.825
5	5	1	14	7	0	0.20-0.20 0.00	10.040	11.081	-1.04	1.00	-1.04
6	6	1	13	12	0	0.20 0.40 0.00	7.660	8.693	-1.03	1.00	-1.03
7	7	1	13	14	0	0.30 0.40 0.00	11.440	10.899	0.541	1.00	0.541
8	8	1	13	16	0	0.40 0.00 0.00	13.500	12.903	0.597	1.00	0.597
9	9	1	11	11	0	0.20-0.30 0.00	4.600	4.162	0.438	1.00	0.438
10	10	1	10	8	0	0.20-0.30 0.00	5.300	4.538	0.762	1.00	0.762
11	11	1	9	12	0	0.20-0.30 0.00	-38.730	-38.395	-0.335	1.00	-0.335
12	12	1	10	16	0	0.20-0.30 0.00	6.620	5.313	1.31	1.00	1.31
13	13	1	9	7	0	0.20-0.30 0.00	6.540	5.111	1.43	1.00	1.43
14	14	1	8	4	0	0.20-0.30 0.00	12.190	12.976	-0.786	1.00	-0.786
15	15	1	8	9	0	0.20-0.30 0.00	4.400	3.956	0.444	1.00	0.444
16	16	1	7	11	0	0.20-0.30 0.00	-61.070	-60.909	-0.161	1.00	-0.161
17	17	1	7	7	0	0.20-0.30 0.00	6.440	6.220	0.220	1.00	0.220
18	18	1	7	14	0	0.20-0.30 0.00	-14.030	-15.046	1.02	1.00	1.02
19	19	1	6	3	0	0.20-0.30 0.00	15.730	17.006	-1.28	1.00	-1.28
20	20	1	6	5	0	0.20-0.30 0.00	14.890	12.394	2.50	1.00	2.50
21	21	1	6	9	0	0.20-0.30 0.00	4.300	3.920	0.380	1.00	0.380
22	22	1	5	11	0	0.20-0.30 0.00	-16.150	-15.816	-0.334	1.00	-0.334
23	23	1	5	14	0	0.20-0.30 0.00	-5.370	-6.814	1.44	1.00	1.44
24	24	1	5	16	0	0.20-0.30 0.00	7.590	7.984	-0.394	1.00	-0.394
25	25	1	4	3	0	0.20-0.30 0.00	42.420	42.371	0.491E-01	1.00	0.491E-01
26	26	1	4	5	0	0.20-0.30 0.00	29.080	29.372	-0.292	1.00	-0.292
27	27	1	4	10	0	0.20-0.30 0.00	1.470	1.411	0.591E-01	1.00	0.591E-01
28	28	1	4	13	0	0.20-0.30 0.00	-7.420	-7.054	-0.366	1.00	-0.366
29	29	1	3	7	0	-0.50 0.20 0.00	6.580	6.126	0.454	1.00	0.454
30	30	1	3	15	0	0.20-0.30 0.00	4.020	5.084	-1.06	1.00	-1.06
31	31	1	1	2	0	0.20-0.30 0.00	85.490	85.431	0.585E-01	1.00	0.585E-01
32	32	1	1	12	0	0.20-0.30 0.00	0.980	0.896	0.844E-01	1.00	0.844E-01

```
STATISTICS FOR THESE RESIDUALS :
MAXIMUM WEIGHTED RESIDUAL  : 0.128E+01 OBS#     19
MINIMUM WEIGHTED RESIDUAL  :-0.250E+01 OBS#     20
AVERAGE WEIGHTED RESIDUAL  :-0.145E+00
# RESIDUALS >= 0. :      13
# RESIDUALS < 0.  :      19
NUMBER OF RUNS  :   23  IN   32 OBSERVATIONS

SUM OF SQUARED WEIGHTED RESIDUALS (HEADS)  24.146

PARAMETERS WITH PRIOR INFORMATION, BY GROUP

 PARAMETER         VALUES                          WEIGHT   WEIGHTED
  #   PID      MEAS.     CALC.     RESIDUAL          **.5    RESIDUAL
  5   CH    0.104E+02  0.102E+02  0.242E+00  0.962       0.232
  6   CH    0.480E+01  0.514E+01 -0.343E+00   2.08      -0.714
  7   CH    0.510E+01  0.522E+01 -0.123E+00   1.96      -0.240
 10   T     0.420E+03  0.488E+03 -0.677E+02  0.119E-01 -0.806
  9   RCH   0.400E-03  0.299E-03  0.101E-03  0.833E+04  0.844
 13   RCH   0.170E-03  0.131E-03  0.388E-04  0.196E+05  0.761
 14   KRB   0.800E-01  0.805E-01 -0.459E-03   125.     -0.574E-01
STATISTICS FOR THESE RESIDUALS :
MAXIMUM WEIGHTED RESIDUAL  : 0.844E+00
MINIMUM WEIGHTED RESIDUAL  :-0.806E+00
AVERAGE WEIGHTED RESIDUAL  : 0.286E-02
# RESIDUALS >= 0. :       3
# RESIDUALS < 0.  :       4
NUMBER OF RUNS  :   27  IN   40 OBSERVATIONS

PARAMETER SUMS WITH PRIOR INFORMATION

                   VALUES                              WEIGHT  WEIGHTED
   EQ.#        MEAS.     CALC.      RESIDUAL             **.5   RESIDUAL
     1     -0.970E+05 -0.969E+05 -0.133E+03  0.515E-03 -0.683E-01
     2     -0.510E+05 -0.510E+05  0.452E+02  0.980E-03  0.443E-01
STATISTICS FOR THESE RESIDUALS :
MAXIMUM WEIGHTED RESIDUAL  : 0.443E-01
MINIMUM WEIGHTED RESIDUAL  :-0.683E-01
AVERAGE WEIGHTED RESIDUAL  :-0.120E-01
# RESIDUALS >= 0. :       1
# RESIDUALS < 0.  :       1
NUMBER OF RUNS  :   28  IN   42 OBSERVATIONS

SUM OF SQUARED WEIGHTED RESIDUALS (ALL DEP. VAR.)   24.146
SUM OF SQUARED WEIGHTED RESIDUALS (WITH PARAMETERS)  26.717
STATISTICS FOR ALL RESIDUALS :
AVERAGE WEIGHTED RESIDUAL  :-0.113E+00
# RESIDUALS >= 0. :      17
# RESIDUALS < 0.  :      24
NUMBER OF RUNS  :   28  IN   42 OBSERVATIONS
RUNS STATISTIC (TOO FEW RUNS):      2.48
 (IF #NEG>10 AND #POS>10, P(STAT < -1.28) = 0.10,
                          P(STAT < -1.645) = 0.05,
                          P(STAT < -1.96) = 0.025)
RUNS STATISTIC (TOO MANY RUNS):     2.15
 (IF #NEG>10 AND #POS>10, P(STAT > 1.28) = 0.10,
                          P(STAT > 1.645) = 0.05,
                          P(STAT > 1.96) = 0.025)

SENSITIVITIES

PARAMETER # :        1         2         3         4         5         6         7         8         9         10
                    11        12        13        14
PARAMETER ID :       Q         Q         Q         Q        CH        CH        CH         T        RCH         T
                    RCH        T        RCH       KRB
OBS# & ID
   1     1      0.132E-02 0.650E-06 0.200E-06 0.401E-06 0.895E-04 0.522E-03 0.323E-01-0.390E-03  641.     -0.388E-02
                0.666E+04 -4.78     0.489E+06 0.611
   2     2      0.340E-03 0.211E-06 0.156E-06 0.468E-06 0.156E-03 0.201E-02 0.155     -0.100E-03  165.     -0.229E-02
                0.248E+04 -5.86     0.587E+06 0.630
   3     3      0.472E-03 0.275E-06 0.177E-06 0.518E-06 0.143E-03 0.143E-02 0.102     -0.139E-03  229.     -0.251E-02
                0.308E+04 -5.15     0.518E+06 0.726
   4     4      0.801E-03 0.421E-06 0.199E-06 0.502E-06 0.105E-03 0.648E-03 0.394E-01-0.237E-03  388.     -0.298E-02
                0.451E+04 -3.44     0.353E+06 0.744
   5     5      0.206E-02 0.972E-06 0.207E-06 0.274E-06 0.583E-04 0.120E-03 0.338E-02-0.607E-03  996.     -0.448E-02
                0.983E+04-0.340     0.564E+05 0.455
```

```
  6     6      0.568E-04 0.112E-06 0.227E-06 0.869E-06 0.172E-03 0.779E-03 0.313E-01-0.168E-04  27.5      -0.120E-02
               0.147E+04-0.285      0.336E+05  1.24
  7     7      0.272E-04 0.618E-07 0.101E-06 0.327E-06 0.200E-03 0.341E-02 0.257     -0.803E-05  13.2      -0.105E-02
               0.107E+04-0.443      0.485E+05 0.365
  8     8      0.284E-04 0.280E-07 0.336E-07 0.106E-06 0.656E-04 0.147E-02 0.747     -0.839E-05  13.8      -0.689E-03
                443.     -0.583     0.604E+05 0.112
  9     9      0.265E-05 0.410E-06 0.150E-05 0.378E-05 0.500E-03 0.812E-03 0.480E-04-0.769E-06  1.28      0.609E-04
                95.2     0.735E-05  61.8      3.87
 10    10      0.306E-03 0.166E-06 0.114E-06 0.190E-06 0.310E-04 0.453E-04 0.515E-04-0.900E-04  148.      -0.104E-03
                694.     -0.123E-03  590.     0.254
 11    11      0.109E-04 0.424E-04 0.130E-03 0.590E-03 0.567E-01 0.937E-01 0.108E-03-0.197E-05  5.10      0.879E-01
               0.464E+04 0.731E-06  15.9      5.27
 12    12      0.115E-07 0.976E-07 0.156E-06 0.396E-06 0.438E-01 0.938     0.396E-04-0.125E-08 0.517E-02 0.749E-04
                22.5     0.489E-08 0.289     0.189E-01
 13    13      0.517E-02 0.244E-05 0.444E-06 0.382E-06 0.103E-03 0.122E-03 0.189E-03-0.152E-02 0.251E+04-0.121E-02
               0.637E+04-0.498E-03 0.231E+04  1.07
 14    14      0.558E-01 0.214E-04 0.732E-06 0.302E-06 0.206E-03 0.172E-03 0.703E-04-0.171E-01 0.266E+05-0.153E-01
               0.329E+05-0.189E-03  871.      1.36
 15    15      0.107E-03 0.981E-06 0.423E-05 0.279E-05 0.808E-03 0.972E-03 0.182E-05-0.307E-04  51.9      0.866E-04
                187.     -0.306E-05  14.3      6.33
 16    16      0.308E-04 0.997E-04 0.603E-03 0.135E-03 0.759E-01 0.788E-01 0.364E-04-0.492E-05  14.5      0.134
               0.546E+04-0.152E-06  5.86      6.86
 17    17      0.126E-01 0.667E-05 0.911E-06 0.332E-06 0.197E-03 0.181E-03 0.455E-04-0.356E-02 0.618E+04-0.333E-02
               0.913E+04-0.122E-03  561.      1.78
 18    18      0.999E-05 0.127E-03 0.141E-03 0.133E-03 0.256     0.346     0.942E-04-0.469E-06  4.35      0.431E-01
               0.668E+04 0.144E-06  5.03      2.28
 19    19      0.763E-01 0.275E-04 0.723E-06 0.289E-06 0.215E-03 0.173E-03 0.630E-04-0.237E-01 0.361E+05-0.226E-01
               0.371E+05-0.170E-03  781.      1.30
 20    20      0.488E-01 0.213E-04 0.745E-06 0.288E-06 0.216E-03 0.176E-03 0.591E-04-0.146E-01 0.237E+05-0.144E-01
               0.260E+05-0.159E-03  732.      1.35
 21    21      0.150E-03 0.288E-05 0.550E-05 0.131E-05 0.143E-02 0.117E-02 0.724E-06-0.289E-04  74.2      0.104E-03
                210.     -0.889E-06  4.15      6.71
 22    22      0.362E-04 0.280E-03 0.185E-03 0.503E-04 0.115     0.754E-01 0.175E-04 0.298E-05  15.8      0.424E-01
               0.616E+04-0.856E-07  2.22      4.30
 23    23      0.139E-04 0.355E-03 0.105E-03 0.572E-04 0.398     0.269     0.301E-04 0.714E-06  5.50      0.279E-01
               0.756E+04 0.243E-07  2.09      1.67
 24    24      0.113E-06 0.352E-05 0.848E-06 0.519E-06 0.590     0.407     0.336E-06 0.573E-08 0.443E-01 0.237E-03
                88.2     0.321E-09 0.197E-01 0.138E-01
 25    25      0.202     0.622E-04 0.642E-06 0.250E-06 0.243E-03 0.173E-03 0.509E-04-0.410     0.820E+05-0.219E-01
               0.296E+05-0.137E-03  631.      1.00
 26    26      0.127     0.749E-04 0.572E-06 0.212E-06 0.269E-03 0.174E-03 0.373E-04-0.266     0.553E+05-0.148E-01
               0.191E+05-0.100E-03  462.      0.755
 27    27      0.447E-04 0.130E-03 0.283E-04 0.842E-05 0.321E-01 0.167E-01 0.310E-05 0.116E-04  18.7      0.626E-02
               0.191E+04-0.660E-07 0.613      2.85
 28    28      0.243E-04 0.717E-03 0.110E-03 0.475E-04 0.345     0.170     0.209E-04 0.233E-05  9.08      0.276E-01
               0.919E+04-0.580E-08  1.77      2.02
 29    29      0.981E-02 0.207E-03 0.175E-06 0.595E-07 0.384E-03 0.157E-03 0.122E-05 0.113E-02 0.267E+04-0.336E-02
               0.140E+04-0.323E-05  14.9     -0.749
 30    30      0.764E-05 0.528E-03 0.271E-04 0.126E-04 0.671     0.187     0.614E-05 0.765E-06  2.68      0.675E-02
               0.438E+04-0.223E-09 0.478      0.519
 31    31      0.563     0.892E-04 0.593E-06 0.229E-06 0.258E-03 0.171E-03 0.452E-04 -1.07     0.123E+06-0.206E-01
               0.267E+05-0.121E-03  559.      0.818
 32    32      0.425E-04 0.252E-02 0.556E-04 0.215E-04 0.275     0.911E-01 0.912E-05 0.480E-05  13.1      0.105E-01
               0.875E+04-0.210E-07 0.888      1.45
```

PARAMETER VALUES AND STATISTICS FOR ALL ITERATIONS

```
          PAR. ID.:
        Q          Q          Q          Q          CH         CH
        CH         T          RCH        T          RCH        T
        RCH        KRB
                                                                   SUM OF SQUARED WEIGHTED RESIDUALS                          AMP OR
ITER                                                               HEADS       W/FLOWS      W/PARAMS      DMX          PAR#   AGMX
     0.450E+02 0.420E+03-0.970E+05-0.510E+05 0.104E+02 0.480E+01
     0.510E+01 0.700E+02 0.400E-03 0.420E+03-0.200E-03 0.150E+02
     0.170E-03 0.800E-01
  1                                                                702.        702.         702.          1.08          2    0.000
     0.877E+02 0.875E+03-0.969E+05-0.511E+05 0.102E+02 0.514E+01
     0.522E+01 0.676E+02 0.298E-03 0.479E+03-0.140E-03 0.133E+02
     0.133E-03 0.805E-01
  2                                                                31.1        31.1         33.4          0.907E-01    11    0.000
     0.841E+02 0.938E+03-0.969E+05-0.510E+05 0.102E+02 0.514E+01
     0.522E+01 0.663E+02 0.298E-03 0.487E+03-0.127E-03 0.129E+02
     0.131E-03 0.805E-01
  3                                                                24.1        24.1         26.7          0.355E-02     2    0.100E-02
     0.843E+02 0.941E+03-0.969E+05-0.510E+05 0.102E+02 0.514E+01
     0.522E+01 0.664E+02 0.298E-03 0.488E+03-0.128E-03 0.129E+02
     0.131E-03 0.805E-01
  4                                                                24.1        24.1         26.7         -0.521E-02    11    0.000
```

```
     0.845E+02 0.944E+03-0.969E+05-0.510E+05 0.102E+02 0.514E+01
     0.522E+01 0.665E+02 0.299E-03 0.488E+03-0.128E-03 0.129E+02
     0.131E-03 0.805E-01
5                                                          24.1      24.1      26.7    -0.518E-04   1  0.000
     0.845E+02 0.944E+03-0.969E+05-0.510E+05 0.102E+02 0.514E+01
     0.522E+01 0.665E+02 0.299E-03 0.488E+03-0.128E-03 0.129E+02
     0.131E-03 0.805E-01
6                                                          24.1      24.1      26.7
```

COVARIANCE MAT.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	622.988	5524.23	-1340.21	-105.164	3.90627	0.496596	-0.126735	381.153	8.333315E-04	94.0038	-1.528137E-03	35.7137	3.649065E-04	1.406541E-03
2	5524.23	230069.	-28512.0	7780.00	-50.6715	4.50571	-1.72681	3897.63	1.283299E-02	904.322	-1.977631E-02	475.420	4.853034E-03	6.535410E-03
3	-1340.21	-28512.0	3.186814E+06	261081.	3.75333	5.06365	-0.646858	-1071.27	-5.194813E-03	-12143.8	-7.155699E-05	-121.348	-1.266235E-03	-7.862717E-03
4	-105.164	7780.00	261081.	887290.	6.01000	-1.81634	-0.274024	-107.957	-8.064179E-04	-3211.41	-9.208089E-04	-11.8393	-1.284579E-04	-9.760942E-03
5	3.90627	-50.6715	3.75333	6.01000	0.600486	-4.803833E-02	-1.251545E-03	2.75140	9.127029E-06	7.942402E-02	-1.411267E-05	0.332923	3.397347E-06	2.011733E-05
6	0.496596	4.50571	5.06365	-1.81634	-4.803833E-02	0.183722	-2.487270E-04	0.348475	1.132305E-06	-0.105960	-1.845650E-06	4.229291E-02	4.308881E-07	2.635596E-06
7	-0.126735	-1.72681	-0.646858	-0.274024	-1.251545E-03	-2.487270E-04	0.222355	-8.989228E-02	-3.041467E-07	-2.139092E-02	4.717472E-07	-3.331387E-02	-3.848981E-07	-2.301360E-06
8	381.153	3897.63	-1071.27	-107.957	2.75140	0.348475	-8.989228E-02	250.937	6.786562E-04	67.2651	-1.078880E-03	25.2922	2.584176E-04	9.509675E-04
9	8.333315E-04	1.283299E-02	-5.194813E-03	-8.064179E-04	9.127029E-06	1.132305E-06	-3.041467E-07	6.786562E-04	2.821650E-09	2.356624E-04	-3.588451E-09	8.515101E-05	8.699049E-10	2.887065E-09
10	94.0038	904.322	-12143.8	-3211.41	7.942402E-02	-0.105960	-2.139092E-02	67.2651	2.356624E-04	103.111	-2.995447E-04	8.09422	8.283317E-05	-2.982097E-03
11	-1.528137E-03	-1.977631E-02	-7.155699E-05	-9.208089E-04	-1.411267E-05	-1.845650E-06	4.717472E-07	-1.078880E-03	-3.588451E-09	-2.995447E-04	5.429574E-09	-1.308984E-04	-1.336331E-09	-9.115427E-09
12	35.7137	475.420	-121.348	-11.8393	0.332923	4.229291E-02	-3.331387E-02	25.2922	8.515101E-05	8.09422	-1.308984E-04	13.2751	1.320840E-04	6.746671E-05
13	3.649065E-04	4.853034E-03	-1.266235E-03	-1.284579E-04	3.397347E-06	4.308881E-07	-3.848981E-07	2.584176E-04	8.699049E-10	8.283317E-05	-1.336331E-09	1.320840E-04	1.315503E-09	6.344654E-10
14	1.406541E-03	6.535410E-03	-7.862717E-03	-9.760942E-03	2.011733E-05	2.635596E-06	-2.301360E-06	9.509675E-04	2.887065E-09	-2.982097E-03	-9.115427E-09	6.746671E-05	6.344654E-10	6.287817E-05

SCALED W PARAM.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	8.726616E-02	6.928034E-02	1.637485E-04	2.438341E-05	4.551128E-03	1.142886E-03	-2.872042E-04	6.781410E-02	3.301548E-02	2.281394E-03	0.140934	3.273533E-02	3.292082E-02	2.069003E-04
2	6.928034E-02	0.258327	3.118931E-04	-1.615028E-04	-5.285584E-03	9.284002E-04	-3.503580E-04	6.208591E-02	4.551978E-02	1.964946E-03	0.163295	3.901500E-02	3.919896E-02	8.607042E-05
3	1.637485E-04	3.118931E-04	3.396261E-04	5.280108E-05	-3.814278E-06	-1.016488E-05	1.278625E-06	1.662492E-04	1.795183E-04	2.570676E-04	-5.756337E-06	9.701839E-05	9.964198E-05	1.008835E-06
4	2.438341E-05	-1.615028E-04	5.280108E-05	3.405307E-04	-1.159027E-05	6.919265E-06	1.027890E-06	3.179325E-05	5.288373E-05	1.290069E-04	-1.405681E-04	1.796263E-05	1.918279E-05	2.376636E-06
5	4.551128E-03	-5.285584E-03	-3.814278E-06	-1.159027E-05	5.819019E-03	-9.195552E-04	-2.359024E-05	4.071598E-03	3.007597E-03	1.603235E-05	1.082566E-02	2.538144E-03	2.549289E-03	2.461328E-05
6	1.142886E-03	9.284002E-04	-1.016488E-05	6.919265E-06	-9.195552E-04	6.946954E-03	-9.260868E-06	1.018651E-03	7.370496E-04	-4.225052E-05	2.796647E-03	6.369171E-04	6.386850E-04	6.369730E-06
7	-2.872042E-04	-3.503580E-04	1.278625E-06	1.027890E-06	-2.359024E-05	-9.260868E-06	8.152138E-03	-2.587449E-04	-1.949450E-04	-8.398732E-06	-7.038719E-04	-4.940107E-04	-5.617770E-04	-5.476746E-06
8	6.781410E-02	6.208591E-02	1.662492E-04	3.179325E-05	4.071598E-03	1.018651E-03	-2.587449E-04	5.670717E-02	3.415100E-02	2.073473E-03	0.126381	2.944575E-02	2.961181E-02	1.776758E-04
9	3.301548E-02	4.551978E-02	1.795183E-04	5.288373E-05	3.007597E-03	7.370496E-04	-1.949450E-04	3.415100E-02	3.161815E-02	1.617629E-03	9.360425E-02	2.207524E-02	2.219700E-02	1.201155E-04
10	2.281394E-03	1.964946E-03	2.570676E-04	1.290069E-04	1.603235E-05	-4.225052E-05	-8.398732E-06	2.073473E-03	1.617629E-03	4.335577E-04	4.786360E-03	1.285423E-03	1.294738E-03	-7.600099E-05
11	0.140934	0.163295	-5.756337E-06	-1.405681E-04	1.082566E-02	2.796647E-03	-7.038719E-04	0.126381	9.360425E-02	4.786360E-03	0.329693	7.899597E-02	7.937643E-02	8.828246E-04
12	3.273533E-02	3.901500E-02	9.701839E-05	1.796263E-05	2.538144E-03	6.369171E-04	-4.940107E-04	2.944575E-02	2.207524E-02	1.285423E-03	7.899597E-02	7.962216E-02	7.797490E-02	6.494026E-05
13	3.292082E-02	3.919896E-02	9.964198E-05	1.918279E-05	2.549289E-03	6.386850E-04	-5.617770E-04	2.961181E-02	2.219700E-02	1.294738E-03	7.937643E-02	7.797490E-02	7.643698E-02	6.010898E-05
14	2.069003E-04	8.607042E-05	1.008835E-06	2.376636E-06	2.461328E-05	6.369730E-06	-5.476746E-06	1.776758E-04	1.201155E-04	-7.600099E-05	8.828246E-04	6.494026E-05	6.010898E-05	9.712951E-03

EIGENVALUES

0.2993D-04	0.7148D-04	0.9311D-04	0.2980D-03	0.6162D-03	0.3569D-02	0.4472D-02	0.7345D-02
0.8149D-02	0.9715D-02	0.2928D-01	0.1054D+00	0.1538D+00	0.6286D+00		

EIGENVECTORS

Q	0.2156D-02	0.2140D+00	0.4440D+00	0.1329D-01	-0.5931D-01	0.1954D-01	0.7809D-02	0.1124D-02
	-0.2747D-03	0.5430D-02	-0.7625D+00	0.2255D+00	0.1519D+00	-0.3123D+00		
Q	-0.2593D-04	-0.3277D-02	0.3392D-02	-0.8954D-03	-0.1619D-02	-0.2682D-01	0.5211D-01	0.2128D-01
	0.6188D-04	0.1480D-02	-0.3910D-01	-0.2254D+00	-0.8466D+00	-0.4765D+00		
Q	0.4347D-02	0.5516D+00	-0.3639D+00	0.4317D+00	-0.6093D+00	-0.7195D-01	-0.1933D-01	0.1893D-02
	0.2973D-03	-0.7654D-03	-0.4497D-02	-0.1055D-02	-0.9749D-03	-0.5070D-03		
Q	0.1360D-02	0.1873D+00	-0.1173D+00	-0.8936D+00	-0.3888D+00	-0.3578D-01	-0.1319D-01	-0.1804D-02
	0.1571D-03	-0.1950D-03	-0.2124D-02	-0.1510D-03	0.7826D-03	0.2212D-03		
CH	-0.2149D-03	-0.2487D-01	0.1534D-01	0.1031D-02	-0.2421D-01	-0.2861D+00	0.8761D+00	0.3761D+00
	0.7454D-03	-0.3828D-02	0.3640D-01	0.3541D-01	0.6978D-01	-0.1466D-01		
CH	-0.2389D-04	-0.1071D-01	0.6527D-02	0.3285D-02	-0.1202D-01	-0.1148D+00	0.3586D+00	-0.9262D+00
	-0.3585D-02	0.1925D-02	0.5385D-02	0.3751D-02	0.4348D-02	-0.5475D-02		
CH	0.6726D-02	-0.1058D-03	-0.2123D-05	-0.4178D-05	0.2643D-03	0.1099D-02	0.1090D-02	-0.3599D-02
	0.1000D+01	-0.3007D-02	-0.3393D-03	0.3847D-02	-0.1968D-02	0.1721D-02		
T	-0.2187D-02	-0.3108D+00	-0.6619D+00	-0.2512D-01	0.1557D+00	-0.4487D+00	-0.1469D+00	-0.3329D-02
	0.4864D-03	-0.1084D-01	-0.3090D+00	0.1757D+00	0.1253D+00	-0.2745D+00		
RCH	0.2708D-02	0.2043D+00	0.4011D+00	0.4817D-02	0.4004D-01	-0.7262D+00	-0.2627D+00	-0.7733D-02
	0.1256D-02	-0.2914D-01	0.3852D+00	0.8307D-01	0.7609D-01	-0.1931D+00		
T	-0.2249D-01	-0.6892D+00	0.2328D+00	0.1194D+00	-0.6678D+00	-0.7227D-01	-0.6754D-01	0.1599D-02
	0.3965D-03	-0.1035D-01	0.5790D-03	0.4956D-02	0.7267D-02	-0.1014D-01		
RCH	-0.6183D-02	-0.1785D-01	-0.5594D-01	0.9147D-03	-0.3554D-01	0.4030D+00	0.7278D-01	-0.1240D-01
	0.1430D-03	0.6614D-02	0.4100D+00	0.3210D+00	0.2767D+00	-0.6926D+00		
T	-0.6967D+00	0.1445D-01	-0.1681D-02	-0.1505D-02	0.1040D-01	-0.9941D-02	-0.1541D-02	0.4705D-04
	0.8000D-02	0.1620D-02	-0.4501D-01	-0.6249D+00	0.2832D+00	-0.2036D+00		
RCH	0.7170D+00	-0.1380D-01	0.2748D-02	0.1232D-02	-0.6219D-02	-0.6709D-02	-0.2053D-02	0.1106D-03
	-0.1602D-02	0.5999D-03	-0.4025D-01	-0.6048D+00	0.2774D+00	-0.2033D+00		
KRB	0.5731D-03	-0.5223D-02	0.4626D-02	0.1183D-02	-0.4126D-02	-0.3044D-01	-0.7900D-02	0.3014D-02
	0.3053D-02	0.9994D+00	0.9595D-02	0.2878D-02	0.1878D-02	-0.1315D-02		

PARAMETER SUMMARY

PARAMETER # :	1	2	3	4	5	6	7	8	9	10	11	12	13	14
PARAMETER ID :	Q	Q	Q	Q	CH	CH	CH	T	RCH	T	RCH	T	RCH	KRB
FINAL VALUES	0.845E+02	0.944E+03	-0.969E+05	-0.510E+05	0.102E+02	0.514E+01	0.522E+01	0.665E+02	0.299E-03	0.488E+03	-0.128E-03	0.129E+02	0.131E-03	0.805E-01
STD. DEV.	0.250E+02	0.480E+03	0.179E+04	0.942E+03	0.775E+00	0.429E+00	0.472E+00	0.158E+02	0.531E-04	0.102E+02	0.737E-04	0.364E+01	0.363E-04	0.793E-02
COEF. OF VAR.	0.295E+00	0.508E+00	0.184E-01	0.185E-01	0.763E-01	0.833E-01	0.903E-01	0.238E+00	0.178E+00	0.208E-01	0.574E+00	0.282E+00	0.276E+00	0.986E-01
+ 2 STD. DEV.	0.134E+03	0.190E+04	-0.933E+05	-0.492E+05	0.117E+02	0.600E+01	0.617E+01	0.982E+02	0.405E-03	0.508E+03	0.190E-04	0.202E+02	0.204E-03	0.963E-01
- 2 STD. DEV.	0.346E+02	-0.156E+02	-0.100E+06	-0.529E+05	0.861E+01	0.429E+01	0.428E+01	0.348E+02	0.192E-03	0.467E+03	-0.276E-03	0.563E+01	0.586E-04	0.646E-01

CORRELATION MAT.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1.00000	0.461426	-3.007835E-02	-4.472944E-03	0.201963	4.641762E-02	-1.076793E-02	0.964002	0.628531	0.370897	-0.830883	0.392714	0.403084	7.106612E-03
2	0.461426	1.00000	-3.329819E-02	1.721937E-02	-0.136328	2.191560E-02	-7.634695E-03	0.512967	0.503671	0.185670	-0.559543	0.272038	0.278958	1.718279E-03
3	-3.007835E-02	-3.329819E-02	1.00000	0.155262	2.713232E-03	6.617659E-03	-7.684345E-04	-3.788263E-02	-5.478226E-02	-0.669921	-5.439898E-04	-1.865678E-02	-1.955645E-02	-5.554490E-04
4	-4.472944E-03	1.721937E-02	0.155262	1.00000	8.233614E-03	-4.498675E-03	-6.169253E-04	-7.234989E-03	-1.611669E-02	-0.335747	-1.326642E-02	-3.449648E-03	-3.759949E-03	-1.306799E-03
5	0.201963	-0.136328	2.713232E-03	8.233614E-03	1.00000	-0.144629	-3.425088E-03	0.224141	0.221731	1.009366E-02	-0.247158	0.117916	0.120877	3.273927E-03
6	4.641762E-02	2.191560E-02	6.617659E-03	-4.498675E-03	-0.144629	1.00000	-1.230605E-03	5.132265E-02	4.973147E-02	-2.434509E-02	-5.843670E-02	2.708125E-02	2.771646E-02	7.754400E-04
7	-1.076793E-02	-7.634695E-03	-7.684345E-04	-6.169253E-04	-3.425088E-03	-1.230605E-03	1.00000	-1.203419E-02	-1.214250E-02	-4.467396E-03	1.357698E-02	-1.939026E-02	-2.250487E-02	-6.154762E-04
8	0.964002	0.512967	-3.788263E-02	-7.234989E-03	0.224141	5.132265E-02	-1.203419E-02	1.00000	0.806522	0.418173	-0.924290	0.438214	0.449774	7.570660E-03
9	0.628531	0.503671	-5.478226E-02	-1.611669E-02	0.221731	4.973147E-02	-1.214250E-02	0.806522	1.00000	0.436905	-0.916796	0.439967	0.451517	6.854173E-03
10	0.370897	0.185670	-0.669921	-0.335747	1.009366E-02	-2.434509E-02	-4.467396E-03	0.418173	0.436905	1.00000	-0.400338	0.218779	0.224909	-3.703567E-02
11	-0.830883	-0.559543	-5.439898E-04	-1.326642E-02	-0.247158	-5.843670E-02	1.357698E-02	-0.924290	-0.916796	-0.400338	1.00000	-0.487566	-0.500017	-1.560070E-02
12	0.392714	0.272038	-1.865678E-02	-3.449648E-03	0.117916	2.708125E-02	-1.939026E-02	0.438214	0.439967	0.218779	-0.487566	1.00000	0.999508	2.335186E-03
13	0.403084	0.278958	-1.955645E-02	-3.759949E-03	0.120877	2.771646E-02	-2.250487E-02	0.449774	0.451517	0.224909	-0.500017	0.999508	1.00000	2.206033E-03
14	7.106612E-03	1.718279E-03	-5.554490E-04	-1.306799E-03	3.273927E-03	7.754400E-04	-6.154762E-04	7.570660E-03	6.854173E-03	-3.703567E-02	-1.560070E-02	2.335186E-03	2.206033E-03	1.00000

THE CORRELATION OF THE FOLLOWING PARAMETER PAIRS >= .95

PARAMETER #	ID	#	ID	CORRELATION
1	Q	8	T	0.96
12	T	13	RCH	1.00

THE CORRELATION OF THE FOLLOWING PARAMETER PAIRS IS BETWEEN .90 AND .95

PARAMETER #	ID	#	ID	CORRELATION
8	T	11	RCH	-0.92
9	RCH	11	RCH	-0.92

THE CORRELATION OF THE FOLLOWING PARAMETER PAIRS IS BETWEEN .85 AND .90

PARAMETER #	ID	#	ID	CORRELATION

```
RSQ (DEP.VAR. ONLY)------ =  24.146
RSQ (W/PARAMETERS)------- =  26.717
CALCULATED ERROR VARIANCE = 0.98952
CORRELATION COEFFICIENT-- = 0.99961
     W/PARAMETERS-------- = 0.99966
ITERATIONS--------------- =        5

MAX LIKE OBJ FUNC =  89.711
AIC STATISTIC---- =  117.71
BIC STATISTIC---- =  141.70
```

```
ORDERED DEPENDENT-VARIABLE WEIGHTED RESIDUALS
NUMBER OF RESIDUALS INCLUDED:        32
 -2.50     -1.44     -1.43     -1.31     -1.14     -1.02     -0.762    -0.597
-0.541     -0.454    -0.444    -0.441    -0.438    -0.380    -0.220    -0.844E-01
-0.591E-01-0.585E-01-0.491E-01 0.161     0.292     0.334     0.335     0.366
 0.394     0.786     0.824     0.825     1.03      1.04      1.06      1.28
CORRELATION BETWEEN ORDERED WEIGHTED RESIDUALS AND
INDEPENDENT NORMAL DEVIATES =    0.967
```

```
ORDERED WEIGHTED RESIDUALS
NUMBER OF RESIDUALS INCLUDED:        41
 -2.50     -1.44     -1.43     -1.31     -1.14     -1.02     -0.806    -0.762
-0.714     -0.597    -0.541    -0.454    -0.444    -0.441    -0.438    -0.380
-0.240     -0.220    -0.844E-01-0.683E-01-0.591E-01-0.585E-01-0.574E-01-0.491E-01
 0.443E-01 0.161     0.232     0.292     0.334     0.335     0.366     0.394
 0.761     0.786     0.824     0.825     0.844     1.03      1.04      1.06
 1.28
CORRELATION BETWEEN ORDERED WEIGHTED RESIDUALS AND
INDEPENDENT NORMAL DEVIATES =    0.969
```

APPENDIX B: GETTING STARTED AND RESOLVING PROBLEMS

Compiling and Loading MODFLOWP

MAIN of MODFLOW (McDonald and Harbaugh, 1988, p. 3-32) was changed substantially for MODFLOWP; a description of the changes and the code are presented in Appendix C. The main of MODFLOWP presented in this report includes call statements for Prudic's (1989) Streamflow-Routing Package and Hill's (1990b) PCG2 Package. The user needs to include these packages when compiling the program, or eliminate the call statements of any missing package.

The following six primary modules and one submodule from MODFLOW were modified for MODFLOWP: BAS1OT, BCF1RP, BCF1FM, RIV1FM, EVT1FM, DRN1FM, and SBCF1H. In addition, STR1FM from the Streamflow-Routing Package was modified. Although in most instances the modifications are minor, problems and confusion probably best can be avoided by eliminating these subroutines from an existing version of MODFLOW and by adding the versions presented in Appendix C. Usually this procedure is unnecessary, however, because the unchanged primary modules and submodules are included in distributed versions of MODFLOWP.

The program presented in this report used the IUNIT positions used in McDonald and Harbaugh (1988) for their packages, IUNIT(18) is used for the Streamflow-Routing Package, IUNIT(13) is used for the PCG2 Package, and IUNIT(15) is used for the Parameter-Estimation Package. To modify these, change references to IUNIT in MAIN, SEN1RW, SEN1RR, and SSEN1G (see Appendix C).

After substituting the new versions of MAIN and the primary modules and submodule discussed above and in Appendix C, including the FORTRAN of the new modules and submodules presented in Appendix C, adjusting the IUNIT positions, and allocating sufficient space in the X, PID, and DID arrays (see next section), the model can be compiled and loaded as usual, with the following exception. New submodule SPAR1E includes calls to subroutine DE2CSF and function DEPISF from IMSL (1987). If this library is available, it needs to be included when loading the program. If this library is not available, the statements in SPAR1E which include DE2CSF and DEPISF need to be commented out or deleted, and LPRINT of DATA SET 13 of the INPUT FILE cannot equal 1.

Space Requirements

In addition to the X-array, which includes space for all real and integer numbers used by MODFLOWP, the Parameter-Estimation Package uses two character arrays. These arrays are all dimensioned in the beginning of MAIN.

The X-array space needed by MODFLOWP without the Parameter-Estimation Package is the same as that needed by MODFLOW, as discussed by McDonald and Harbaugh (1988, Appendix B). To use the Parameter-Estimation Package, X-array storage needs to be increased as follows:

[NODES=NROW×NCOL×NLAY; see the data entry instructions for LINES 3, 5, and 6 of the the INPUT FILE or the listing of variables in Appendix C for the definitions of all other variable names]

A. 2×NODES + 2 + (4+NLLI1)×NP + (NMM+NZM)×NROW×NCOL + 2 + 13×NH + MAXM×MOBS + 4×NQ + NQT + 4×NQC + (NP+4)×(NH+NQT) + NSM×(3+LZI1) + 4×NSN + NRWD + NZER + NPNG.

B. If NPER>0 (NPER is specified in the Basic Package input file), the following additional X-array storage is needed:

(2×NPER)-1.

C. If IPAR≥0 and ISN>0 (the gradient is calculated using adjoint states), the following additional X-array storage is needed:

3×(NPER+1) + 2×NP.

If IPAR≥0 and ISN<0 (sensitivity-equation sensitivities are calculated) and NPER>0, the following additional X-array storage is needed:

NODES

D. If IPAR=1, parameter estimation is performed and the following additional X-array storage is needed:

(5+MPR)×NP + (NP+1)×MPR.

If ISN<0 or ISN=4 (the variance-covariance matrix is calculated), the following additional X-array storage is needed:

$2\times NP^2+2\times NP$.

If ISN=1 (quasi-Newton conjugate-direction iterations are used), the following additional X-array storage is needed:

$NP^2 + NP+2$.

If ISN>0 (a conjugate-direction optimization method is used), the following additional X-array storage is needed:

NP + 2×(NH+NQT) + 2×NP.

If the top model layer is a water-table layer and there is more than one layer in the model, the following additional X-array storage is needed:

NROW×NCOL.

The two character arrays are named PID and DID and are dimensioned in the beginning of MAIN. PID needs to be dimensioned to NP (LINE 3) or greater; DID needs to be dimensioned to NH+NQT (LINE 5) or greater. On some computers, failure to provide sufficient space in these arrays causes integer constants, such as IOUT or LCname (see "List of Variables" in Appendix C) to be changed to very large negative or positive values while executing the program.

Adapting the Residual Analysis Program for use with the Parameter-Estimation Package

To perform graphical analyses of weighted residuals (see section "Analysis of Results for Linear Problems"), realizations from the probability distribution function $N(\underline{0}, (\underline{\underline{I}} - \underline{\underline{X}} (\underline{\underline{X}}^T \underline{\underline{\omega}} \underline{\underline{X}})^{-1} \underline{\underline{X}}^T \underline{\underline{\omega}})\sigma^2)$ are needed. Such realizations can be produced by RESAN, a program presented by Cooley and Naff (1990, p. 176-183). The Parameter-Estimation Package can produce output that is compatible with a slightly modified version of RESAN, RESAN.MODP. In RESAN.MODP, the variance-covariance matrix on the parameters, the weights, and the sensitivities are read from a formatted instead of an unformatted file by making the following changes in the residual analysis program of Cooley and Naff (1990, p. 176-183).

(1) Add after format statement 16:
```
500 FORMAT (6F13.0)
```
(2) Change: `READ(ITB) (C(I,J),I=J,NVAR)`
To: `READ(ITB,500) (C(I,J),I=J,NVAR)`

(3) Change: `READ(ITB) (W(I),I=1,NOBS)`
To: `READ(ITB,20) (W(I),I=1,NOBS)`

(4) Change: `READ(ITB) (X(I,J),I=1,NVAR)`
To: `READ(ITB,20) (X(I,J),I=1,NVAR)`

In addition, the following changes can be made to avoid having to change dimensions of the program in multiple places when a larger or smaller problem is considered:

(1) Add before the first DIMENSION statement:

```
PARAMETER (NP=20, ND=70)
```

(2) Change first DIMENSION statement from:

```
DIMENSION X(20,20), COV(20,20), W(20), WP(20),
  IPR(20), R(90,90), D(90), G(90), F(90)
```

To:

```
DIMENSION X(NP,NP), COV(NP,NP), W(NP), WP(NP),
  IPR(NP), R(ND+NP, ND+NP), D(ND+NP), G(ND+NP), F(ND+NP)
```

(3) Change: NVD=20
NTD=70

To: NVD=NP
NTD=ND

Then, to change the dimensions of RESAN.MODP, just change NP and ND in the PARAMENTER statement so that they are at least as large as NP and NH+NQT of LINES 3 and 5 of the INPUT FILE of the problem considered.

Common Problems

1. Problem: Overshoot of a parameter to a very small absolute value.

Discussion: In the iteration after the overshoot, the regression routine might try to increase the absolute value of the parameter, but the calculated fractional change is huge because the absolute value of the parameter is so small. To respect the DMAX (DATA SET 13) limit on the maximum fractional change for any iteration, all changes are scaled by a small number by making AP (ρ_r of eq. 50b) small, and little improvement occurs in the sum of squared residuals (RSQ). (The value of AP is printed for each parameter-estimation iteration; values of RSQ are printed in a table at the end of the simulation.)

Resolution: Make the initial parameter value of the overshoot parameter smaller in absolute value.

2. Problem: Lack of convergence of parameters that have relatively small scaled sensitivity-equation sensitivities (printed when ISN<0 and ISCALS≠0; see LINE 6 of the INPUT FILE) or scaled gradient of the objective function (the gradient is printed; scale by multiplying by the parameter value).

Discussion: The parameter(s) that cause the lack of convergence of parameter estimation is(are) generally the parameter(s) for which the maximum fractional change (printed for each parameter-estimation iteration) is calculated in latter parameter-estimation iterations. Inspection indicates that these parameters are associated with relatively small scaled sensitivity-equation sensitivities or gradient of the objective function.

Resolution: Modified Gauss-Newton optimization with quasi-Newton updating, activated using NOPT of LINE 6 and NFIT of DATA SET 13, sometimes can solve these problems more effectively than the other methods. Alternatively, this situation was successfully resolved by R.M. Yager (U.S. Geological Survey, oral commun., 1990) by executing two runs of the parameter-estimation routine using modified Gauss-Newton optimization. In the first run, the relatively insensitive parameters were set to reasonable values, and only the relatively sensitive parameters were estimated. The resulting parameter estimates were used as initial estimates for the relatively sensitive parameters in the second run of the parameter-estimation program, in which all the parameters were estimated. This was accomplished conveniently by including the insensitive parameters in the last repetitions of DATA SET 2 and making IWPG of DATA SET 9 negative for these parameters for the first run. The IWPG values were changed to positive values for the second run. If these procedures are unsuccessful, consider reparameterizing to produce parameters that are more sensitive; see the section "Parameterization" in the main text of this report.

3. Problem: Parameter estimates are unreasonable (see item 1 in this section if the unreasonable parameter value is small in absolute value).

Discussion: Unrealistic parameter estimates can be produced if (a) there are inconsistencies between the dependent-variable data and the way the model is constructed or parameterized, or (b) the calculated values ($\hat{y}$) related to the dependent-variable observations (y) are insensitive to the estimated parameters.

Resolution: For (a), evaluate the calculated weighted and unweighted residuals and the scaled sensitivity-equation sensitivities (ISN>0; LINE 6 of the INPUT FILE). The dependent-variable observations that are most influential in producing unrealistic parameter estimates generally are associated with large weighted residuals and large scaled sensitivity-equation sensitivities for the parameter considered. Once these data are identified, evaluate the data, their weights, the model construction, and the parameterization for inconsistencies (see section "Adjustments Commonly Required During Parameter Estimation"). Make appropriate changes and try estimating the parameters again.

For (b), If the scaled sensitivity-equation sensitivities for a parameter are all small, reparameterize to produce parameters with larger sensitivities. This generally is accomplished by decreasing the number of parameters. Alternatively, prior information can be used to maintain parameters at reasonable values.

4. Problem: The solver does not function satisfactorily when calculating hydraulic heads, sensitivity-equation sensitivities, adjoint states, or the derivative of hydraulic head with respect to the iteration parameter of Newton's method. For iterative solvers, convergence is not reached in a reasonable number of iterations, execution is terminated, and an error message is printed, or there are unacceptably large errors (generally greater than 1 percent) in the volumetric budgets for the entire model. For direct solvers, excessive round-off errors produce an inaccurate solution, producing large errors in the volumetric budgets for the entire model.

Discussion: Parameter estimates in initial or subsequent parameter-estimation iterations can produce problems that are too difficult for most solvers. This difficulty happens most commonly when horizontal or vertical hydraulic-conductivity values in adjacent finite-difference cells are too different. For example, R.M. Yager (oral commun., 1990) had difficulty with Hill's (1990b) conjugate-gradient solver preconditioned with a modified incomplete Cholesky preconditioner when adjacent horizontal hydraulic-conductivity values differed by five orders of magnitude.

Resolution: Resolution depends on whether the problem occurs on the first or subsequent parameter-estimation iterations, and whether it occurs when calculating hydraulic heads or other quantities. The possibilities and suggested resolutions are as follows:

(A) Solver problems occur for the first parameter-estimation iteration.

Resolution: Change the initial parameter estimates.

(B) Solver problems occur for quantities other than hydraulic head for subsequent parameter-estimation iterations.

(a) The parameter values are unreasonable.

Resolution: See item 3 of this section.

(b) The parameter values are reasonable.

Resolution: Try changing how the solver is used. For example, consider changes in convergence criteria or other user-defined parameters. If problems persist, try improving the calibration by trial and error, using the Parameter-Estimation Package to easily change parameter values and provide information on the residuals. Attempt to use the Parameter-Estimation Package again after the model calibration has been improved.

(C) Solver problems occur for hydraulic heads for subsequent parameter-estimation iterations.

(a) The parameter values are unreasonable.

Resolution: See item 3 of this section.

(b) The parameter values are reasonable. This can occur, for example, if a fracture is being represented by a few rows or columns of a finite-difference grid, and the estimated hydraulic conductivity along the fracture is so much greater than the surrounding material that solver performance is adversely affected.

Resolution: First, consider using another solver; PCG2 (Hill, 1990b) or another preconditioned conjugate-gradient solver probably are most effective. Increasing the convergence criteria might allow the solver to converge, but check the errors in the volumetric budgets of the entire model to ensure that the solution is sufficiently accurate.

If changes to the solver do not work, another model needs to be considered. In the situation described above, an analytical or numerical model designed to represent flow through fractures needs to be considered. Parameter estimation probably will need to be accomplished by trial and error.

5. Problem: Sum-of-squared weighted residuals increases from one parameter-estimation iteration to the next.

Discussion:

(a) The sum-of-squared weighted residuals can increase when using the modified Gauss-Newton optimization because of overshoot. Overshoot occurs when the optimization procedure goes past the parameter values associated with the small value of the objective function for which it was aiming to parameter values associated with a large value of the objective function (fig. 17).

(b) The sum-of-squared weighted residuals also can increase because of problems 1 and 3 discussed above, in which case the reader is referred to those discussions. Alternatively, the increase might indicate that observations that had been omitted are included (see section "Omited Observations"). If the weighted residual of these observations is large, their alternating omission and inclusion can cause the sum of squared weighted residuals to fluctuate and to inhibit the convergence of parameter estimation.

Resolution: If the sum-of-squared weighted residuals increases for only a few iterations and the increase is small, no action is required. More persistent problems can be addressed by reducing the maximum fractional parameter change (DMAX of DATA SET 13). Note that DMAX $\leq$ 1.0 prohibits parameter values from changing sign.

Additional techniques for addressing 5.b were discussed in the section "Omitted Observations."

Restarting the Parameter-Estimation Routine

The parameter-estimation routine is restarted when estimated parameter values from one run are used as the initial parameter values in a subsequent run. A user might wish to restart a run if, for example, a parameter value becomes physically unreasonable. If the value is first unreasonable in the fifth parameter-estimation iteration, parameter values from the fourth iteration can be used to restart the run. See the preceding section entitled "Common Problems" for suggestions on how the restarted run could differ from the original run so that physically reasonable values are maintained.

The parameter estimates from each iteration of the parameter-estimation routine are saved on the output unit specified as IOUB on LINE 7 of the INPUT FILE, as discussed in the section "Parameter Values for Restarting Parameter Estimation." These values are printed to be compatible with the input format required for initial parameter values, and can be used to produce DATA SET 8 of the INPUT FILE for the restarted run.

When prior estimates of the parameters are specified by using DATA SET 10 of the INPUT FILE, the initial values are used as the prior estimates. When the parameter-estimation routine is restarted, the new initial values rarely equal the original initial values. To preserve the original prior estimates, specify them using repetitions of DATA SET 12. For example, to represent prior on the Ith parameter using DATA SET 12, set PRM(I) equal to 1.0, PRM(J), J≠I, equal to 0.0, and PRM(NP+1) equal to the prior estimate of the parameter. WP of DATA SET 12 needs to equal the old value of WP(I) of DATA SET 10, and WP(I) of DATA SET 10 needs to be set to zero.

Parameters also can be added or eliminated when restarting the parameter-estimation routine. This addition or deletion is accomplished by (1) adding or deleting negative values in DATA SET 9, or by (2) adding or subtracting appropriate repetitions of DATA SETS 2 and 2A or 2B. If the first method is used, remember that negative group numbers must follow all positive group numbers, and that the repetitions of DATA SET 2 may need to be rearranged. If the second method is used, add or eliminate numbers in DATA SETS 9, 10, and 12, and change the values on LINE 3, as needed.

Reducing Computer-Execution Time of Modified Gauss-Newton Optimization

Early iterations of modified Gauss-Newton optimization do not need hydraulic heads and sensitivities to be calculated accurately because inaccurate values still can produce substantial improvements in the estimated parameters. Accurate values, of course, are needed as the optimum parameter values are approached. When an iterative solver, such as PCG2 (Hill, 1990b) is used, larger solver convergence criteria can be used in the early parameter-estimation iterations to decrease the number of solver iterations and, thus, save computer time. Smaller convergence criteria needs to be used as the final parameter estimates are approached.

In this report, this change in solver convergence criteria is implemented with the user-specified variables FCONV and TOL (DATA SET 13), and the internally calculated variables AP and ADMX. FCONV is used to control the size of the solver convergence criteria for the first parameter-estimation iteration; TOL is the closure criteria for the parameter-estimation procedure; AP is equal to ρ_r of equation (50b), and ADMX is defined as discussed in the section "The Modified Gauss-Newton Method." AP times ADMX approaches TOL as the optimal parameter values are approached and, thus, can be used to indicate how close the parameter-estimation routing is to the final parameter estimates.

At the first parameter-estimation iteration, if FCONV is larger than zero, ADMX is set equal to 10×FCONV for the purpose of this calculation, AP is set equal to 1.0, and all convergence criteria are increased as follows:

$$CONV' = CONV\ (AP \times ADMX)/TOL, \qquad (B1)$$

where CONV is the original convergence criteria (from the solver input file or as calculated as described in the section "Sensitivity-Equation Method of Calculating Sensitivities" and printed as noted in the section "Printing and Checking of the INPUT FILE"), and CONV' is the increased convergence criteria. If, for example, FCONV = 5.0 and TOL = 0.01, the convergence criteria would be increased by a factor of 5,000.

After the first parameter-estimation iteration, AP and ADMX are calculated by the modified Gauss-Newton routine (AP equals ρ_r of eq. 54 and ADMX is calculated as in eq. 52), and equation (B1) is used to increase the convergence criteria. If (AP×ADMX)/TOL is less than 10, CONV' is set equal to CONV, the original convergence criteria; if (AP×ADMX)/TOL is greater than 10,000, CONV' is set to CONV×10,000.

Numerical experiments based on test case 1 of Appendix A that use this procedure with modified Gauss-Newton optimization indicate that computer execution times can be reduced by as much as 50 percent. Although it seems likely that a similar procedure might reduce the computer execution time for conjugate-direction optimization, this option is not available in the Parameter-Estimation Package.

APPENDIX C: PROGRAM DESCRIPTION AND FORTRAN LISTINGS

Program Description

McDonald and Harbaugh (1988) use the terms "primary module," "submodule," and "utility module" when referring to subroutines in MODFLOW, and their terminology is followed in this report. Only modules that were modified from those presented in McDonald and Harbaugh (1988) and Prudic (1989) or which are new are discussed in this section. For modified modules, only the modifications are discussed.

The modules of the Parameter-Estimation Package are referred to as new in the following discussion and are named with the label SEN if they are used to calculate sensitivity-equation sensitivities or the gradient of the objective function, and with the label PAR if they are used for parameter estimation. Use of two labels made it easier to organize the large number of modules and, hopefully, will make it easier for those who wish to modify or add to the Parameter-Estimation Package. Also, the SEN modules can be used without the PAR modules, as might be required for some applications. The remaining characters of the module names were assigned according to the conventions used by McDonald and Harbaugh (1988) where possible. The names of new utility modules begin with "u," except for those modified from Press and others (1989).

This documentation describes the function of each new primary module, submodule, and utility module, and the changes made to MAIN, existing primary modules, and to one existing submodule. The discussion is organized by what is calculated: calculation of hydraulic heads is presented first, calculation of sensitivity-equation sensitivities or the gradient of the objective function follows, and parameter estimation is described last. Flowcharts and outlines illustrate the logic the program follows and how the new primary modules fit into this logic. It is anticipated that a more detailed understanding of the program will be pursued only by users who are considering making modifications to the Parameter-Estimation Package and know FORTRAN, and that these users will examine the program itself. The program is written to be as straightforward as possible, and numerous comments are used to explain the purpose of each part of the program. A list of variables is included to help those interested in examining the program at this detailed level.

Hydraulic Heads

If the Parameter-Estimation Package is not used (IUNIT(15)=0), MODFLOWP functions like MODFLOW (McDonald and Harbaugh, 1988; and Prudic, 1989), except that if the solver does not converge, execution stops and no hydraulic heads, drawdowns, or budget terms are printed or saved. MODFLOW prints and saves the solution even if the solver does not converge. If the Parameter-Estimation Package is used (IUNIT(15)≠0), but only hydraulic heads are calculated (IPAR=-1 on LINE 6 of the INPUT FILE), the parameter values from DATA SET 8 and parameter definition from DATA SETS 2, 2A, and 2B of the INPUT FILE are used to replace the related model-input values read from the input files from other packages. This is accomplished by using new primary module SEN1FN, which is discussed in the next section. Calculation of hydraulic heads with the Parameter-Estimation Package is discussed in more detail in section "Calculating Hydraulic Heads and Flows" in Appendix A.

Sensitivity-Equation Sensitivities or the Gradient of the Objective Function

To calculate sensitivity-equation sensitivities or the gradient of the objective function, MAIN, 8 primary modules, and 1 submodule were modified, and there are 14 new primary modules, 17 new submodules, and 5 new utility modules.

<u>Modified MAIN</u>

Call statements for the new primary modules are added. Statements are added to coordinate the following: calculation of steady-state and transient hydraulic heads in a single model run; the calculation of sensitivity-equation sensitivities, or adjoint states and $\partial \underline{h}/\partial \rho_r$ (eq. 60).

<u>Modified primary modules</u>--Listed in order of first appearance in MAIN.

- BAS1OT Assigns KSTP as shown in table A3.
- BCF1RP Transmissivity and hydraulic conductivity are saved in array ST; if the top layer is unconfined, the vertical leakance beneath the top layer is saved in array SV.
- BCF1FM Two time-step lengths are used to formulate the matrix equations; they are unequal only when calculating adjoint states.

RIV1FM, EVT1FM, DRN1FM, STR1FM, GHB1FM For these five modules, hydraulic heads are stored in array SHNW when calculating adjoint states, $\partial \underline{h}/\partial \rho_r$ (eq. 60), or sensitivity-equation sensitivities with ISN=-2. When calculating adjoint states, no contributions are added to RHS.

Modified submodule

SBCF1H Hydraulic heads are stored in array SHNW when calculating adjoint states, $\partial \underline{h}(n)/\partial \rho_r$ (eq. 60), or sensitivity-equation sensitivities with ISN=-2. If cells went dry in the previous parameter-estimation iteration, HOLD for those cells is set equal to BOT+10.0.

New primary modules--Listed in order of first appearance in MAIN.

SEN1AL Reads LINES 1 through 8 of the INPUT FILE and allocates space for calculating sensitivity-equation sensitivities and adjoint states.

SEN1RP Calls SSEN1G to read, check, print, and store the part of the INPUT FILE that defines the parameters (DATA SETS 2, 2A, 2B, 3, and 4); reads DATA SET 5; calls SSEN1H and SSEN1F to read, check, print, and store observations from the INPUT FILE (DATA SETS 6, 6A, 6B, 6C, 7, 7A, 7B); reads initial parameter estimates and group numbers (DATA SETS 8 and 9); and calculates convergence criteria for sensitivity-equation sensitivities.

SEN1ST Reads the stress period data from the Basic Package input file and calculates and stores the time-step lengths.

SEN1RW Rewinds input files that are to be read again.

SEN1FM Reads hydraulic heads or sensitivity-equation sensitivities from the files defined using IUHEAD (LINE 7) and stores them in the appropriate model arrays.

SEN1RR Reads input files from other packages of the MODFLOW without printing anything.

SEN1FN Resets parameter values that are negative but need to be positive to be physically reasonable, calls SSEN1J, SSEN1K, and SSEN1M to calculate model-input values from the parameter values and DATA SETS 2, 2A, and 2B, and puts the new model-input values in model arrays. Rereads IBOUND arrays, if required.

SEN1RH Modifies hydraulic heads at constant hydraulic-head boundaries (see section "IUNHEA" of Appendix A).

SEN1AR Calls submodules SSEN1U to calculate the objective-function derivative required to calculate adjoint states (eq. 73).

SEN1AP Calls submodules SSEN1C, SSEN1D, SSEN1P, SSEN1R, and SSEN1N to calculate the gradient of the objective function from the adjoint states (eq. 75) and to calculate the RHS array required to calculate sensitivity-equation sensitivities (eq. 69 or 71).

SEN1NZ Checks arrays HNEW and RHS for nonzero values. If none are located, the matrix equation is not solved.

SEN1HZ Sets all elements of array HNEW to zero at time steps listed in DATA SET 1B when calculating hydraulic heads.

SEN1FD Sets IFLD, which, when ISN<0 and IPAR=0, is used to limit the printing and saving of cell-by-cell sensitivity-equation sensitivity arrays to time steps at which there are observations.

SEN1OT When ISN≠-1, writes hydraulic heads on files defined using IUHEAD (LINE 7); when ISN>0, stores the hydraulic heads from the last time step in SHNW; calls SSEN1U and SSEN1V to calculate, print, and save hydraulic heads or head-dependent boundary flows and sensitivity-equation sensitivities related to the NH+NQT observations; and calls SSEN1O to print weighted residuals for dependent-variable observations and prior parameter estimates and related statistics.

New submodules--Divided by function and then listed in alphabetical order.

The following five new submodules are used to read and prepare data.

SSEN1A Called by SEN1RP when ICHECK>0 (LINE 4) to check on multiple use of grid cells for parameters used to calculate model-input arrays.

SSEN1B Called by SSEN1G when PID=CH to calculate coefficients for linear interpolation.

SSEN1F Called by SEN1RP to read and prepare data for head-dependent boundary flow observations (DATA SETS 7, 7A, and 7B).

SSEN1G Called by SEN1RP to read and prepare data that defines the parameters (DATA SETS 2, 2A, 2B, 3, and 4).

SSEN1H Called by SEN1RP to read and prepare data for hydraulic-head observations (DATA SETS 6, 6A, 6B, 6C).

SSEN1I Called by SSEN1H to calculate interpolation coefficients for hydraulic heads from ROFF and COFF (fig. 6). Also called by SEN1FN and SEN1OT if cells used in the interpolation have gone dry.

Three new submodules are called by SEN1FN to put parameter values in model arrays from other packages and to check the parameter definitions.

SSEN1J Called when PID=CH to put new constant hydraulic-head values in HNEW.

SSEN1K Called when PID=Q, KRB, KDR, KST, or GHB to put new parameter values in model arrays.

SSEN1M Called when PID=KV, TKV, S1, RCH, or ETM to put new parameter values in model arrays.

Eight new submodules are used to calculate sensitivity-equation sensitivities or the gradient of the objective function. For the adjoint-state method, four new submodules are called by SEN1AP to calculate the matrix and vector derivatives of equation (75), multiply them by hydraulic heads and adjoint states, as needed, and sum the results to get the gradient. For the sensitivity-equation method, the same four new submodules calculate the right-hand side of equation (69) or the identical terms on the right-hand side of equation (71). The four submodules are:

SSEN1C Called when PID=CH;

SSEN1D Called when PID=Q, KRB, KDR, KST, or GHB;

SSEN1P Called when PID=T, KV, TKV, ANI, or S1; and

SSEN1R Called when PID=RCH or ETM.

One new submodule is called by SEN1AP when there are water-table layers to calculate the term on the right-hand side of equation (71) that is calculated using equation (72).

SSEN1N Called for water-table layers.

The following three new submodules are called by SEN1OT:

SSEN1O Prints weighted residuals for dependent-variable observations and prior parameter estimates and selected statistics.

SSEN1U Interpolates calculated hydraulic heads and associated sensitivity-equation sensitivities to locations not at cell centers, interpolates values temporally if needed, and calculates contributions to array RHS when calculating adjoint states. Recalculates interpolation coefficients if cells used in the interpolation have gone dry.

SSEN1V Calculates flows at head-dependent boundaries and associated sensitivity-equation sensitivities.

New utility modules - Listed in alphabetical order

U2DINN Called by SEN1RR to read integer arrays without printing anything. Very similar to U2DINT (McDonald and Harbaugh, 1988, p. 14-30).

U2DREN Called by SEN1RR to read real arrays without printing anything. Very similar to U2DREL (McDonald and Harbaugh, 1988, p. 14-26).

UNORM Called by SSEN1O to find the random variable from a N(0,1) probability distribution given a cumulative probability.

UREADF Reads a NROW×NCOL array from an unformatted file with multiple NROW×NCOL arrays.

USAVEF Puts elements of one array, which can be single or double precision, into another array, which also can be single or double precision.

Flowcharts for calculating sensitivity-equation sensitivities and the gradient of the objective function

The following flowcharts display the logic followed by the program when calculating sensitivity-equation sensitivities or the gradient of the objective function using the adjoint-state method. Selected primary module names are included at steps for which the modules are used.

Flowchart for calculating sensitivity-equation sensitivities with ISN =-1

START

↓

Allocate space (SEN1AL)

↓

Read stress period data and store time-step lengths (SEN1ST)

↓

Read and store up to and including DATA SET 9 of the INPUT FILE (SEN1RP)

↓

Start at time step 0 (steady state)

↓

yes ⟶ Read stress-period input (existing modules)

↓

Substitute parameter values (DATA SET 8) into model arrays (SEN1FN)

↓

Begin next stress period? ⟶ no ⟶ Calculate hydraulic heads for one time step, print and save them according to the Output Control input file, and store them in SHNW; store hydraulic heads from last time step in SHOLD (existing modules; SEN1FM)

↓

Initialize HNEW to zero, or to hydraulic head or the sensitivities for this parameter from the previous time step (SEN1FM)

↓

⟶ Calculate the right-hand side of equations (69) or (71) for this parameter for this time step (SEN1AP)

↓

Calculate the sensitivity of hydraulic head at this time step to this parameter for all active cells; print and save them according to the Output Control input file if there is an observation this time step, and store values on IUHEAD

↓

Calculate and save sensitivities for observations (SEN1OT)

↓

increment parameter ← no ← Last parameter?

↓ yes

increment time step ← no ← Last time step?

↓ yes

Print sensitivities for observations on output unit IOUT (SEN1OT)

↓

STOP

Flowchart for calculating sensitivity-equation sensitivities with ISN =-2

START

↓

Allocate space (SEN1AL)

↓

Read stress-period data and store time-step lengths (SEN1ST)

↓

Read and store up to and including DATA SET 9
of the INPUT FILE (SEN1RP)

↓

For all time steps: read stress-period input; substitute parameter
values from DATA SET 8 into model arrays; calculate hydraulic heads
and print and save them according to the Output Control input file,
and store them on IUHEAD (existing modules; SEN1FN, SEN1FM)

↓

→ Start at time step 0 (steady state)

↓

Initialize sensitivities (HNEW) to zero (SEN1FM)

↓

yes → Read stress-period input (existing modules; SEN1RR)

↓

Substitute parameter values from DATA SET 8
into model arrays (SEN1FN)

↓

Begin next stress period? → no → Read and store hydraulic heads from
last and present time steps (SEN1FM)

↓

Calculate the right-hand side of equations (69) or
(71) for this parameter for this time step (SEN1AP)

↓

Calculate the sensitivity of hydraulic head at this
time step to this parameter for all active cells;
print and save according to the Output Control input
file if there is an observation this time step

↓

Calculate and save sensitivities for observations (SEN1OT)

↓

increment time step ← no ← Last time step?

↓ yes

increment parameter ← no ← Last parameter?

↓ yes

Print sensitivities for observations on output unit IOUT (SEN1OT)

↓

STOP

Flowchart for calculating the gradient of the objective function using the adjoint-state method (ISN>0)

START

↓

Allocate space (SEN1AL)

↓

Read stress-period data and store time-step lengths (SEN1ST)

↓

Read, store, and print up to and including
DATA SET 9 of the INPUT FILE (SEN1RP)

↓

For all time steps: read stress-period input; substitute parameter values from DATA SET 8 into model arrays; calculate hydraulic heads and print and save them according to the Output Control input file, and store them on IUHEAD (existing modules; SEN1FN, SEN1FM)

↓

Print information for observations (SEN1OT)

↓

Start at time step n=N (latest time step at which a component of S($\underline{b}$) occurs)

↓

yes ⟶ Read and store stress-period input (SEN1RR)

↓

Substitute parameter values from DATA SET 8 into model arrays (SEN1FN)

↓

End of prior stress period? ⟶ no ⟶ Calculate contribution to RHS from last term of equation (73) (SEN1AR)

↓

Calculate adjoint state for time step n-1

↓

Read and store hydraulic heads from time steps n and n-1 (SEN1FM)

↓

Calculate contribution of this adjoint state to all components of the gradient (eq. 75) (SEN1AP)

↓

n=0? ⟶ no ⟶ increment time step backwards (n=n-1) ⟶ End of prior stress period?

↓ yes

Print gradient (SEN1AP)

↓

STOP

Parameter Estimation

To perform parameter estimation, MAIN was modified, and there are 7 new primary modules, 7 new submodules, and 3 new utility modules. PAR1AS, PAR1AQ, PAR1AP, and SPAR1P were developed from algorithms provided by Richard L. Cooley (U.S. Geological Survey, Lakewood, Colo., 1988); the 3 utility modules were developed from algorithms from Press and others (1989).

Modified MAIN

Call statements for the new primary modules were added, and statements were added to execute multiple parameter-estimation iterations.

New primary modules--Listed in order of appearance in MAIN

PAR1AL	Allocates space for the parameter-estimation calculations.
PAR1RP	Reads, checks, prints, and stores the part of the INPUT FILE used only for parameter estimation when IPAR>0 (DATA SETS 10 through 13 or 14).
PAR1AS	Calculates the search direction and prints data for each conjugate-direction nonlinear regression iteration. Checks for convergence and prints final statistics.
PAR1AQ	Calculates the iteration parameter, updates parameter estimates, and prints data for each conjugate-direction nonlinear regression iteration.
PAR1AP	Executes one iteration of nonlinear regression by using the modified Gauss-Newton method and prints data for the iteration.
PAR1OT	Final output for parameter estimation when the variance-covariance matrix for the parameter estimates (eq. 35) is printed.
PAR1RE	Calculates R_N^2 (eq. 38) and prints the ordered weighted residuals and R_N^2.

New submodules--Listed in alphabetical order

SPAR1A	Called by PAR1AS (ISN>0) or PAR1OT (ISN<0) to calculate R (eq. 37).
SPAR1E	Called by PAR1OT to calculate and print the eigenvalues and eigenvectors of the variance-covariance matrix on the parameters.

SPAR1I Complete calculation of the inverse of the scaled modified Gauss-Newton coefficient matrix starting with the decomposed matrix from SPAR1R.

SPAR1M Called by PAR1AS (ISN>0) or PAR1OT (ISN<0) to calculate statistics based on the maximum likelihood objective function (eq. 48, 64, and 65).

SPAR1P Called by PAR1AS (ISN>0), PAR1OT (ISN<0), or MAIN (if convergence is not achieved when solving for hydraulic heads, sensitivity-equation sensitivities, adjoint states, or $\partial \underline{h}/\partial \rho_r$) to read parameter values and statistics for all iterations from file IOUB and to print them on the output file.

SPAR1Q Called by PAR1AP to compute the component of the Hessian calculated with quasi-Newton updating and add it to the modified Gauss-Newton approximation of the Hessian.

SPAR1R Called by PAR1AP to compute parameter step lengths for modified Gauss-Newton optimization by using the Marquardt procedure.

Utility modules--Modified from Press and others (1989).

TRED2 Uses the Householder method to reduce a matrix to a tridiagonal form.

TQLI Uses the QL method to calculate eigenvalues and eigenvectors of a matrix using the output from TRED2.

EIGSRT Sorts the eigenvalues from smallest to largest and correspondingly rearranges the eigenvectors.

Outlines for Parameter-Estimation Calculations

The following outlines display the logic followed by the Parameter-Estimation Package when estimating parameters by modified Gauss-Newton or conjugate-direction optimization. Primary PAR module names are included at the step for which the modules are used.

Modified Gauss-Newton Optimization

1. Allocate space and read data required for parameter estimation (PAR1AL, PAR1RP).

2. Using the most recent parameter estimates, calculate hydraulic heads and sensitivity-equation sensitivities (see McDonald and Harbaugh, 1988, and previous section for documentation of the modules used).

3. Solve for scaled changes in parameters (eq. 50a).

4. Solve for new parameters (eq. 50b) (PAR1AP)

5. Check for convergence.

6. If not converged and not yet reached the maximum number of parameter-estimation iterations, return to (2) after replacing the old parameter values with the new parameter values (PAR1FM). If converged, print final output (PAR1OT and PAR1RE). If not converged and reached the maximum number of iterations, print a message that indicates the lack of convergence and a table with parameter estimates and iteration information, and stop (PAR1OT).

Conjugate-Direction Optimization

1. Allocate space and read data required for parameter estimation (PAR1AL, PAR1RP).

2. Using the most recent parameter estimates, calculate hydraulic heads and the gradient of the objective function (see McDonald and Harbaugh, 1988, and previous section for documentation of the modules used).

3. Calculate the sum of squared, weighted residuals. Check for convergence, and, if not converged and not yet reached the maximum number of iterations, calculate the direction vector (eq. 57 or 59) and proceed to step 4. If converged and the variance-covariance matrix on the parameter estimates is not to be calculated (DATA SET 13: LASTX=0), calculate and print the available statistics, print a table with parameter estimates and iteration information, and stop. (PAR1AS)

If convergence is reached and the variance-covariance matrix on the parameter estimates is to be calculated (LASTX>0), use the final parameter estimates to calculate hydraulic head and sensitivity-equation sensitivities (see McDonald and Harbaugh, 1988, and the previous section for documentation of the modules used), calculate and print the variance-covariance matrix and related statistics, print a table with parameter estimates and iteration information, and stop. (PAR1AP, PAR1OT, and PAR1RE)

If not converged and reached the maximum number of iterations, print a message that indicates the lack of convergence and a table with parameter estimates and iteration information, and stop (PAR1AS).

4. Calculate the $\partial \underline{h}/\partial \rho_r$ (eq. 60) (existing modules, SEN1FM, SEN1AP).

5. Calculate optimum step size and new model parameters (PAR1AQ).

6. Return to step 2.

Program Portability

This program is written in FORTRAN 77 in a style similar to that of MODFLOW. The only difference between the programming in the codes is the extensive use of block IF statements in the Parameter-Estimation Package. McDonald and Harbaugh (1988, appendix A) discuss portability in detail.

International Mathematical and Statistical Library (IMSL) routines DE2CSF and DEPISF are used to calculate eigenvalues and eigenvectors of the variance-covariance matrix in SPAR1E. If IMSL is not available, comment out the lines where these routines are used, and set LPRINT$\neq$1 (see DATA SET 13 in the section "Instructions for the INPUT FILE").

As discussed by McDonald and Harbaugh (1988, p. A-2 to A-4), in some circumstances it might be necessary to convert the flow code to double precision. Although there is no advantage to converting all regression variables and arrays to double precision, the conversion of the flow code probably is accomplished most easily if all of MODFLOWP is converted to double precision, using the instructions from McDonald and Harbaugh (1988, p. A-3). In addition, the intrinsic function DBLE(·), which is used in some of the new and modified subroutines of MODFLOWP, needs to be omitted from the code.

List of Variables

The following list includes most of the variables that occur in the FORTRAN program of the next sections; however, two types of variables are not included. These are: (1) variables that are used to store only the quantities that McDonald and Harbaugh (1988) and Prudic (1989) indicate they store, and (2) variables that are used briefly in a few calculations so their meaning can be clearly identified from a small part of the FORTRAN code. Note that HNEW and HOLD are included in the list below because they are used in the Parameter-Estimation Package to store other arrays besides the hydraulic head of the present and previous time step, as discussed in McDonald and Harbaugh (1988).

In the following list, the "Range" is listed as "SEN" and "PAR" for variables that are used only within SEN or PAR modules of the Parameter-Estimation Package, respectively. Variables used within both SEN and PAR modules are listed with a range of "Package". Variables used in any part of the Parameter-Estimation Package and in other parts of MODFLOWP are listed with a range of "Global". Variables that are used within a single module have that module listed under "Range".

Variable	Range	Definition
ADMX	Global	Absolute value of DMX.
ALPH	PAR1AQ	Iteration parameter calculated by Newton's method.
ATS	SEN	DIMENSION (2×(NPER+1)), sequence of time-step lengths used to calculate adjoint states.
AVE	SSEN10	Arithmetic average of the weighted residuals of observed hydraulic heads and drawdowns, observed head-dependent boundary gains and losses, or prior parameter estimates.
AVET	SSEN10	Arithmetic average of the weighted residuals of all observations and prior parameter estimates.
B	Package	DIMENSION (NP), updated parameter estimates.
B1	Package	DIMENSION (NP), original parameter estimates.
BSTP	SEN	DIMENSION (NPER+1), last time step of each pumping period, starting with the last. Used only for calculating adjoint-state sensitivities.
BUFF	Global	DIMENSION (NCOL, NROW, NLAY), general-purpose storage space used as required in many modules and sub-modules. Generally uses same storage space as RHS (McDonald and Harbaugh, 1988, p. 4-11).
C	PAR	DIMENSION (NP,NP), double precision coefficient matrix of the Gauss-Newton method, matrix $\underline{\underline{H}}$ of the quasi-Newton conjugate-direction method, or variance-covariance and correlation matrices of the parameters.
CELS	SEN	DIMENSION (4,NSN), cell information for parameters defined by lists of cells. Read from repetitions of DATA SET 2A.
COFF	SEN	DIMENSION (NH), column offsets which, with ROFF, define the locations of hydraulic-head observation points relative to cell centers. Read from repetitions of DATA SET 6.

Variable	Range	Definition
CONV	Global	DIMENSION (2,NP+1), solver convergence criteria for hydraulic head and sensitivity-equation sensitivities.
CSA	PAR	Search direction adjustment parameter used in Marquardt procedure (CSA=0.08 is suggested by Cooley and Naff, 1990, p. 71-72, 91).
DD	PAR	DIMENSION (NP), double-precision array of the calculated changes in parameters.
DELT1	Global	Time-step length used to calculate $\underline{\underline{B}}(n+1)$ of equation (23).
DID	Global	DIMENSION (ND), character array used to store labels for each of the ND dependent-variable observations.
DMAX	PAR	Maximum allowed fractional change for parameter values in one iteration.
DMX	PAR	Maximum calculated fractional change for any parameter value in one parameter-estimation iteration.
EV	PAR	Estimated common error variance.
FCONV	Global	User-defined factor that produces coarser solver convergence criteria when calculating heads and sensitivity-equation sensitivities for the first parameter-estimation iterations.
G	PAR	DIMENSION (NP), double-precision array of the gradient of the objective function with respect to each of the NP parameters.
H	Package	DIMENSION (ND), calculated dependent-variable values.
HNEW	Global	DIMENSION (NODES), double precision array of hydraulic heads, sensitivity-equation sensitivities, adjoint states, or the derivative of hydraulic heads with respect to the iteration parameter from Newton's method. Always for the present time step.
HOBS	Package	DIMENSION (ND), observed dependent-variable values.
HOLD	Global	DIMENSION (NODES), hydraulic heads, sensitivity-equation sensitivities, adjoint states, or the derivative of hydraulic heads with respect to the iteration parameter from Newton's method. Always for the prior time step. When calculating adjoint-state sensitivities, HOLD in module SEN1AP contains the hydraulic heads from the previous time step.

Variable	Range	Definition
IBT	SEN	DIMENSION (2,NQ), IBT(1,NQ) indicates the package used to simulate the head-dependent boundary for which a gain or loss is observed: IBT(1,NQ) — Package 1 — River 2 — General-Head Boundary 3 — Stream 4 — Drain IBT(2,NQ) equals zero unless one of the NP parameters is used to calculate the conductance of the head-dependent boundary, when IBT (2,NQ) equals the number of that parameter.
ICH	SEN	Counter for cells included in DATA SET 2A for PID=CH.
ICHECK	SEN	If ICHECK >0, model inputs determined with PID=T, KV, TKV, S1, RCH, and ETM are checked to ensure that that cells do not occur more than NUMR times.
IDIS	SEN	Number of head-dependent boundary gain or loss observations eliminated because the head-dependent boundary is no longer calculated by using the simulated hydraulic head in the aquifer.
IDRY	SEN	Number of hydraulic-head observations eliminated because cells in water-table layers have gone dry.
IFLCH	Global	Counter for parameters included for PID=CH.
IFLCH1	MAIN	Saves previous value of IFLCH, as required when an iterative solver is used.
IFO	Global	Flag that equals one when parameter estimation has converged.
IHESS	SEN	When ISN>0, IHESS=0 when adjoint states are calculated, and IHESS=1 when the derivative of hydraulic heads with respect to the iteration parameter of Newton's method is calculated.
IOFF	SEN	DIMENSION (NH), used to identify neighboring cells that are to be used to interpolate hydraulic heads of observation locations.
IOUR	PAR	Output unit for unformatted data to be used by RESAN, a separate FORTRAN program used to analyze residuals.

Variable	Range	Definition
IP	Global	IP=0 when hydraulic heads are calculated. Positive, nonzero values of IP equal the parameter numbers when ISN<0, and in SEN1RP, PAR1RP, SEN1AP, SEN1OT, and SSEN1C when ISN>0. In other modules and submodules used when ISN>0, IP=1 when adjoint states are calculated, and IP=2 when the derivative of hydraulic heads with respect to the iteration parameter of Newton's method is calculated.
IPP	MAIN	Flag that equals one when a new pumping period has been reached.
IPAR	Package	Flag that controls program execution: IPAR — Result <0 — Hydraulic heads are calculated 0 — Hydraulic heads and sensitivity-equation sensitivities or the gradient of the objective function are calculated. 1 — Hydraulic heads and sensitivity-equation sensitivities or the gradient of the objective function are calculated and parameter-estimation is performed. For more information, see the definition in the INPUT FILE instructions (LINE 6).
IPRC	PAR	Format code with which the covariance and correlation matrices are printed when parameter estimation converges and the modified Gauss-Newton coefficient matrix is calculated.
IPRINT	PAR	See definition in the input instructions for the last DATA SET 13 of the INPUT FILE.
IPRIOR	PAR	All prior information on the parameters is disregarded if IPRIOR=0.
IQOB	SEN	DIMENSION (NQT), time steps of observed fluxes at head-dependent boundaries.
IRBOT	SSENIV	Flag that controls the printing preceding and following the list of cells or reaches affected by nonlinearities when calculating head-dependent boundary gains and losses along observed reaches.
IRWD	SEN	DIMENSION (NRWD), FORTRAN input unit numbers that are identified from within the Recharge or Evapotranspiration Packages by LOCAT.

Variable	Range	Definition
ISCALS	SEN	If = 0, actual sensitivity-equation sensitivities are printed. If $\neq 0$, printed sensitivities are scaled by multiplying by the parameter value and multiplying by the square root of the weight.
ISN	Package	Flag identifying which numerical and optimization methods are used. See the definition in the INPUT FILE instructions (LINE 6).
ISUM	Global	Index number of the lowest element in the X array that has not yet been allocated. When space is allocated for an array, the size of the array is added to ISUM.
ITMXP	PAR	Maximum number of parameter-estimation iterations.
ITT	SEN	Flag for transient observations: ITT — Description 1 — Observed hydraulic heads are used for parameter estimation. 2 — Initial observed hydraulic heads and subsequent changes in hydraulic head at each location are used for parameter estimation.
IUBD	SEN	If IUBD>0, IBOUND arrays are read from FORTRAN unit IUBD.
IUHEAD	SEN	Output/input unit for hydraulic heads or sensitivity-equation sensitivities stored in binary format.
IUNHEA	SEN	Flag indicating that known hydraulic heads at constant hydraulic-head boundaries change during the simulation.
IWP	PAR	Flag indicating how the weights are to be calculated from the user-provided data for prior information on the parameters. See discussion for DATA SETS 10, 11, and 12 of the INPUT FILE instructions.
IWPG	PAR	Dimension (NP), parameter group numbers read from DATA SET 9 of the INPUT FILE. Weighted residuals for prior estimates of single parameter values are printed out by groups. Negative values in the last entries of th IWPG array indicate that the related parameters are not estimated.
IWT	SEN	Flag indicating how the weights are to be calculated from the user-provided data for dependent-variable observations. See discussion for DATA SETS 6 and 7A of the INPUT FILE instructions.

Variable	Range	Definition
JDRY	Package	Number of hydraulic-head observations interpolations that have been changed because cells in water-table aquifers have gone dry.
JOFF	SEN	DIMENSION (NH), used to identify neighboring cells that are to be used to interpolate hydraulic heads of observation locations.
JRBOT	SSENIV	Flag that controls printing of explanation about the effect of nonlinearities in head-dependent boundary flows.
JT	Global	Latest time step required for calculating dependent-variable values related to observations.
KNST	Global	Used in SEN1FM when calculating adjoint states to indicate that all components of HNEW are to be set to zero (KNST=1), hydraulic heads from the previous time step are to be read and stored in HOLD (KNST=-1), or the hydraulic heads in HOLD are to be stored in SHNW (KNST=0).
KNT	Global	Number of times the change in parameters calculated by the conjugate-direction method has been diminished because the value of the objective function was not reduced.
KPER	Global	The time step plus one. At steady state, KPER=1, at time step 1, KPER=2, and so on.
KPRINT	PAR	Controls printing of final hydraulic-head array and sensitivity-equation sensitivity ratios (ISN<0 only) when parameter estimation converges.
KRBOT	SSENIV	The number of cells in a head-dependent boundary observation group at which flow is not calculated using calculated hydraulic head or any parameter. The head-dependent boundary observation is eliminated if KRBOT=NQCL.
KSTP	Global	Always equals 1 when using the Parameter-Estimation Package. See KPER for the time-step number.
LASTX	PAR	See definition in the input instructions for the last DATA CARD of the INPUT FILE.
LCname	Package	Location in the X array (McDonald and Harbaugh, 1988, p. 3-22) of the first element of array 'name'. If name is longer than four letters, only the first four letters are used.

Variable	Range	Definition
LM	SEN	DIMENSION (NSM), multiplication array numbers. Read from DATA SET 2B.
LN	Package	DIMENSION (NP), flag that indicates whether the log-transform of a parameter is to be considered. If LN(I)=0, no log-transform is applied to the Ith parameter.
LPRINT	PAR	Controls calculation and printing of eigenvectors and eigenvalues of the scaled variance-covariance matrix on the parameters.
LZ	SEN	DIMENSION (LZI1+1,NSM), LZ(1) is a zone array number; LZ(I,NM), I=2,LZI1 are zone numbers; NM is incremented from 1 to NSM. Read from DATA SET 2B. LZ(1) is called LZA in the INPUT FILE instructions.
LZI1	SEN	Used to dimension array LZ. Read on LINE 3 of the INPUT FILE; set to the value read plus 1 in MODFLOWP.
MATZ	SEN	DIMENSION (NCOL, NROW, NZM). Zone arrays used in parameterization.
MAXM	Package	Maximum number of layers used for any of the MOBS locations.
MLAY	Package	DIMENSION (MAXM,MOBS), layers of a multilayer observation well. Read from DATA SET 6A of the INPUT FILE.
MOBS	SEN	Number of the NH locations and times at which observed hydraulic head or changes in hydraulic head are multilayer.
MPR	Package	Number of equations of prior information on the parameters that include more than one parameter.
ND	Package	Total number of dependent-variable observations: ND=NH+NQT.
NDER	SEN	DIMENSION (5,NH), layer, row, column, and time step of hydraulic-head observations, and the number of the first observation, which is used to calculate changes in hydraulic head for ITT=2.
NFIT	PAR	Number of Fletcher-Reeves iterations used in combined Fletcher-Reeves/quasi-Newton optimization.

Variable	Range	Definition
NH	Package	Number of locations and times at which there are observed hydraulic heads or changes in hydraulic heads.
NLL	SEN	DIMENSION (NLLI1,NP), used to define the parameters to which sensitivities are to be calculated or which are to be estimated. Use depends on PID of parameter involved. See description of DATA SET 2 of the INPUT FILE.
NLLI1	SEN	Largest required first dimension for array NLL. Read on LINE 3 of the INPUT FILE.
NLLIT	SEN	Calculated as (NLLI1-1)/2, and used as described in the INPUT FILE instructions for DATA FILE 2.
NM	Global	Counter that is incremented from one to NSM.
NMM	SEN	Number of multiplication arrays in DATA SET 3 of the INPUT FILE.
NNEG	SSEN10	Number of negative weighted residuals for one type of observation.
NNEGT	SSEN10	Number of negative weighted residuals for all observations.
NODES	Global	Total number of nodes in the finite-difference grid. Equals the product of the number of rows, columns, and layers.
NP	Package	Read from LINE 3. Changed in MODFLOWP to the number of estimated parameters.
NP1	Package	Original value of NP read from LINE 3; includes estimated and unestimated parameters.
NPER	Global	Value read from the Basic Package input file needs to be equal to or greater than JT+1. In MAIN of the Parameter-Estimation Package, NPER is set equal to JT+1 after the INPUT FILE is read.
NPOS	SSEN10	Number of positive weighted residuals for one type of observation.
NPOST	SSEN10	Number of positive weighted residuals for all observations.
NPR	Package	Number of prior estimates of single parameters. Equals the number of nonzero values in DATA SET 10 of the INPUT FILE. In the text of this report, NPR is used to indicate the sum of NPR and MPR as they are defined in the appendices.

Variable	Range	Definition
NQ	SEN	Number of cell groups for which the sum of cell-by-cell, head-dependent flows are observed.
NQC	SEN	Total number of individual cells, or nodes, in the NQ cell groups.
NQCL	SEN	DIMENSION (NQ), number of cells included in each of the NQ cell groups.
NQOB	SEN	DIMENSION (NQ), number of head-dependent boundary flux observations for each of the NQ groups.
NQT	Package	Total number of head-dependent boundary flux observations.
NRP	Global	Number of pumping periods of data to be read to reach the pumping period in questions. Equals one except when calculating adjoint states.
NRWD	SEN	Number of FORTRAN input units identified from within the Recharge and Evapotranspiration Packages by LOCAT.
NS	Global	Counter that is incremented from one to NSN.
NSM	SEN	Number of NLL values larger than zero for the T, KV, TKV, S1, RCH, and ETM parameter identifiers in DATA SET 2 of the INPUT FILE.
NSN	SEN	Number of cell locations listed for parameter definition in the INPUT FILE (equals the sum of the absolute values of NLL(1) for all GHB, Q, KRB, and CH parameter identifiers in DATA SET 2).
NUMR	SEN	See explanation for ICHECK.
NZM	SEN	Number of zone arrays in DATA SET 4 of the INPUT FILE.
PID	Package	DIMENSION (NP), character variable used to identify parameter type (see table A1).
PR	SEN	DIMENSION (5,MOBS), proportion assigned to each layer of a multilayer observation well. Read from DATA SET 6A.
PRM	PAR	DIMENSION (NP+1), coefficients and prior estimate for prior information on the parameters that can include more than one parameter. Read from DATA SET 12.
PV	PAR1AQ	DIMENSION (ND), double-precision array of derivatives of observed dependent variables with respect to the iteration parameter of Newton's method.

Variable	Range	Definition
QCLS	SEN	DIMENSION (4,NQC), layer, row, and column of cells in the head-dependent boundary groups. The fourth position is zero unless one of the NP parameters being estimated is used to calculate the conductance of this head-dependent boundary. Then, information required to calculate the derivative is stored in the fourth position.
RINT	SEN	DIMENSION (4,NH), values of interpolation basis functions for hydraulic-head observation locations.
ROFF	SEN	DIMENSION (NH), row offsets that, with COFF, define the locations of observation points relative to cell centers. Read from repetitions of DATA SET 6.
RSQ	Package	Sum of squared weighted residuals for dependent-variable observations.
RSQP	Package	Sum of squared weighted residuals for dependent-variable observations and prior parameter estimates.
SCL	PAR	DIMENSION (NP), user-defined parameter scaling factors used in conjugate-direction optimization methods.
SCLE	PAR	DIMENSION (NP), double precision array of the diagonal components of the unscaled coefficient matrix of the Gauss-Newton method.
SFAC	SEN	DIMENSION (NSM), multiplication factor used to define parameters.
SHOLD	SEN	DIMENSION (NODES), hydraulic-head array from previous time step. Used only when calculating sensitivity-equation sensitivities.
SHNW	SEN	DIMENSION (NODES), hydraulic-head array for current time step. Used only when calculating sensitivity-equation sensitivities, adjoint-state sensitivities, or the derivative of hydraulic head with respect to the iteration parameter from Newton's method.
SMAT	SEN	DIMENSION (NCOL, NROW, NMM), multiplication matrices used in parameterization. Read from DATA SET 3.
SSG	PAR1AS	Double-precision scalar variable of the sum of the squared components of the gradient; used in the Fletcher-Reeves method, equation (57).
SSG1	PAR1AS	Single precision scalar variable used to save the previous value of SSG, as required for equation (57).

Variable	Range	Definition
ST	SEN	DIMENSION (NODES), transmissivity or hydraulic-conductivity values for each cell.
STP	SEN	DIMENSION (NPER+1), first time step of each pumping period, starting with the first.
TOFF	SEN	DIMENSION (ND), time offsets that define the time of dependent-variable observations relative to the end of the preceding time step.
TMEAN	SEN1OT	Sum of the absolute values of all scaled sensitivities divided by ND×NP.
TOL	PAR	Parameter-estimation closure criterion. Solution is reached when the convergence criteria described in the text for the optimization methods are less than or equal to TOL.
TS	SEN	DIMENSION (NPER), or DIMENSION (1) when NPER=0, time-step lengths.
VMAX	SSEN1O	Maximum weighted residual for one type of observation.
VMIN	SSEN1O	Minimum weighted residual for one type of observation.
WP	PAR	DIMENSION (NP+MPR), weights of prior information on the parameters.
WT	Package	DIMENSION (ND), weights of the observed hydraulic heads, changes in hydraulic heads, and head-dependent flows.
X	Package	DIMENSION (NP,ND) calculated sensitivity-equation sensitivities of observed dependent variables. Used only for modified Gauss-Newton regression (ISN<0 or LASTX>0).
XD	PAR1AS	Double-precision scalar variable equal to $(\underline{g}_{r+1}-\underline{g}_r)^T\underline{d}_r$ (terms are defined after eq. 57).
XX	Package	DIMENSION (NP), double-precision array of the gradient of the objective function. Used only for conjugate-direction optimization (ISN>0).

FORTRAN Listings

Modified MAIN

```
C     ******************************************************************
C     MAIN CODE FOR MODULAR MODEL --    9/1/87
C          BY MICHAEL G. MCDONALD AND ARLEN W. HARBAUGH
C-----VERSION 1638 24JUL1987 MAIN1
C     ******************************************************************
C     MODIFIED FOR ADJOINT SENSITIVITY CALCULATIONS             9/1/87
C          BY MARY C. HILL AND RUSSELL NELSON, WRD, USGS, TRENTON NJ.
C     ******************************************************************
C     MODIFIED FOR PARAMETER ESTIMATION BY MARY C. HILL         2/1/92
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 PID(30),DID(600)
      COMMON X(900000)
      DIMENSION HEADNG(32),VBNM(4,20),VBVL(4,20),IUNIT(24)
      DOUBLE PRECISION DUMMY
      EQUIVALENCE (DUMMY,X(1))
C     ------------------------------------------------------------------
C
C1------SET SIZE OF X ARRAY. REMEMBER TO REDIMENSION X.
      LENX=900000
C
C2------ASSIGN BASIC INPUT UNIT AND PRINTER UNIT.
      INBAS=5
      IOUT=6
C
C3------DEFINE PROBLEM__ROWS,COLUMNS,LAYERS,STRESS PERIODS,PACKAGES
      CALL BAS1DF(ISUM,HEADNG,NPER,ITMUNI,TOTIM,NCOL,NROW,NLAY,
     1            NODES,INBAS,IOUT,IUNIT)
C-------INITIALIZE SCALARS USED TO ALLOCATE SPACE IN X ARRAY FOR ARRAYS
C-------THAT MAY NOT BE USED, BUT ARE INCLUDED IN CALLS FOR THE
C-------PARAMETER-ESTIMATION PACKAGE.
      LCWELL=1
      LCDRAI=1
      LCIRCH=1
      LCRECH=1
      LCIEVT=1
      LCEVTR=1
      LCEXDP=1
      LCSURF=1
      LCRIVR=1
      LCSTRM=1
      ICSTRM=1
      LCTBAR=1
      LCIVAR=1
      LCBNDS=1
      LCST=1
      LCSV=1
      LCC=1
      LCSCLE=1
      LCG=1
      LCDD=1
      LCWP=1
      LCB1=1
      LCB=1
      LCPRM=1
      LCIWPG=1
      LCSCL=1
      LCSHNW=1
C-------SET VARIABLES TO ZERO
      MXDRN=0
      MXSTRM=0
      MXBND=0
      MXRIVR=0
      KNT=0
      JDRY=0
      IDRY=0
      KPRINT=0
      LPRINT=0
      NPR=0
      MPR=0
C
C4------ALLOCATE SPACE IN "X" ARRAY.
      CALL BAS1AL(ISUM,LENX,LCHNEW,LCHOLD,LCIBOU,LCCR,LCCC,LCCV,
     1            LCHCOF,LCRHS,LCDELR,LCDELC,LCSTRT,LCBUFF,LCIOFL,
     2            INBAS,ISTRT,NCOL,NROW,NLAY,IOUT)
      IF(IUNIT(1).GT.0) CALL BCF1AL(ISUM,LENX,LCSC1,LCHY,
     1     LCBOT,LCTOP,LCSC2,LCTRPY,IUNIT(1),ISS,
```

```
     2      NCOL,NROW,NLAY,IOUT,IBCFCB)
      IF(IUNIT(2).GT.0) CALL WEL1AL(ISUM,LENX,LCWELL,MXWELL,NWELLS,
     1                IUNIT(2),IOUT,IWELCB)
      IF(IUNIT(3).GT.0) CALL DRN1AL(ISUM,LENX,LCDRAI,NDRAIN,MXDRN,
     1                IUNIT(3),IOUT,IDRNCB)
      IF(IUNIT(8).GT.0) CALL RCH1AL(ISUM,LENX,LCIRCH,LCRECH,NRCHOP,
     1             NCOL,NROW,IUNIT(8),IOUT,IRCHCB)
      IF(IUNIT(5).GT.0) CALL EVT1AL(ISUM,LENX,LCIEVT,LCEVTR,LCEXDP,
     1             LCSURF,NCOL,NROW,NEVTOP,IUNIT(5),IOUT,IEVTCB)
      IF(IUNIT(4).GT.0) CALL RIV1AL(ISUM,LENX,LCRIVR,MXRIVR,NRIVER,
     1             IUNIT(4),IOUT,IRIVCB)
      IF(IUNIT(7).GT.0) CALL GHB1AL(ISUM,LENX,LCBNDS,NBOUND,MXBND,
     1             IUNIT(7),IOUT,IGHBCB)
      IF(IUNIT(9).GT.0) CALL SIP1AL(ISUM,LENX,LCEL,LCFL,LCGL,LCV,
     1           LCHDCG,LCLRCH,LCW,MXITER,NPARM,NCOL,NROW,NLAY,
     2           IUNIT(9),IOUT)
      IF(IUNIT(11).GT.0) CALL SOR1AL(ISUM,LENX,LCA,LCRES,LCHDCG,LCLRCH,
     1            LCIEQP,MXITER,NCOL,NLAY,NSLICE,MBW,IUNIT(11),IOUT)
      IF(IUNIT(18).GT.0) CALL STR1AL(ISUM,LENX,LCSTRM,ICSTRM,MXSTRM,
     1                  NSTREM,IUNIT(18),IOUT,ISTCB1,ISTCB2,NSS,NTRIB,
     2                    NDIV,ICALC,CONST,LCTBAR,LCTRIB,LCIVAR)
      IF(IUNIT(13).GT.0) CALL PCG2AL(ISUM,LENX,LCV,LCSS,LCP,LCCD,
     1        LCHCHG,LCLHCH,LCRCHG,LCLRCH,MXITER,ITER1,NCOL,NROW,NLAY,
     2        IUNIT(13),IOUT,NPCOND)
      IF(IUNIT(15).NE.0) CALL SEN1AL(ISUM,LENX,NCOL,NROW,NLAY,NPER,IOUT,
     1           LCSHNW,LCNLL,LCSMAT,LCCELS,IUHEAD,NP,NSM,NSN,ISN,LCTS,
     2           LCATS,LCST,NH,LCNDER,LCSHOL,LCSTP,NRWD,LCIRWD,IUNIT(15),
     3           LCBSTP,IPAR,LCB,LCH,LCX,LCRINT,LCJOFF,LCIOFF,LCCONV,
     4           LCCOFF,LCROFF,LCMLAY,LCPR,MOBS,LCLN,NQ,NQC,NQT,LCIBT,
     5           LCNQOB,LCNQCL,LCIQOB,LCQCLS,IUBD,ISCALS,LCTOFF,MPR,
     6           LCXX,MAXM,LCLZ,NLLI1,NLLIT,NMM,NZM,LZI1,LCSFAC,LCLM,
     7           LCMATZ,LCHOBS,LCWT,IOUB,LCSV,IPRNP,IUNHEA,NUMR,ICHECK,
     8           NPNG,LCIPNG,NZER,LCIZER,LCIWPG,NOPT,LCXD)
      ND=NH+NQT
      IF(IUNIT(15).GT.0.AND.(IPAR.GE.1.OR.ISN.GT.0)) CALL PAR1AL(ISUM,
     1           LENX,IOUT,NP,NH,LCC,LCSCLE,LCG,LCDD,LCWP,LCB1,NQT,MPR,
     2           LCPRM,ISN,LCSCL,LCPV,ND,LCR,LCU,LCGD,LCS)
C
C5------IF THE "X" ARRAY IS NOT BIG ENOUGH THEN STOP.
      IF(ISUM-1.GT.LENX) STOP
C
C6------READ AND PREPARE INFORMATION FOR ENTIRE SIMULATION.
      CALL BAS1RP(X(LCIBOU),X(LCHNEW),X(LCSTRT),X(LCHOLD),
     1        ISTRT,INBAS,HEADNG,NCOL,NROW,NLAY,NODES,VBVL,X(LCIOFL),
     2        IUNIT(12),IHEDFM,IDDNFM,IHEDUN,IDDNUN,IOUT)
      IF(IUNIT(1).GT.0) CALL BCF1RP(X(LCIBOU),X(LCHNEW),X(LCSC1),
     1           X(LCHY),X(LCCR),X(LCCC),X(LCCV),X(LCDELR),
     2            X(LCDELC),X(LCBOT),X(LCTOP),X(LCSC2),X(LCTRPY),
     3            IUNIT(1),ISS,NCOL,NROW,NLAY,NODES,IOUT,
     4            IUNIT(15),X(LCST),X(LCSV))
      IF(IUNIT(9).GT.0) CALL SIP1RP(NPARM,MXITER,ACCL,HCLOSE,X(LCW),
     1          IUNIT(9),IPCALC,IPRSIP,IOUT)
      IF(IUNIT(10).GT.0) LINO=LINEAR
      IF(IUNIT(11).GT.0) CALL SOR1RP(MXITER,ACCL,HCLOSE,IUNIT(11),
     1          IPRSOR,IOUT)
      IF(IUNIT(13).GT.0) CALL PCG2RP(MXITER,ITER1,HCLOSE,RCLOSE,NPCOND,
     1        NBPOL,RELAX,IPRPCG,IUNIT(13),IOUT,MUTPCG,NITER)
      NPO=NP
      IF(IUNIT(15).NE.0) CALL SEN1RP(NCOL,NROW,NLAY,NSN,NSM,MXBND,
     1          MXWELL,MXRIVR,NPER,IUNIT(15),IOUT,NP,PID,DID,X(LCNLL),
     2          X(LCCELS),X(LCSMAT),NH,X(LCNDER),JT,NRWD,X(LCIRWD),
     3          X(LCIBOU),ISN,X(LCJOFF),X(LCIOFF),X(LCHOBS),X(LCWT),
     4          X(LCB),X(LCB1),IPAR,X(LCCONV),HCLOSE,RCLOSE,X(LCTOP),
     5          X(LCBOT),X(LCDELR),X(LCDELC),X(LCRINT),X(LCCOFF),
     6          X(LCROFF),X(LCMLAY),X(LCPR),MOBS,X(LCLN),NQ,NQC,NQT,
     7          X(LCIBT),X(LCNQOB),X(LCNQCL),X(LCIQOB),X(LCQCLS),IUBD,
     8          X(LCBUFF),X(LCTOFF),MXDRN,MXSTRM,MAXM,NMM,NZM,LZI1,
     9          X(LCSFAC),X(LCLZ),X(LCLM),X(LCMATZ),NLLI1,IUNIT,NLLIT,
     X          EV,NUMR,ICHECK,NPNG,X(LCIPNG),NZER,X(LCIZER),
     1          X(LCIWPG),NPO)
      IF(IUNIT(15).NE.0) NPER=JT+1
      IF(IUNIT(15).NE.0.AND.ISS.NE.0.AND.NPER.GT.1)  THEN
        WRITE(IOUT,540)
540     FORMAT(' DEPENDENT-VARIABLE DATA HAS TIME STEPS >0, BUT '
     1          'ISS OF THE BCF PACKAGE INDICATES STEADY-STATE --'
     2          ' STOP EXECUTION')
        STOP
      ENDIF
      IF(IUNIT(15).NE.0.AND.IPAR.GE.1) CALL PAR1RP(IUNIT(15),IOUT,NP,
     1         PID,X(LCWP),X(LCB),X(LCCONV),X(LCLN),LASTX,DMAX,CSA,
     1         TOL,ITMXP,X(LCDD),IOUR,IPRC,IPRINT,FCONV,NPR,EV,NH,
     1         X(LCWT),NQT,KPRINT,MPR,X(LCPRM),X(LCIWPG),X(LCSCL),
     1         ISN,NFIT,LPRINT,NPO,SOSC,NPER,SOSR)
```

```
C
C-------INITIALIZE VALUES FOR PARAMETER ESTIMATION
      DMX=DMAX
      AP=1.
      IFO=0
      FAC=1.
      ADMX=10.*FCONV
      ISN1=ISN
      IF(ISN.EQ.10) ISN=-1
C-------BEGIN ITERATION LOOP FOR PARAMETER ESTIMATION
      ITERP=0
      ITERPF=ITMXP
  290 CONTINUE
      IF(ISN1.EQ.10.AND.MOD(ITERP-1,1+NFIT).EQ.0.AND.IFO.EQ.0) ISN=1
      IF(ISN1.EQ.10.AND.MOD(ITERP,1+NFIT).EQ.0.AND.IFO.EQ.0) ISN=-1
      ITERP=ITERP+1
      IF(IUNIT(15).GT.0.AND.ISN.GT.0.AND.ITERP.GT.1.AND.KNT.GT.0)
     1  ITERP=ITERP-1
      IF(IUNIT(15).GT.0.AND.IFO.GT.0) THEN
        ITERPF=ITERP-1
        ITERP=ITMXP
      ENDIF
      IP=-1
      NS=0
      NM=0
      IFLCH=0
C-------PREPARE FACTOR FOR CONVERGENCE CRITERIA FOR FCONV OPTION
      IF(IUNIT(15).GT.0.AND.IPAR.GT.0.AND.FCONV.GT.0.) THEN
        FAC=(AP*ADMX)/TOL
        IF(FAC.GT.1000.) FAC=1000.
        IF(FAC.LT.10.) FAC=1.
        WRITE(IOUT,550) FAC
      ENDIF
  550 FORMAT(//' CONVERGENCE CRITERIA INCREASED BY FACTOR OF',G14.5,//)
C-------BEGIN LOOP FOR CALCULATING HEADS, SENSITIVITY-EQUATION
C-------SENSITIVITIES, ADJOINT STATES, AND DERIVATIVES OF HEAD WITH
C-------RESPECT TO THE ITERATION PARAMETER
  295 CONTINUE
      IP=IP+1
      IS=0
C-------ASSIGN CONVERGENCE CRITERIA.
      IF(IUNIT(15).NE.0.AND.ISN.LT.0) THEN
        HCLOSE=X(LCCONV+IP)
        RCLOSE=X(LCCONV+NPO+1+IP)
        HCLOSE=FAC*HCLOSE
        RCLOSE=FAC*RCLOSE
      ENDIF
C
C7------SIMULATE EACH STRESS PERIOD.
      DO 300 KPER=1,NPER
      KKPER=KPER
C
C7A-----READ STRESS PERIOD TIMING INFORMATION.
      IF(IUNIT(15).EQ.0) CALL BAS1ST(NSTP,DELT,TSMULT,PERTIM,KKPER,
     #      INBAS,IOUT)
C
C----------PREPARE TIME STEPS FOR ADJOINT STATES AND SENSITIVITIES.
      IF(IUNIT(15).NE.0) THEN
        IF(KPER.EQ.1.AND.IP.EQ.0.AND.ITERP.EQ.1) CALL SEN1ST(
     #    JT,INBAS,PERLEN,NSTP,X(LCTS),X(LCATS),X(LCSTP),ISN,X(LCBSTP))
C
C----------ASSIGN TIME STEPS TO VARIABLES.
        IF(IP.EQ.1.AND.ISN.GT.0) THEN
          DELT=X((2*KPER-1)+LCATS-1)
          DELT1=X((2*KPER)+LCATS-1)
          IF(NPER.EQ.1) DELT1=0.
        ELSE
          IF(KPER.EQ.1) DELT=1.E22
          IF(KPER.GT.1) DELT=X(KPER-1+LCTS-1)
        ENDIF
        IF(KPER.EQ.2) THEN
          PERTIM=0.
          TOTIM=0.
        ENDIF
C
C----------IF THIS TIMESTEP IS THE FIRST IN A NEW PUMPING PERIOD,
C----------SET IPP=1 AND IS=IS+1.
        IPP=0
        IF(KPER.EQ.1) THEN
          IPP=1
        ELSE
          IF((ISN.LT.0.OR.IP.EQ.0.OR.IP.EQ.2).AND.
     1      KPER-1.EQ.INT(X(LCSTP+IS)))THEN
```

```
                IS=IS+1
                IPP=1
            ENDIF
            IF((ISN.GT.0.AND.IP.EQ.1).AND.
     1        NPER-KPER.EQ.INT(X(LCBSTP+1+IS))) THEN
                IS=IS+1
                IPP=1
                X(LCBSTP)=X(LCBSTP)-1
            ENDIF
          ENDIF
        ENDIF
C-----PREPARE HEADS, REWIND SOME FILES, REINITIALIZE HNEW
        IF(IUNIT(15).NE.0) THEN
C         REWIND SOME FILES
          IF((IP.GT.0.OR.ITERP.GT.1).AND.((ISN.EQ.-2.AND.KPER.EQ.1)
     1      .OR.(ISN.EQ.-1.AND.KPER.EQ.1.AND.IP.EQ.0)
     2      .OR.(ISN.GT.0.AND.(KPER.EQ.1.OR.IP.EQ.1))))
     3      CALL SEN1RW(IUNIT,NRWD,X(LCIRWD))
          KNST=1
          IF(IPAR.GE.0.AND.ISN.NE.-1)
     1       CALL SEN1FM(KNST,NCOL,NROW,NLAY,KPER,IUHEAD,NPER,X(LCSHNW),
     2       X(LCSHOL),ISN,IP,X(LCHNEW),NODES,PID,NP,KSTP,PERTIM,TOTIM,
     3       X(LCBUFF),X(LCHOLD),ITERP,ISN1,IOUT)
        ENDIF
C
C7B-----READ AND PREPARE INFORMATION FOR STRESS PERIOD.
C----------READ USING PACKAGE READ AND PREPARE MODULES.
        IF(IUNIT(15).EQ.0.OR.IPP.EQ.1) THEN
        IF(IUNIT(15).EQ.0.OR.(ITERP.EQ.1.AND.IP.EQ.0)) THEN
        IF(IUNIT(2).GT.0) CALL WEL1RP(X(LCWELL),NWELLS,MXWELL,IUNIT(2),
     1              IOUT)
        IF(IUNIT(3).GT.0) CALL DRN1RP(X(LCDRAI),NDRAIN,MXDRN,IUNIT(3),
     1                  IOUT)
        IF(IUNIT(8).GT.0) CALL RCH1RP(NRCHOP,X(LCIRCH),X(LCRECH),
     1             X(LCDELR),X(LCDELC),NROW,NCOL,IUNIT(8),IOUT)
        IF(IUNIT(5).GT.0) CALL EVT1RP(NEVTOP,X(LCIEVT),X(LCEVTR),
     1             X(LCEXDP),X(LCSURF),X(LCDELR),X(LCDELC),NCOL,NROW,
     1             IUNIT(5),IOUT)
        IF(IUNIT(4).GT.0) CALL RIV1RP(X(LCRIVR),NRIVER,MXRIVR,IUNIT(4),
     1             IOUT)
        IF(IUNIT(7).GT.0) CALL GHB1RP(X(LCBNDS),NBOUND,MXBND,IUNIT(7),
     1               IOUT)
        IF(IUNIT(18).GT.0) CALL STR1RP(X(LCSTRM),X(ICSTRM),NSTREM,
     1                       MXSTRM,IUNIT(18),IOUT,X(LCTBAR),NDIV,NSS,
     1                       NTRIB,X(LCIVAR),ICALC,IPTFLG)
        ELSE
C-----READ USING REPEATED READ MODULE FROM SEN PACKAGE -- NO PRINTING
          NRP=1
          IF(ISN.GT.0.AND.IP.EQ.1.AND.NPER.NE.1)
     1      NRP=X(LCBSTP)+1
C---------IUNIT POSITION FOR STR PACKAGE IS 18 IN SEN1RR
          IF(ISN.NE.-1.OR.IP.EQ.0)
     1       CALL SEN1RR (IUNIT,X(LCWELL),NWELLS,MXWELL,X(LCDRAI),NDRAIN,
     1            MXDRN,NRCHOP,X(LCIRCH),X(LCRECH),X(LCDELR),X(LCDELC),
     1            NROW,NCOL,NLAY,NEVTOP,X(LCIEVT),X(LCEVTR),X(LCEXDP),
     1            X(LCSURF),X(LCRIVR),NRIVER,MXRIVR,X(LCBNDS),NBOUND,
     1            MXBND,IOUT,IP,ITERP,X(LCSTRM),X(ICSTRM),NSTREM,
     1            MXSTRM,X(LCTBAR),NDIV,NSS,NTRIB,X(LCIVAR),ICALC,NRP)
        ENDIF
C-----PUT PARAMETERS IN MODEL ARRAYS.
        IF(IUNIT(15).NE.0.AND.(ISN.NE.-1.OR.IP.EQ.0))
     1     CALL SEN1FN(X(LCCC),X(LCCR),X(LCCV),X(LCCELS),X(LCSMAT),NSN,
     2          NSM,X(LCSC1),X(LCST),X(LCHNEW),X(LCBNDS),NBOUND,
     3          X(LCWELL),NWELLS,X(LCRIVR),NRIVER,X(LCRECH),X(LCDELR),
     4          X(LCDELC),NROW,NCOL,NLAY,NP,X(LCNLL),X(LCIBOU),IP,
     5          X(LCB),KPER,NPER,MXBND,MXRIVR,MXWELL,X(LCTRPY),PID,
     6          ISN,ITERP,X(LCB1),IOUT,X(LCLN),X(LCHY),X(LCEVTR),
     7          X(LCDRAI),NDRAIN,MXDRN,X(LCSTRM),MXSTRM,NSTREM,
     8          X(ICSTRM),IUBD,NMM,NZM,LZI1,X(LCSFAC),X(LCLZ),X(LCLM),
     9          X(LCMATZ),NLLI1,NLLIT,X(LCSV),X(LCWT),ND,IPRNP,X(LCBUFF),
     X          X(LCIWPG),NPO,NPNG,X(LCIPNG),IOUB,X(LCNDER),X(LCCOFF),
     1          X(LCROFF),X(LCRINT),X(LCJOFF),X(LCIOFF),X(LCMLAY),NH,
     2          MAXM,MOBS,IDRY,JDRY)
        ENDIF
C
C7C-----SIMULATE EACH TIME STEP.
        DO 200 KSTP=1,NSTP
        KKSTP=KSTP
C
C7C1----CALCULATE TIME STEP LENGTH. SET HOLD=HNEW.
       CALL BAS1AD(DELT,TSMULT,TOTIM,PERTIM,X(LCHNEW),X(LCHOLD),KKSTP,
     1             NCOL,NROW,NLAY)
C-------MODIFY HEADS AT CONSTANT-HEAD BOUNDARIES
```

```
      IF(IUNIT(15).GT.0.AND.IUNHEA.GT.0) CALL SEN1RH(IUNHEA,X(LCBUFF),
     1        X(LCHNEW),KPER,NROW,NCOL,NLAY,X(LCIBOU))
C
C    FOR CALCULATING ANYTHING BUT ADJOINT STATES:
       IF(IUNIT(15).EQ.0.OR.ISN.LT.0.OR.IP.EQ.0.OR.IP.EQ.2) DELT1=DELT
C
C7C2----ITERATIVELY FORMULATE AND SOLVE THE EQUATIONS.
   90 DO 100 KITER=1,MXITER
        KKITER=KITER
        IF(IUNIT(15).NE.0) THEN
          IF((ISN.EQ.-2.AND.IP.GT.0)
     1       .AND.(KPER.NE.1.OR.KITER.NE.1)) THEN
            NS=NS1
            NM=NM1
            IFLCH=IFLCH1
          ENDIF
          IF(ISN.EQ.-1) THEN
            IF(IP.EQ.1) THEN
              NS=0
              NM=0
              IFLCH=0
            ENDIF
            IF(KITER.GT.1) THEN
              NS=NS1
              NM=NM1
              IFLCH=IFLCH1
            ENDIF
            IF(IP.GT.0) GO TO 95
          ENDIF
        ENDIF
C
C7C2A---FORMULATE THE FINITE DIFFERENCE EQUATIONS.
      CALL BAS1FM(X(LCHCOF),X(LCRHS),NODES)
      IF(IUNIT(1).GT.0) CALL BCF1FM(X(LCHCOF),X(LCRHS),X(LCHOLD),
     1          X(LCSC1),X(LCHNEW),X(LCIBOU),X(LCCR),X(LCCC),X(LCCV),
     2          X(LCHY),X(LCTRPY),X(LCBOT),X(LCTOP),X(LCSC2),
     3          X(LCDELR),X(LCDELC),DELT,ISS,KKITER,KKSTP,KKPER,NCOL,
     4          NROW,NLAY,IOUT,
     5          DELT1,X(LCSHNW),IUNIT(15),IP,ISN,IPAR)
      IF(IUNIT(2).GT.0.AND.(IUNIT(15).EQ.0.OR.IP.EQ.0))
     1           CALL WEL1FM(NWELLS,MXWELL,X(LCRHS),X(LCWELL),
     1           X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(3).GT.0) CALL DRN1FM(NDRAIN,MXDRN,X(LCDRAI),X(LCHNEW),
     1          X(LCHCOF),X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY,
     2          X(LCSHNW),IUNIT(15),IP,ISN)
      IF(IUNIT(8).GT.0.AND.(IUNIT(15).EQ.0.OR.IP.EQ.0))
     1            CALL RCH1FM(NRCHOP,X(LCIRCH),X(LCRECH),
     1            X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY)
      IF(IUNIT(5).GT.0) CALL EVT1FM(NEVTOP,X(LCIEVT),X(LCEVTR),
     1            X(LCEXDP),X(LCSURF),X(LCRHS),X(LCHCOF),X(LCIBOU),
     1            X(LCHNEW),NCOL,NROW,NLAY,
     2            X(LCSHNW),IUNIT(15),IP,ISN)
      IF(IUNIT(4).GT.0) CALL RIV1FM(NRIVER,MXRIVR,X(LCRIVR),X(LCHNEW),
     1            X(LCHCOF),X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY,
     2            X(LCSHNW),IUNIT(15),IP,ISN)
      IF(IUNIT(18).GT.0) CALL STR1FM(NSTREM,X(LCSTRM),X(ICSTRM),
     1                                X(LCHNEW),X(LCHCOF),X(LCRHS),
     2                X(LCIBOU),MXSTRM,NCOL,NROW,NLAY,IOUT,NSS,
     3              X(LCTBAR),NTRIB,X(LCTRIB),X(LCIVAR),ICALC,CONST,
     4            X(LCSHNW),IUNIT(15),IP,ISN)
      IF(IUNIT(7).GT.0) CALL GHB1FM(NBOUND,MXBND,X(LCBNDS),X(LCHCOF),
     1            X(LCRHS),X(LCIBOU),NCOL,NROW,NLAY,
     2            IUNIT(15),IP,ISN)
C
C-------PREPARE RHS TO CALCULATE . . .
C--------- . . . ADJOINT STATES
      IF(IUNIT(15).NE.0.AND.ISN.GT.0.AND.IP.EQ.1)
     1  CALL SEN1AR(IOUT,NROW,NCOL,NLAY,NP,ISN,NH,X(LCNDER),X(LCHNEW),
     2       PID,DID,IP,KPER,BUFF,KSTP,PERTIM,TOTIM,X(LCX),X(LCDELR),
     3       X(LCDELC),X(LCIBOU),X(LCCOFF),X(LCROFF),X(LCH),X(LCWT),
     4       X(LCHOBS),IFO,ITERP,IPAR,X(LCRINT),X(LCJOFF),X(LCIOFF),
     5       X(LCMLAY),X(LCPR),MOBS,NPER,X(LCB),X(LCLN),NQ,NQC,NQT,
     6       NQOB,NQCL,X(LCIQOB),X(LCQCLS),X(LCIBT),MXBND,NBOUND,
     7       X(LCBNDS),MXRIVR,NRIVER,X(LCRIVR),X(LCSHNW),LASTX,ISCALS,
     8       X(LCTOFF),MXDRN,NDRAIN,X(LCDRAI),MXSTRM,NSTREM,X(LCSTRM),
     9       X(ICSTRM),X(LCRHS),MAXM)
C--------- . . . SENSITIVITY-EQUATION SENSITIVITIES OR ITERATION PARAM
   95 IF(IUNIT(15).NE.0.AND.((ISN.LT.0.AND.IP.GT.0).OR.
     1                      (ISN.GT.0.AND.IP.EQ.2))) THEN
        IF(ISN.LT.0) THEN
          NM1=NM
          NS1=NS
          IFLCH1=IFLCH
```

```
        ENDIF
        CALL SEN1AP(PID,KPER,NPER,DELT,IP,NP,NCOL,NROW,NLAY,
     1              NODES,X(LCHOLD),X(LCSHNW),X(LCHNEW),X(LCXX),
     2              X(LCSMAT),X(LCNLL),NM,NSM,X(LCDELR),X(LCDELC),
     3              X(LCST),X(LCIBOU),X(LCTRPY),X(LCCELS),NS,NSN,
     4              MXWELL,NWELLS,MXBND,NBOUND,MXRIVR,NRIVER,X(LCBNDS),
     5              X(LCWELL),X(LCRIVR),NRCHOP,X(LCIRCH),ISN,X(LCRHS),
     6              X(LCSHOL),X(LCSC1),X(LCCR),X(LCCC),X(LCCV),
     7              IFLCH,IOUT,X(LCHY),X(LCBOT),X(LCSC2),
     8              KITER,NDRAIN,MXDRN,X(LCDRAI),X(LCSURF),X(LCEXDP),
     9              NEVTOP,MXSTRM,NSTREM,X(LCSTRM),X(ICSTRM),X(LCDD),
     X              X(LCB),X(LCLN),X(LCBUFF),X(LCH),X(LCHOBS),X(LCRINT),
     1              X(LCIOFF),X(LCJOFF),X(LCNDER),NH,X(LCTOFF),X(LCSCL),
     2              NMM,NZM,LZI1,X(LCSFAC),X(LCLZ),X(LCLM),X(LCMATZ),
     3              NLLI1,NLLIT)
      ENDIF
C-------IF HNEW=HOLD=0 AND RHS=0, NO NEED TO SOLVE.
      CALL SEN1NZ(X(LCRHS),X(LCHNEW),NODES,ISA)
      IF(IUNIT(15).NE.0.AND.NZER.GT.0.AND.ISA.NE.0)
     1     CALL SEN1HZ(X(LCHNEW),KPER,NODES,NZER,X(LCIZER),X(LCRHS),
     2              ISA)
      IF(ISA.EQ.0) THEN
        ICNVG=1
        GO TO 110
      ENDIF
C
C7C2B---MAKE ONE CUT AT AN APPROXIMATE SOLUTION OR . . .
      IF(IUNIT(9).GT.0) CALL SIP1AP(X(LCHNEW),X(LCIBOU),X(LCCR),X(LCCC),
     1     X(LCCV),X(LCHCOF),X(LCRHS),X(LCEL),X(LCFL),X(LCGL),X(LCV),
     2     X(LCW),X(LCHDCG),X(LCLRCH),NPARM,KKITER,HCLOSE,ACCL,ICNVG,
     3     KKSTP,KKPER,IPCALC,IPRSIP,MXITER,NSTP,NCOL,NROW,NLAY,NODES,
     4     IOUT)
      IF(IUNIT(11).GT.0) CALL SOR1AP(X(LCHNEW),X(LCIBOU),X(LCCR),
     1     X(LCCC),X(LCCV),X(LCHCOF),X(LCRHS),X(LCA),X(LCRES),X(LCIEQP),
     2     X(LCHDCG),X(LCLRCH),KKITER,HCLOSE,ACCL,ICNVG,KKSTP,KKPER,
     3     IPRSOR,MXITER,NSTP,NCOL,NROW,NLAY,NSLICE,MBW,IOUT)
C . . . . . . . . . . EXECUTE MULTIPLE INNER ITERATIONS OR . . .
      KPER1=KPER
      KSTP1=KSTP
      MX1=MXITER
      ITER11=ITER1
      IF(IUNIT(15).NE.0) THEN
        KPER1=IP
        KSTP1=KPER-1
      ENDIF
      IF(IUNIT(13).GT.0) CALL PCG2AP(X(LCHNEW),X(LCIBOU),X(LCCR),
     1      X(LCCC),X(LCCV),X(LCHCOF),X(LCRHS),X(LCV),X(LCSS),X(LCP),
     2      X(LCCD),X(LCHCHG),X(LCLHCH),X(LCRCHG),X(LCLRCH),
     3      KKITER,NITER,HCLOSE,RCLOSE,ICNVG,KSTP1,KPER1,IPRPCG,
     4      MX1,ITER11,NPCOND,NBPOL,NSTP,NCOL,NROW,NLAY,NODES,
     5      RELAX,IOUT,MUTPCG,IUNIT(15),IP,SN,SP,SR)
C
      IF(ICNVG.EQ.1) GO TO 110
  100 CONTINUE
  105 CONTINUE
C
C7C2C---IF CONVERGENCE CRITERION HAS NOT BEEN MET STOP EXECUTION.
      IF(ICNVG.EQ.0) THEN
        IF(IUNIT(15).NE.0) THEN
           WRITE(IOUT,502) IP,KSTP1,KSTP1
  502      FORMAT(' *****WARNINGS ABOUT FOLLOWING TABLE : ',/,
     1         '     IP=',I2,/,'      ',
     2         'IF ISN=-1 OR IP=0, DEPENDENT VARIABLES AT TIME STEPS',
     3         ' AFTER',I5,' ARE CALCULATED WITH OLD PARAMETER VALUES.',
     4         /,'     IF IP=0, DEPENDENT VARIABLES',
     5         ' CALCULATED AT TIME STEP',I5,' ARE IN ERROR BECAUSE THE',
     6         ' SOLVER DID NOT CONVERGE (SEE BELOW)')
           CALL SSEN1O(NP,X(LCB),X(LCB1),X(LCIWPG),PID,X(LCWP),IOUT,NH,
     1         X(LCH),X(LCHOBS),X(LCWT),DID,X(LCNDER),X(LCROFF),
     2         X(LCCOFF),X(LCTOFF),NQT,X(LCIQOB),ND,IPAR,NPR,MPR,
     3         X(LCPRM),RSQ,RSQP,1,X(LCLN),RSQH,NPO)
           IF(IOUB.GT.0.AND.IPAR.GT.0) CALL SPAR1P(PID,X(LCBUFF),NP,
     1         IOUB,IOUT,ISN)
        ENDIF
        WRITE(IOUT,500) KPER1,KSTP1
  500   FORMAT(' ******** SOLUTION DID NOT CONVERGE FOR KPER=',I5,
     1         ', KSTP=',I5,' ********')
        IF(IUNIT(15).NE.0) WRITE(IOUT,505) IP
  505   FORMAT(' PARAMETER NUMBER (0 FOR HEADS) =',I5)
        STOP
      ENDIF
  110 CONTINUE
C
```

```
C7C3----DETERMINE WHICH OUTPUT IS NEEDED.
      IF(IUNIT(15).EQ.0 .OR. IPAR.LT.0 .OR.
     1     (IPAR.EQ.0.AND.(IP.EQ.0.OR.(ISN.NE.-1.AND.ISA.NE.0))) .OR.
     2     (IPAR.GT.0.AND.IP.EQ.0.AND.
     3          (IFO.GT.0.AND.(KPRINT.EQ.1.OR.KPRINT.EQ.3))))
     4CALL BAS1OC(NSTP,KKSTP,ICNVG,X(LCIOFL),NLAY,
     5  IBUDFL,ICBCFL,IHDDFL,IUNIT(12),IOUT)
C
C7C4----CALCULATE BUDGET TERMS. SAVE CELL-BY-CELL FLOW TERMS.
C---------SKIP WHEN CALCULATING ADJOINT STATES OR SENSITIVITIES.
      IF(IUNIT(15).EQ.0.OR.(IP.EQ.0.AND.(IPAR.LE.0.OR.
     1     (IFO.GT.0.AND.(KPRINT.EQ.1.OR.KPRINT.EQ.3))))) THEN
      MSUM=1
      IF(IUNIT(1).GT.0) CALL BCF1BD(VBNM,VBVL,MSUM,X(LCHNEW),
     1     X(LCIBOU),X(LCHOLD),X(LCSC1),X(LCCR),X(LCCC),X(LCCV),
     2     X(LCTOP),X(LCSC2),DELT,ISS,NCOL,NROW,NLAY,KKSTP,KKPER,
     3     IBCFCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(2).GT.0) CALL WEL1BD(NWELLS,MXWELL,VBNM,VBVL,MSUM,
     1     X(LCWELL),X(LCIBOU),DELT,NCOL,NROW,NLAY,KKSTP,KKPER,IWELCB,
     1     ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(3).GT.0) CALL DRN1BD(NDRAIN,MXDRN,VBNM,VBVL,MSUM,
     1     X(LCDRAI),DELT,X(LCHNEW),NCOL,NROW,NLAY,X(LCIBOU),KKSTP,
     2     KKPER,IDRNCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(8).GT.0) CALL RCH1BD(NRCHOP,X(LCIRCH),X(LCRECH),
     1     X(LCIBOU),NROW,NCOL,NLAY,DELT,VBVL,VBNM,MSUM,KKSTP,KKPER,
     2     IRCHCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(5).GT.0) CALL EVT1BD(NEVTOP,X(LCIEVT),X(LCEVTR),
     1     X(LCEXDP),X(LCSURF),X(LCIBOU),X(LCHNEW),NCOL,NROW,NLAY,
     2     DELT,VBVL,VBNM,MSUM,KKSTP,KKPER,IEVTCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(4).GT.0) CALL RIV1BD(NRIVER,MXRIVR,X(LCRIVR),X(LCIBOU),
     1     X(LCHNEW),NCOL,NROW,NLAY,DELT,VBVL,VBNM,MSUM,
     2     KKSTP,KKPER,IRIVCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(7).GT.0) CALL GHB1BD(NBOUND,MXBND,VBNM,VBVL,MSUM,
     1     X(LCBNDS),DELT,X(LCHNEW),NCOL,NROW,NLAY,X(LCIBOU),KKSTP,
     2     KKPER,IGHBCB,ICBCFL,X(LCBUFF),IOUT)
      IF(IUNIT(18).GT.0) CALL STR1BD(NSTREM,X(LCSTRM),X(ICSTRM),
     1         X(LCIBOU),MXSTRM,X(LCHNEW),NCOL,NROW,NLAY,DELT,VBVL,
     2       VBNM,MSUM,KKSTP,KKPER,ISTCB1,ISTCB2,ICBCFL,X(LCBUFF),IOUT,
     3      NTRIB,NSS,X(LCTRIB),X(LCTBAR),X(LCIVAR),ICALC,CONST,IPTFLG)
      ENDIF
C------SET FLAG WHICH INDICATES WHETHER THERE IS AN OBS. THIS TIME STEP
      IF(ISN.LT.0.AND.IPAR.EQ.0) CALL SEN1FD(IFLD,X(LCNDER),NH,KPER,
     1   X(LCTOFF),NQ,X(LCNQOB),X(LCIQOB),NQT)
C
C7C5---PRINT AND/OR SAVE HEAD, DRAWDOWN OR SENSITIVITY MATRICES.
C------PRINT OVERALL WATER BUDGET.
      IF(IUNIT(15).EQ.0.OR.IPAR.LT.0.OR.
     1    (IPAR.EQ.0.AND.(ISN.GT.0.OR.IFLD.EQ.1)).OR.
     1    (IPAR.GT.0.AND.IP.EQ.0.AND.(IFO.GT.0.AND.
     1             (KPRINT.EQ.1.OR.KPRINT.EQ.3)))) THEN
        KSTP1=KSTP
        KPER1=KPER
        IF(IUNIT(15).NE.0.AND.IHDDFL.NE.0) THEN
          IF(IP.NE.0) THEN
            IF(ISN.LT.0) THEN
              WRITE(IOUT,510) IP,PID(IP)
            ELSE
             IF(IP.EQ.1) WRITE(IOUT,515)
             IF(IP.EQ.2) WRITE(IOUT,520)
            ENDIF
          ENDIF
  510     FORMAT(/,' SENSITIVITY MATRIX FOR PARAMETER NO.',I5,
     1           ', ID ',A4,', SCALED BY *B',/)
  515     FORMAT(/,' ADJOINT STATES',/)
  520     FORMAT(/,' DERIVATIVE OF HEAD WITH RESPECT TO ITERATION',
     1         ' PARAMETER',/)
          KPER1=0
          KSTP1=KPER-1
          IF(ISN.GT.0.AND.IP.EQ.1) KSTP1=NPER-KPER
        ENDIF
        IF(ISA.EQ.0) THEN
          WRITE(IOUT,525) KSTP1
525       FORMAT(/,' FOR TIME STEP',I5,' ALL VALUES ARE 0.0',/)
        ELSE
          CALL BAS1OT(X(LCHNEW),X(LCSTRT),ISTRT,X(LCBUFF),X(LCIOFL),
     1        MSUM,X(LCIBOU),VBNM,VBVL,KSTP1,KPER1,DELT,
     2        PERTIM,TOTIM,ITMUNI,NCOL,NROW,NLAY,ICNVG,
     3        IHDDFL,IBUDFL,IHEDFM,IHEDUN,IDDNFM,IDDNUN,IOUT,
     4        IUNIT(15),IP,ISN,X(LCB+IP-1))
          IF(IUNIT(15).GT.0.AND.IP.GT.0.AND.
     1        IUNIT(13).GT.0.AND.MXITER.GT.1) THEN
            SE=100.*SR/((SP-SN)/2.)
            WRITE(IOUT,530) SP,SN,SR,SE
```

```
530       FORMAT(//,'    SUM OF ALL POSITIVE RATES.......... =',G20.5,
    1              /,'    SUM OF ALL NEGATIVE RATES.......... =',G20.5,
    2              /,'    SUM OF ALL RATES (SUM OF RESIDUALS) =',G20.5,
    3              /,'    PERCENT DISCREPANCY................ =',F20.2)
          ENDIF
        ENDIF
      ENDIF
C-----WRITE HEADS TO IUHEAD; INTERPOLATE, SAVE AND PRINT DATA AND
C-----SENSITIVITIES FOR OBSERVATIONS.
      IF(IUNIT(15).NE.0.AND.ICNVG.NE.0.AND.(ISN.LT.0.OR.IP.EQ.0))
    1  CALL SEN1OT(IUHEAD,IOUT,NROW,NCOL,NLAY,NP,ISN,NH,X(LCNDER),
    2     X(LCHNEW),PID,DID,IP,KPER,X(LCBUFF),KSTP,PERTIM,TOTIM,
    3     X(LCX),X(LCDELR),X(LCDELC),X(LCIBOU),X(LCCOFF),X(LCROFF),
    4     X(LCH),X(LCWT),X(LCHOBS),IPRINT,IFO,ITERP,IPAR,X(LCRINT),
    5     X(LCJOFF),X(LCIOFF),X(LCMLAY),X(LCPR),MOBS,NPER,X(LCB),
    6     X(LCLN),NQ,NQC,NQT,X(LCNQOB),X(LCNQCL),X(LCIQOB),X(LCQCLS),
    7     X(LCIBT),MXBND,NBOUND,X(LCBNDS),MXRIVR,NRIVER,X(LCRIVR),
    8     X(LCSHNW),LASTX,ISCALS,X(LCTOFF),MXDRN,NDRAIN,X(LCDRAI),
    9     MXSTRM,NSTREM,X(LCSTRM),X(ICSTRM),MAXM,KPRINT,JDRY,IDRY,NPR,
    X     X(LCWP),MPR,X(LCPRM),X(LCIWPG),X(LCB1),IOUB,RSQ,RSQP,RSQO,
    1     RSQOO,NPO,SOSC,SOSR)
C
C7C6----IF ITERATION FAILED TO CONVERGE THEN STOP.
      IF(ICNVG.EQ.0) STOP
C-------IF CONVERGENCE ACHIEVED BY SUM OF SQUARES CRITERIA
        IF(IFO.EQ.2.AND.ISN.LT.0.AND.IP.EQ.NP) THEN
          ITERPF=ITERP
          ITERP=ITMXP
          GO TO 200
        ENDIF
C-------FOR ISN=-1 PUT CORRECT VALUES IN HNEW, SET CONVERGENCE CRITERIA,
C-------AND LOOP BACK TO APPROPRIATE PLACE
      IF(IUNIT(15).NE.0.AND.ISN.EQ.-1) THEN
        IF(IPAR.LT.0) GO TO 200
        IF(IFO.EQ.0.OR.LASTX.NE.0)
    1      CALL SEN1FM(KNST,NCOL,NROW,NLAY,KPER,IUHEAD,NPER,X(LCSHNW),
    2      X(LCSHOL),ISN,IP,X(LCHNEW),NODES,PID,NP,KSTP,PERTIM,TOTIM,
    3      X(LCBUFF),X(LCHOLD),ITERP,ISN1,IOUT)
        IF(IP.EQ.NP) THEN
          IF(KPER.NE.NPER) IP=0
          HCLOSE=X(LCCONV)
          RCLOSE=X(LCCONV+NPO+1)
          HCLOSE=FAC*HCLOSE
          RCLOSE=FAC*RCLOSE
          GO TO 200
        ENDIF
        IF(IFO.EQ.1.AND.LASTX.EQ.0) THEN
          IF(KPER.EQ.1) IP=0
          GO TO 200
        ENDIF
        IP=IP+1
        HCLOSE=X(LCCONV+IP)
        RCLOSE=X(LCCONV+NPO+1+IP)
        HCLOSE=FAC*HCLOSE
        RCLOSE=FAC*RCLOSE
        GO TO 90
      ENDIF
C
C-----ADJOINT STATE AND CONJUGATE-DIRECTION CALCULATIONS
      IF(IUNIT(15).NE.0.AND.ISN.GT.0) THEN
C--------CALCULATE CONTRIBUTION TO THE GRADIENT FROM THE ADJOINT STATE
C--------FROM ONE TIME STEP
        IF(IP.EQ.1) THEN
C-----------PRINT TIME STEP LENGTHS USED TO CALCULATE ADJOINT STATES.
          IF(IPAR.EQ.0) WRITE(IOUT,25) DELT,DELT1
25        FORMAT(//,' TIME STEP LENGTHS :',F15.5,', ',F15.5)
C-----------PUT APPROPRIATE HEAD VECTORS IN SHNW AND HOLD
          KNST=-1
          CALL SEN1FM(KNST,NCOL,NROW,NLAY,KPER,IUHEAD,NPER,X(LCSHNW),
    2      X(LCSHOL),ISN,IP,X(LCHNEW),NODES,PID,NP,KSTP,PERTIM,TOTIM,
    3      X(LCBUFF),X(LCHOLD),ITERP,ISN1,IOUT)
C-----------CALCULATE THE GRADIENT
          CALL SEN1AP(PID,KPER,NPER,DELT,IP,NP,NCOL,NROW,NLAY,
    1                NODES,X(LCHOLD),X(LCSHNW),X(LCHNEW),X(LCXX),
    2                X(LCSMAT),X(LCNLL),NM,NSM,X(LCDELR),X(LCDELC),
    3                X(LCST),X(LCIBOU),X(LCTRPY),X(LCCELS),NS,NSN,
    4                MXWELL,NWELLS,MXBND,NBOUND,MXRIVR,NRIVER,X(LCBNDS),
    5                X(LCWELL),X(LCRIVR),NRCHOP,X(LCIRCH),ISN,X(LCRHS),
    6                X(LCSHOL),X(LCSC1),X(LCCR),X(LCCC),X(LCCV),
    7                IFLCH,IOUT,X(LCHY),X(LCBOT),X(LCSC2),
    8                KITER,NDRAIN,MXDRN,X(LCDRAI),X(LCSURF),X(LCEXDP),
    9                NEVTOP,MXSTRM,NSTREM,X(LCSTRM),X(ICSTRM),X(LCDD),
    X                X(LCB),X(LCLN),X(LCBUFF),X(LCH),X(LCHOBS),X(LCRINT),
```

```
     1              X(LCIOFF),X(LCJOFF),X(LCNDER),NH,X(LCTOFF),X(LCSCL),
     2              NMM,NZM,LZI1,X(LCSFAC),X(LCLZ),X(LCLM),X(LCMATZ),
     3              NLLI1,NLLIT)
C-----------SAVE HEAD VECTOR THAT WAS STORED IN HOLD
        KNST=0
        CALL SEN1FM(KNST,NCOL,NROW,NLAY,KPER,IUHEAD,NPER,X(LCSHNW),
     2   X(LCSHOL),ISN,IP,X(LCHNEW),NODES,PID,NP,KSTP,PERTIM,TOTIM,
     3   X(LCBUFF),X(LCHOLD),ITERP,ISN1,IOUT)
      ENDIF
C--------NONLINEAR REGRESSION BY GRADIENT SEARCH
C-----------CALCULATE SEARCH DIRECTION
      IF(IPAR.GT.0.AND.KPER.EQ.NPER.AND.IP.NE.2)
     X  CALL PAR1AS(RSQP,KNT,AP,X(LCB),
     1        X(LCDD),ITERP,X(LCG),NP,IP,X(LCH),X(LCHOBS),X(LCSCL),
     2        X(LCWP),X(LCWT),ND,MPR,IOUT,SSX,NFIT,ITG,PID,
     3        X(LCBUFF),X(LCB1),X(LCC),X(LCXX),X(LCLN),IFO,LASTX,ADMX,
     4        X(LCPRM),IPAR,TOL,ITMXP,NPR,RSQO,AGMX,DID,X(LCNDER),
     5        X(LCROFF),X(LCCOFF),X(LCTOFF),NH,NQT,X(LCIQOB),IDRY,
     6        X(LCIWPG),IOUB,EV,ISN,ISN1,X(LCSCLE),X(LCC),NPO)
C-----------UPDATE PARAMETERS
      IF(IP.EQ.2.AND.IFO.EQ.0) CALL PAR1AQ(ND,X(LCHOBS),X(LCH),
     1        X(LCB1),X(LCB),X(LCDD),NP,X(LCWT),X(LCWP),PID,X(LCHNEW),
     2        NCOL,NROW,NLAY,ISN,IP,X(LCSCL),X(LCPV),NH,X(LCNDER),
     3        X(LCIOFF),X(LCJOFF),KPER,X(LCMLAY),X(LCIBOU),X(LCRINT),
     4        DID,X(LCCOFF),X(LCROFF),X(LCDELR),X(LCDELC),X(LCPR),
     5        MOBS,X(LCLN),NQT,X(LCTOFF),NQ,NQC,NQOB,NQCL,
     6        X(LCIQOB),X(LCQCLS),X(LCIBT),MXBND,NBOUND,X(LCBNDS),
     7        MXRIVR,NRIVER,X(LCRIVR),X(LCSHNW),NPER,MXDRN,NDRAIN,
     8        X(LCDRAI),MXSTRM,NSTREM,X(LCSTRM),X(ICSTRM),
     9        ADMX,X(LCPRM),MPR,IOUT,DMX,DMAX,AP,NPR,ITERP,AGMX,MAXM,
     X        JDRY,IPAR,IOUB,NPO)
      ENDIF
C-----END OF TIME STEP (KSTP) AND STRESS PERIOD (KPER) LOOPS
  200 CONTINUE
  300 CONTINUE
C-----INCREMENT IP AND CONTINUE?
      IF(IUNIT(15).NE.0.AND.
     1 (ISN.EQ.-2.AND.((IPAR.EQ.0.AND.IP.NE.NP).OR.
     1                 (IPAR.GT.0.AND.IP.NE.NP.AND.(IFO.EQ.0.OR.
     1                    LASTX.NE.0))))
     1 .OR.
     1 (ISN.GT.0.AND.((IP.EQ.0.AND.IPAR.GE.0.AND.KNT.EQ.0).OR.
     1              (IP.EQ.1.AND.IPAR.GT.0.AND.IFO.EQ.0)))) GO TO 295
C-----ANOTHER GRADIENT-SEARCH REGRESSION ITERATION?
      IF(IPAR.GT.0.AND.ISN.GT.0.AND.ITERP.LT.ITMXP.AND.IFO.EQ.0)
     1    GO TO 290
C-----CALCULATE FINAL GAUSS-NEWTON STATISTICS FOR GRADIENT-SEARCH
C-----REGRESSION?
      IF(IPAR.GT.0.AND.ISN.GT.0.AND.IFO.NE.0.AND.LASTX.NE.0) THEN
        ISN=-LASTX
        GO TO 290
      ENDIF
C
C-----NONLINEAR REGRESSION BY MODIFIED GAUSS-NEWTON
      IF(IUNIT(15).NE.0.AND.IPAR.GT.0.AND.ISN.LT.0)THEN
C--------EXECUTE ONE GAUSS-NEWTON ITERATION
        CALL PAR1AP(X(LCX),X(LCB),ND,NP,X(LCHOBS),X(LCWT),X(LCWP),
     1    X(LCC),X(LCSCLE),X(LCG),X(LCH),X(LCDD),DMAX,CSA,TOL,PID,
     1    IND,IFO,AMP,AP,DMX,IOUT,X(LCB1),ITERP,IPRINT,ADMX,X(LCLN),
     1    LASTX,MPR,X(LCPRM),NPR,IOUB,RSQ,JMAX,ITMXP,ISN1,NFIT,
     1    NPO,X(LCR),X(LCGD),X(LCU),NOPT,X(LCXD),NPER,X(LCS),SOSR)
C--------DO CALCULATIONS REQUIRED TO DO ANOTHER GAUSS-NEWTON ITERATION?
        IF(ITERP.LT.ITMXP.AND.IND.EQ.0) GO TO 290
C--------FINAL OUTPUT
        IF(IND.GT.0.OR.IFO.EQ.0) CALL SSEN1O(NP,X(LCB),X(LCB1),
     1     X(LCIWPG),PID,X(LCWP),IOUT,NH,X(LCH),X(LCHOBS),X(LCWT),DID,
     2     X(LCNDER),X(LCROFF),X(LCCOFF),X(LCTOFF),NQT,X(LCIQOB),ND,
     3     IPAR,NPR,MPR,X(LCPRM),RSQ,RSQP,1,X(LCLN),RSQH,NPO)
        IF(IOUB.GT.0) CALL SPAR1P(PID,X(LCBUFF),NP,IOUB,IOUT,ISN)
        CALL PAR1OT(X(LCC),X(LCWT),NP,IOUR,RSQ,IOUT,X(LCBUFF),ND,IPRC,
     1     IFO,IND,PID,X(LCSCLE),X(LCB),X(LCHOBS),X(LCH),NPR,X(LCB1),
     2     X(LCWP),ITERPF,X(LCX),X(LCHNEW),X(LCLN),MPR,X(LCPRM),
     3     X(LCIWPG),NODES,LPRINT,IDRY,EV,RSQP,ISN,NPO)
      ENDIF
C-------RESIDUAL ANALYSIS
      IF(IUNIT(15).GT.0) CALL PAR1RE(NP,X(LCB),X(LCB1),X(LCIWPG),
     1          X(LCWP),IOUT,NH,X(LCH),X(LCHOBS),X(LCWT),DID,NQT,ND,
     2          IPAR,NPR,MPR,X(LCPRM),NPO)
C
C8------END PROGRAM
      STOP
C
      END
```

```
      SUBROUTINE BAS1OT(HNEW,STRT,ISTRT,BUFF,IOFLG,MSUM,IBOUND,VBNM,
     1  VBVL,KSTP,KPER,DELT,PERTIM,TOTIM,ITMUNI,NCOL,NROW,NLAY,ICNVG,
     2  IHDDFL,IBUDFL,IHEDFM,IHEDUN,IDDNFM,IDDNUN,IOUT,
     3  IU,IP,ISN,B)
C-----VERSION 1522 12MAY1987 BAS1OT
C-----VERSION MODIFIED FOR MODFLOWP 1000 01FEB1992
C
C     ******************************************************************
C     OUTPUT TIME, VOLUMETRIC BUDGET, HEAD, AND DRAWDOWN
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 VBNM
      DOUBLE PRECISION HNEW,DB
C
      DIMENSION HNEW(NCOL,NROW,NLAY),STRT(NCOL,NROW,NLAY),
     1          VBNM(4,20),VBVL(4,20),IOFLG(NLAY,4),
     2          IBOUND(NCOL,NROW,NLAY),BUFF(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------CLEAR PRINTOUT FLAG (IPFLG)
      IPFLG=0
C
C2------IF ITERATIVE PROCEDURE FAILED TO CONVERGE PRINT MESSAGE
      IF(ICNVG.EQ.0) WRITE(IOUT,1) KSTP,KPER
    1 FORMAT(1H0,10X,'****FAILED TO CONVERGE IN TIME STEP',I3,
     1      ' OF STRESS PERIOD',I3,'****')
C
C3------IF HEAD AND DRAWDOWN FLAG (IHDDFL) IS SET WRITE HEAD AND
C3------DRAWDOWN IN ACCORDANCE WITH FLAGS IN IOFLG.
      IF(IHDDFL.EQ.0) GO TO 100
C
      IDUM=KPER
      IF(IU.GT.0.AND.IP.GT.0) THEN
        IDUM=IP
        IF(ISN.LT.0) THEN
          DB=DBLE(B)
          DO 50 K=1,NLAY
          DO 50 I=1,NROW
          DO 50 J=1,NCOL
   50     HNEW(J,I,K)=HNEW(J,I,K)*DB
        ENDIF
      ENDIF
      CALL SBAS1H(HNEW,BUFF,IOFLG,KSTP,IDUM,NCOL,NROW,
     1     NLAY,IOUT,IHEDFM,IHEDUN,IPFLG,PERTIM,TOTIM)
      CALL SBAS1D(HNEW,BUFF,IOFLG,KSTP,KPER,NCOL,NROW,NLAY,IOUT,
     1 IDDNFM,IDDNUN,STRT,ISTRT,IBOUND,IPFLG,PERTIM,TOTIM)
      IF(IU.GT.0.AND.IP.GT.0.AND.ISN.LT.0) THEN
        DO 70 K=1,NLAY
        DO 70 I=1,NROW
        DO 70 J=1,NCOL
   70   HNEW(J,I,K)=HNEW(J,I,K)/DB
      ENDIF
C
C4------PRINT TOTAL BUDGET IF REQUESTED
C-------MODIFIED FOR CALCULATING SENSITIVITIES
  100 IF(IBUDFL.EQ.0.OR.IP.NE.0) GO TO 120
      CALL SBAS1V(MSUM,VBNM,VBVL,KSTP,KPER,IOUT)
      IPFLG=1
C
C5------END PRINTOUT WITH TIME SUMMARY AND FORM FEED IF ANY PRINTOUT
C5------WILL BE PRODUCED.
  120 IF(IPFLG.EQ.0) RETURN
      CALL SBAS1T(KSTP,KPER,DELT,PERTIM,TOTIM,ITMUNI,IOUT)
      WRITE(IOUT,101)
  101 FORMAT(1H1)
C
C6------RETURN
      RETURN
      END
```

```
      SUBROUTINE BCF1RP(IBOUND,HNEW,SC1,HY,CR,CC,CV,DELR,DELC,          SUB0072
     1        BOT,TOP,SC2,TRPY,IN,ISS,NCOL,NROW,NLAY,NODES,IOUT,        SUB0073
     2        IU,ST,SV)                                                 SUB0074
C                                                                       SUB0075
C-----VERSION 1636 15MAY1987 BCF1RP                                     SUB0076
C-----VERSION MODIFIED FOR MODFLOWP 1000 01FEB1992                      SUB0077
C                                                                       SUB0078
C                                                                       SUB0079
C     ******************************************************************SUB0080
C     READ AND INITIALIZE DATA FOR BLOCK-CENTERED FLOW PACKAGE          SUB0081
C     ******************************************************************SUB0082
C                                                                       SUB0083
C        SPECIFICATIONS:                                                SUB0084
C     ------------------------------------------------------------------SUB0085
      CHARACTER*4 ANAME                                                 SUB0086
      DOUBLE PRECISION HNEW                                             SUB0087
C                                                                       SUB0088
      DIMENSION HNEW(NODES),SC1(NODES),HY(NODES),CR(NODES),CC(NODES),   SUB0089
     1          CV(NODES),ANAME(6,10),DELR(NCOL),DELC(NROW),BOT(NODES), SUB0090
     1          TOP(NODES),SC2(NODES),TRPY(NLAY),IBOUND(NODES)          SUB0091
     2          ,ST(NCOL,NROW,NLAY),SV(NCOL*NROW)                       SUB0092
C                                                                       SUB0093
      COMMON /FLWCOM/LAYCON(80)                                         SUB0094
C                                                                       SUB0095
      DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),      SUB0096
     1 ANAME(6,1) /'    ','PRIM','ARY ','STOR','AGE ','COEF'/           SUB0097
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),      SUB0098
     1 ANAME(6,2) /'    ','TRAN','SMIS','. AL','ONG ','ROWS'/           SUB0099
      DATA ANAME(1,3),ANAME(2,3),ANAME(3,3),ANAME(4,3),ANAME(5,3),      SUB0100
     1 ANAME(6,3) /'   H','YD. ','COND','. AL','ONG ','ROWS'/           SUB0101
      DATA ANAME(1,4),ANAME(2,4),ANAME(3,4),ANAME(4,4),ANAME(5,4),      SUB0102
     1 ANAME(6,4) /'VERT',' HYD',' CON','D /T','HICK','NESS'/           SUB0103
      DATA ANAME(1,5),ANAME(2,5),ANAME(3,5),ANAME(4,5),ANAME(5,5),      SUB0104
     1 ANAME(6,5) /'    ','    ','    ','    ','  BO','TTOM'/           SUB0105
      DATA ANAME(1,6),ANAME(2,6),ANAME(3,6),ANAME(4,6),ANAME(5,6),      SUB0106
     1 ANAME(6,6) /'    ','    ','    ','    ','    ',' TOP'/           SUB0107
      DATA ANAME(1,7),ANAME(2,7),ANAME(3,7),ANAME(4,7),ANAME(5,7),      SUB0108
     1 ANAME(6,7) /'  SE','COND','ARY ','STOR','AGE ','COEF'/           SUB0109
      DATA ANAME(1,8),ANAME(2,8),ANAME(3,8),ANAME(4,8),ANAME(5,8),      SUB0110
     1 ANAME(6,8) /'COLU','MN T','O RO','W AN','ISOT','ROPY'/           SUB0111
      DATA ANAME(1,9),ANAME(2,9),ANAME(3,9),ANAME(4,9),ANAME(5,9),      SUB0112
     1 ANAME(6,9) /'    ','    ','    ','    ','    ','DELR'/           SUB0113
      DATA ANAME(1,10),ANAME(2,10),ANAME(3,10),ANAME(4,10),ANAME(5,10), SUB0114
     1 ANAME(6,10) /'    ','    ','    ','    ','    ','DELC'/          SUB0115
C     ------------------------------------------------------------------SUB0116
C                                                                       SUB0117
C1------CALCULATE NUMBER OF NODES IN A LAYER AND READ TRPY,DELR,DELC    SUB0118
      NIJ=NCOL*NROW                                                     SUB0119
C                                                                       SUB0120
      CALL U1DREL(TRPY,ANAME(1,8),NLAY,IN,IOUT)                         SUB0121
      CALL U1DREL(DELR,ANAME(1,9),NCOL,IN,IOUT)                         SUB0122
      CALL U1DREL(DELC,ANAME(1,10),NROW,IN,IOUT)                        SUB0123
C                                                                       SUB0124
C2------READ ALL PARAMETERS FOR EACH LAYER                              SUB0125
      KT=0                                                              SUB0126
      KB=0                                                              SUB0127
      DO 200 K=1,NLAY                                                   SUB0128
      KK=K                                                              SUB0129
C                                                                       SUB0130
C2A-----FIND ADDRESS OF EACH LAYER IN THREE DIMENSION ARRAYS.           SUB0131
      IF(LAYCON(K).EQ.1 .OR. LAYCON(K).EQ.3) KB=KB+1                    SUB0132
      IF(LAYCON(K).EQ.2 .OR. LAYCON(K).EQ.3) KT=KT+1                    SUB0133
      LOC=1+(K-1)*NIJ                                                   SUB0134
      LOCB=1+(KB-1)*NIJ                                                 SUB0135
      LOCT=1+(KT-1)*NIJ                                                 SUB0136
C                                                                       SUB0137
C2B-----READ PRIMARY STORAGE COEFFICIENT INTO ARRAY SC1 IF TRANSIENT    SUB0138
      IF(ISS.EQ.0)CALL U2DREL(SC1(LOC),ANAME(1,1),NROW,NCOL,KK,IN,IOUT) SUB0139
C                                                                       SUB0140
C2C-----READ TRANSMISSIVITY INTO ARRAY CC IF LAYER TYPE IS 0 OR 2       SUB0141
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.1) GO TO 100                  SUB0142
      CALL U2DREL(CC(LOC),ANAME(1,2),NROW,NCOL,KK,IN,IOUT)              SUB0143
      IF(IU.GT.0) THEN                                                  SUB0144
        DO 300 I=1,NROW                                                 SUB0145
        DO 300 J=1,NCOL                                                 SUB0146
  300   ST(J,I,K)=CC(LOC-1+(J+(I-1)*NCOL))                              SUB0147
      ENDIF                                                             SUB0148
      GO TO 110                                                         SUB0149
C                                                                       SUB0150
C2D-----READ HYDRAULIC CONDUCTIVITY(HY) AND BOTTOM ELEVATION(BOT)       SUB0151
C2D-----IF LAYER TYPE IS 1 OR 3                                         SUB0152
  100 CALL U2DREL(HY(LOCB),ANAME(1,3),NROW,NCOL,KK,IN,IOUT)             SUB0153
      CALL U2DREL(BOT(LOCB),ANAME(1,5),NROW,NCOL,KK,IN,IOUT)            SUB0154
```

```
C-------ADDED FOR PARAMETER ESTIMATION
      IF(IU.GT.0) THEN
        DO 310 I=1,NROW
        DO 310 J=1,NCOL
  310   ST(J,I,K)=HY(LOCB-1+(J+(I-1)*NCOL))
      ENDIF
C
C2E-----READ VERTICAL HYCOND/THICK INTO ARRAY CV IF NOT BOTTOM LAYER
C2E----- READ AS HYCOND/THICKNESS -- CONVERTED TO CONDUCTANCE LATER
  110 IF(K.EQ.NLAY) GO TO 120
      CALL U2DREL(CV(LOC),ANAME(1,4),NROW,NCOL,KK,IN,IOUT)
C-------ADDED FOR PARAMETER ESTIMATION
      IF(IU.GT.0.AND.LAYCON(K).EQ.1) THEN
        DO 115 II=1,NIJ
  115   SV(II)=CV(II)
      ENDIF
C
C2F-----READ SECONDARY STORAGE COEFFICIENT INTO ARRAY SC2 IF TRANSIENT
C2F-----AND LAYER TYPE IS 2 OR 3
  120 IF(LAYCON(K).NE.3 .AND. LAYCON(K).NE.2) GO TO 200
      IF(ISS.EQ.0)CALL U2DREL(SC2(LOCT),ANAME(1,7),NROW,NCOL,KK,IN,IOUT)
C
C2G-----READ TOP ELEVATION(TOP) IF LAYER TYPE IS 2 OR 3
      CALL U2DREL(TOP(LOCT),ANAME(1,6),NROW,NCOL,KK,IN,IOUT)
  200 CONTINUE
C
C3------PREPARE AND CHECK BCF DATA
      CALL SBCF1N(HNEW,IBOUND,SC1,SC2,CR,CC,CV,HY,TRPY,DELR,DELC,ISS,
     1          NCOL,NROW,NLAY,IOUT)
C
C4------RETURN
      RETURN
      END


      SUBROUTINE BCF1FM(HCOF,RHS,HOLD,SC1,HNEW,IBOUND,CR,CC,CV,HY,TRPY,
     1                 BOT,TOP,SC2,DELR,DELC,DELT,ISS,KITER,KSTP,KPER,
     2                 NCOL,NROW,NLAY,IOUT,
     3                 DELT1,SHNW,IU,IP,ISN,IPAR)
C-----VERSION 1640 15MAY1987 BCF1FM
C-----VERSION MODIFIED FOR MODFLOWP 1000 01FEB1992
C
C
C     ******************************************************************
C     ADD LEAKAGE CORRECTION AND STORAGE TO HCOF AND RHS, AND CALCULATE
C     CONDUCTANCE AS REQUIRED
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION HCOF(NCOL,NROW,NLAY),RHS(NCOL,NROW,NLAY),
     1    HOLD(NCOL,NROW,NLAY),SC1(NCOL,NROW,NLAY),HNEW(NCOL,NROW,NLAY),
     2    IBOUND(NCOL,NROW,NLAY),CR(NCOL,NROW,NLAY),
     3    CC(NCOL,NROW,NLAY),CV(NCOL,NROW,NLAY),HY(NCOL,NROW,NLAY),
     4    TRPY(NLAY),BOT(NCOL,NROW,NLAY),TOP(NCOL,NROW,NLAY),DELR(NCOL),
     5    DELC(NROW),SC2(NCOL,NROW,NLAY)
     6    ,SHNW(NCOL,NROW,NLAY)
C
      COMMON /FLWCOM/LAYCON(80)
C     ------------------------------------------------------------------
      KB=0
      KT=0
C
C1------FOR EACH LAYER: IF T VARIES CALCULATE HORIZONTAL CONDUCTANCES
      DO 100 K=1,NLAY
      KK=K
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) KT=KT+1
C
C1A-----IF LAYER TYPE IS NOT 1 OR 3 THEN SKIP THIS LAYER.
      IF(LAYCON(K).NE.3 .AND. LAYCON(K).NE.1) GO TO 100
      KB=KB+1
C
C1B-----FOR LAYER TYPES 1 & 3 CALL SBCFH1 TO CALCULATE
C1B-----HORIZONTAL CONDUCTANCES.
C-------MODIFIED FOR PARAMETER ESTIMATION
      CALL SBCF1H(HNEW,IBOUND,CR,CC,CV,HY,TRPY,DELR,DELC,BOT,TOP,
     1          KK,KB,KT,KITER,KSTP,KPER,NCOL,NROW,NLAY,IOUT,
     2          SHNW,IU,IP,ISN,IPAR)
  100 CONTINUE
C
C2------IF THE SIMULATION IS TRANSIENT ADD STORAGE TO HCOF AND RHS
```

```
      IF(ISS.NE.0) GO TO 201
      TLED=1./DELT
C-------TIME STEP FOR CALCULATION OF ADJOINT STATES
      TLED1=1./DELT1
      KT=0
      DO 200 K=1,NLAY
C
C3------SEE IF THIS LAYER IS CONVERTIBLE OR NON-CONVERTIBLE.
      IF(LAYCON(K).EQ.3 .OR. LAYCON(K).EQ.2) GO TO 150
C4------NON-CONVERTIBLE LAYER, SO USE PRIMARY STORAGE
      DO 140 I=1,NROW
      DO 140 J=1,NCOL
      IF(IBOUND(J,I,K).LE.0) GO TO 140
      RHO=SC1(J,I,K)*TLED
C-------FOR ADJOINT STATES
      RHOI=SC1(J,I,K)*TLED1
      HCOF(J,I,K)=HCOF(J,I,K)-RHO
C-------LINE MODIFIED TO CALCULATE ADJOINT STATES (RHO TO RHOI)
      RHS(J,I,K)=RHS(J,I,K)-RHOI*HOLD(J,I,K)
  140 CONTINUE
      GO TO 200
C
C5------A CONVERTIBLE LAYER, SO CHECK OLD AND NEW HEADS TO DETERMINE
C5------WHEN TO USE PRIMARY AND SECONDARY STORAGE
  150 KT=KT+1
      DO 180 I=1,NROW
      DO 180 J=1,NCOL
C
C5A-----IF THE CELL IS EXTERNAL THEN SKIP IT.
      IF(IBOUND(J,I,K).LE.0) GO TO 180
      TP=TOP(J,I,KT)
      RHO2=SC2(J,I,KT)*TLED
      RHO1=SC1(J,I,K)*TLED
C-------TO CALCULATE ADJOINT STATES
      RHO21=SC2(J,I,KT)*TLED1
      RHO11=SC1(J,I,K)*TLED1
C
C5B-----FIND STORAGE FACTOR AT START OF TIME STEP.
C-------MODIFIED FOR ADJOINT STATES (RHO2 TO RHO21, AND RHO1 TO RHO11)
      SOLD=RHO21
      IF(HOLD(J,I,K).GT.TP) SOLD=RHO11
C
C5C-----FIND STORAGE FACTOR AT END OF TIME STEP.
      HTMP=HNEW(J,I,K)
      SNEW=RHO2
      IF(HTMP.GT.TP) SNEW=RHO1
C-------ADDED FOR ADJOINT STATES
      SNEW1=RHO21
      IF(HTMP.GT.TP) SNEW1=RHO11
C
C5D-----ADD STORAGE TERMS TO RHS AND HCOF.
      HCOF(J,I,K)=HCOF(J,I,K)-SNEW
C-------MODIFIED FOR ADJOINT STATES (SNEW TO SNEW1)
      RHS(J,I,K)=RHS(J,I,K) - SOLD*(HOLD(J,I,K)-TP) - SNEW1*TP
C
  180 CONTINUE
C
  200 CONTINUE
C
C6------FOR EACH LAYER DETERMINE IF CORRECTION TERMS ARE NEEDED FOR
C6------FLOW DOWN INTO PARTIALLY SATURATED LAYERS.
  201 KT=0
      DO 300 K=1,NLAY
C
C7------SEE IF CORRECTION IS NEEDED FOR LEAKAGE FROM ABOVE.
      IF(LAYCON(K).NE.3 .AND. LAYCON(K).NE.2) GO TO 250
      KT=KT+1
      IF(K.EQ.1) GO TO 250
C
C7A-----FOR EACH CELL MAKE THE CORRECTION IF NEEDED.
      DO 220 I=1,NROW
      DO 220 J=1,NCOL
C
C7B-----IF THE CELL IS EXTERNAL(IBOUND<=0) THEN SKIP IT.
      IF(IBOUND(J,I,K).LE.0) GO TO 220
      HTMP=HNEW(J,I,K)
C
C7C-----IF HEAD IS ABOVE TOP THEN CORRECTION NOT NEEDED
      IF(HTMP.GE.TOP(J,I,KT)) GO TO 220
C
C7D-----WITH HEAD BELOW TOP ADD CORRECTION TERMS TO RHS AND HCOF.
      RHS(J,I,K)=RHS(J,I,K) + CV(J,I,K-1)*TOP(J,I,KT)
      HCOF(J,I,K)=HCOF(J,I,K) + CV(J,I,K-1)
```

```
  220 CONTINUE
C
C8------SEE IF THIS LAYER MAY NEED CORRECTION FOR LEAKAGE TO BELOW.
  250 IF(K.EQ.NLAY) GO TO 300
      IF(LAYCON(K+1).NE.3 .AND. LAYCON(K+1).NE.2) GO TO 300
      KTT=KT+1
C
C8A-----FOR EACH CELL MAKE THE CORRECTION IF NEEDED.
      DO 280 I=1,NROW
      DO 280 J=1,NCOL
C
C8B-----IF CELL IS EXTERNAL (IBOUND<=0) THEN SKIP IT.
      IF(IBOUND(J,I,K).LE.0) GO TO 280
C
C8C-----IF HEAD IN THE LOWER CELL IS LESS THAN TOP ADD CORRECTION
C8C-----TERM TO RHS.
      HTMP=HNEW(J,I,K+1)
      IF(HTMP.LT.TOP(J,I,KTT)) RHS(J,I,K)=RHS(J,I,K)
     1                       - CV(J,I,K)*(TOP(J,I,KTT)-HTMP)
  280 CONTINUE
  300 CONTINUE
C
C9------RETURN
      RETURN
      END


      SUBROUTINE SBCF1H(HNEW,IBOUND,CR,CC,CV,HY,TRPY,DELR,DELC
     1,BOT,TOP,K,KB,KT,KITER,KSTP,KPER,NCOL,NROW,NLAY,IOUT,
     2SHNW,IU,IP,ISN,IPAR)
C
C-----VERSION 1442 31DEC1986 SBCF1H
C-----VERSION MODIFIED FOR MODFLOWP 1000 01FEB1992
C
C
C     ******************************************************************
C     COMPUTE CONDUCTANCE FROM SATURATED THICKNESS AND HYDRAULIC
C     CONDUCTIVITY
C     ******************************************************************
C
C      SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW,HD
C
      DIMENSION HNEW(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY)
     1, CR(NCOL,NROW,NLAY), CC(NCOL,NROW,NLAY), CV(NCOL,NROW,NLAY)
     2,HY(NCOL,NROW,NLAY), TRPY(NLAY), DELR(NCOL), DELC(NROW)
     3, BOT(NCOL,NROW,NLAY),TOP(NCOL,NROW,NLAY),
     4  SHNW(NCOL,NROW,NLAY)
C
      COMMON /FLWCOM/LAYCON(80)
C     ------------------------------------------------------------------
C
C1------CALCULATE TRANSMISSIVITY AT EACH ACTIVE CELL. TRANSMISSIVITY
C1------WILL BE STORED TEMPORARILY IN THE CC ARRAY.
      DO 200 I=1,NROW
      DO 200 J=1,NCOL
C
C2------IF CELL IS INACTIVE THEN SET T=0 & MOVE ON TO NEXT CELL.
      IF(IBOUND(J,I,K).NE.0) GO TO 10
      CC(J,I,K)=0.
      GO TO 200
C
C3------CALCULATE SATURATED THICKNESS.
   10 HD=HNEW(J,I,K)
C-------LINE ADDED FOR PARAMETER ESTIMATION PACKAGE
      IF(IU.NE.0.AND.ISN.NE.-1.AND.IP.NE.0) HD=SHNW(J,I,K)
      IF(LAYCON(K).EQ.1) GO TO 50
      IF(HD.GT.TOP(J,I,KT)) HD=TOP(J,I,KT)
   50 THCK=HD-BOT(J,I,KB)
C
C4------CHECK TO SEE IF SATURATED THICKNESS IS GREATER THAN ZERO.
      IF(THCK.LE.0.) GO TO 100
C
C5------IF SATURATED THICKNESS>0 THEN T=K*THICKNESS.
      CC(J,I,K)=THCK*HY(J,I,KB)
      GO TO 200
C
C6------WHEN SATURATED THICKNESS < 0, PRINT A MESSAGE AND SET
C6------TRANSMISSIVITY, IBOUND, AND VERTICAL CONDUCTANCE =0
  100 WRITE(IOUT,150) K,I,J,KITER,KSTP,KPER
  150 FORMAT(1H0,10('*'),'NODE',3I4,' (LAYER,ROW,COL) WENT DRY'
    1       ,' AT ITERATION =',I3,'  TIME STEP =',I3
```

```
     2       ,'  STRESS PERIOD =',I3)
      HNEW(J,I,K)=1.D30
C--------LINE ADDED FOR PARAMETER-ESTIMATION PACKAGE
      IF(IU.NE.0.AND.IPAR.GE.0) HNEW(J,I,K)=BOT(J,I,KB)+10.
      CC(J,I,K)=0.
      IBOUND(J,I,K)=0
      IF(K.LT.NLAY) CV(J,I,K)=0.
      IF(K.GT.1) CV(J,I,K-1)=0.
      GO TO 200
  200 CONTINUE
C
C7------COMPUTE HORIZONTAL BRANCH CONDUCTANCES FROM TRANSMISSIVITY
      CALL SBCF1C(CR,CC,TRPY,DELR,DELC,K,NCOL,NROW,NLAY)
C
C8------RETURN
      RETURN
      END


      SUBROUTINE RIV1FM(NRIVER,MXRIVR,RIVR,HNEW,HCOF,RHS,IBOUND,
     1                  NCOL,NROW,NLAY,
     2                  SHNW,IU,IP,ISN)
C
C-----VERSION 0915 27AUG1982 RIV1FM
C-----VERSION MODIFIED FOR MODFLOWP 1000 01FEB1992
C
C     ******************************************************************
C     ADD RIVER TERMS TO RHS AND HCOF
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
C
      DOUBLE PRECISION HNEW
      DIMENSION RIVR(6,MXRIVR),HNEW(NCOL,NROW,NLAY),
     1          HCOF(NCOL,NROW,NLAY),RHS(NCOL,NROW,NLAY),
     2          IBOUND(NCOL,NROW,NLAY)
     3          ,SHNW(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C
C1------IF NRIVER<=0 THERE ARE NO RIVERS. RETURN.
      IF(NRIVER.LE.0)RETURN
C
C2------PROCESS EACH CELL IN THE RIVER LIST.
      DO 100 L=1,NRIVER
C
C3------GET COLUMN, ROW, AND LAYER OF CELL CONTAINING REACH
      IL=RIVR(1,L)
      IR=RIVR(2,L)
      IC=RIVR(3,L)
C
C4------IF THE CELL IS EXTERNAL SKIP IT.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 100
C
C5------SINCE THE CELL IS INTERNAL GET THE RIVER DATA.
      HRIV=RIVR(4,L)
      CRIV=RIVR(5,L)
      RBOT=RIVR(6,L)
      HHNEW=HNEW(IC,IR,IL)
C-------ADDED FOR ADJOINT AND SENSITIVITY CALCULATIONS
      IF(IU.NE.0.AND.IP.GT.0) HHNEW=DBLE(SHNW(IC,IR,IL))
C
C6------COMPARE AQUIFER HEAD TO BOTTOM OF STREAM BED.
      IF(HHNEW.LE.RBOT)GO TO 96
C
C7------SINCE HEAD>BOTTOM ADD TERMS TO RHS AND HCOF.
C-------FOR ADJOINT STATES, ONLY CALCULATE CONTRIBUTION TO HCOF.
      IF(IU.EQ.0.OR.IP.EQ.0.OR.ISN.LT.0)
     1 RHS(IC,IR,IL)=RHS(IC,IR,IL)-CRIV*HRIV
      HCOF(IC,IR,IL)=HCOF(IC,IR,IL)-CRIV
      GO TO 100
C
C8------SINCE HEAD<BOTTOM ADD TERM ONLY TO RHS.
   96 IF(IU.EQ.0.OR.IP.EQ.0.OR.ISN.LT.0)
     1 RHS(IC,IR,IL)=RHS(IC,IR,IL)-CRIV*(HRIV-RBOT)
  100 CONTINUE
C
C9------RETURN
      RETURN
      END
```

```
      SUBROUTINE DRN1FM(NDRAIN,MXDRN,DRAI,HNEW,HCOF,RHS,IBOUND,
     1                  NCOL,NROW,NLAY,
     2                  SHNW,IU,IP,ISN)
C
C-----VERSION 1030 10APR1985 DRN1FM
C-----VERSION MODIFIED FOR MODFLOWP 1000 01FEB1992
C
C
C     ******************************************************************
C     ADD DRAIN FLOW TO SOURCE TERM
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
C
      DIMENSION DRAI(5,MXDRN),HNEW(NCOL,NROW,NLAY),
     1          RHS(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),
     1          HCOF(NCOL,NROW,NLAY),
     1          SHNW(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------IF NDRAIN<=0 THERE ARE NO DRAINS. RETURN
      IF(NDRAIN.LE.0) RETURN
C
C2------PROCESS EACH CELL IN THE DRAIN LIST
      DO 100 L=1,NDRAIN
C
C3------GET COLUMN, ROW AND LAYER OF CELL CONTAINING DRAIN.
      IL=DRAI(1,L)
      IR=DRAI(2,L)
      IC=DRAI(3,L)
C
C4-------IF THE CELL IS EXTERNAL SKIP IT.
      IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
C
C5-------IF THE CELL IS INTERNAL GET THE DRAIN DATA.
      EL=DRAI(4,L)
      HHNEW=HNEW(IC,IR,IL)
C-------ADDED FOR ADJOINT AND SENSITIVITY CALCULATIONS
      IF(IU.NE.0.AND.IP.GT.0) HHNEW=SHNW(IC,IR,IL)
C
C6------IF HEAD IS LOWER THAN DRAIN THEN SKIP THIS CELL.
      IF(HHNEW.LE.EL) GO TO 100
C
C7------HEAD IS HIGHER THAN DRAIN. ADD TERMS TO RHS AND HCOF.
C-------FOR ADJOINT STATES, ONLY CALCULATE CONTRIBUTION TO HCOF.
      C=DRAI(5,L)
      HCOF(IC,IR,IL)=HCOF(IC,IR,IL)-C
      IF(IU.EQ.0.OR.IP.EQ.0.OR.ISN.LT.0)
     1 RHS(IC,IR,IL)=RHS(IC,IR,IL)-C*EL
  100 CONTINUE
C
C8------RETURN
      RETURN
      END


      SUBROUTINE EVT1FM(NEVTOP,IEVT,EVTR,EXDP,SURF,RHS,HCOF,
     1                  IBOUND,HNEW,NCOL,NROW,NLAY,
     2                  SHNW,IU,IP,ISN)
C
C-----VERSION 1031 10APR1985 EVT1FM
C-----VERSION MODIFIED FOR MODFLOWP 1000 01FEB1992
C
C     ******************************************************************
C        ADD EVAPOTRANSPIRATION TO RHS AND HCOF
C     ******************************************************************
C
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW
      DIMENSION IEVT(NCOL,NROW),EVTR(NCOL,NROW),EXDP(NCOL,NROW),
     1          SURF(NCOL,NROW),RHS(NCOL,NROW,NLAY),
     2          HCOF(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY),
     3          HNEW(NCOL,NROW,NLAY),
     4          SHNW(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------PROCESS EACH HORIZONTAL CELL LOCATION
      DO 10 IR=1,NROW
      DO 10 IC=1,NCOL
C
```

```
C2------SET THE LAYER INDEX EQUAL TO 1
      IL=1
C
C3------IF OPTION 2 IS SPECIFIED THEN GET LAYER INDEX FROM IEVT ARRAY
      IF(NEVTOP.EQ.2)IL=IEVT(IC,IR)
C
C4------IF THE CELL IS EXTERNAL IGNORE IT.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 10
      C=EVTR(IC,IR)
      S=SURF(IC,IR)
      H=HNEW(IC,IR,IL)
C-------ADDED FOR ADJOINT AND SENSITIVITY CALCULATIONS
      IF(IU.NE.0.AND.IP.GT.0) H=SHNW(IC,IR,IL)
C
C5------IF AQUIFER HEAD IS GREATER THAN OR EQUAL TO SURF, ET IS CONSTANT
      IF(H.LT.S) GO TO 5
C
C5A-----SUBTRACT -EVTR FROM RHS
C-------FOR ADJOINT STATES, ONLY CALCULATE CONTRIBUTION TO HCOF
      IF(IU.EQ.0.OR.IP.EQ.0.OR.ISN.LT.0)
     1 RHS(IC,IR,IL)=RHS(IC,IR,IL) + C
      GO TO 10
C
C6------IF DEPTH TO WATER>=EXTINCTION DEPTH THEN ET IS 0
    5 D=S-H
      X=EXDP(IC,IR)
      IF(D.GE.X)GO TO 10
C
C7------LINEAR RANGE. ADD ET TERMS TO BOTH RHS AND HCOF.
C-------FOR ADJOINT STATES, ONLY CALCULATE CONTRIBUTION TO HCOF
      IF(IU.EQ.0.OR.IP.EQ.0.OR.ISN.LT.0)
     1 RHS(IC,IR,IL)=RHS(IC,IR,IL)+C-C*S/X
      HCOF(IC,IR,IL)=HCOF(IC,IR,IL)-C/X
   10 CONTINUE
C
C8------RETURN
      RETURN
      END


      SUBROUTINE STR1FM(NSTREM,STRM,ISTRM,HNEW,HCOF,RHS,IBOUND,MXSTRM,
     1                 NCOL,NROW,NLAY,IOUT,NSS,ITRBAR,NTRIB,ARTRIB,
     2                       IDIVAR,ICALC,CONST,
     3                       SHNW,IU,IP,ISN)
C
C-----VERSION   1 23OCT1987 STR1FM
C-----VERSION MODIFIED FOR MODFLOWP 1000 01FEB1992
C
C     ******************************************************************
C     ADD STREAM TERMS TO RHS AND HCOF IF FLOW OCCURS IN MODEL CELL
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
C
      DOUBLE PRECISION HNEW
      DIMENSION STRM(11,MXSTRM),ISTRM(5,MXSTRM),HNEW(NCOL,NROW,NLAY),
     1          HCOF(NCOL,NROW,NLAY),RHS(NCOL,NROW,NLAY),
     2          IBOUND(NCOL,NROW,NLAY),ITRBAR(NSS,NTRIB),ARTRIB(NSS),
     3          IDIVAR(NSS),SHNW(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------IF NSTREM<=0 THERE ARE NO STREAMS. RETURN.
      IF(NSTREM.LE.0)RETURN
C
C2------PROCESS EACH CELL IN THE STREAM LIST.
C
C3------DETERMINE LAYER, ROW, COLUMN OF EACH REACH.
      DO 500 L=1,NSTREM
      LL=L-1
      IL=ISTRM(1,L)
      IR=ISTRM(2,L)
      IC=ISTRM(3,L)
C
C4------DETERMINE IF CELL IS OUTSIDE OF MODEL BOUNDARIES.
      IF(IBOUND(IC,IR,IL).LE.0)GO TO 500
C
C5------DETERMINE STREAM SEGMENT AND REACH NUMBER.
      ISTSG=ISTRM(4,L)
      NREACH=ISTRM(5,L)
C
C6------SET FLOWIN EQUAL TO STREAM SEGMENT INFLOW IF FIRST REACH.
      IF(NREACH.GT.1) GO TO 200
```

```
      FLOWIN=STRM(1,L)
C
C7------STORE OUTFLOW FROM PREVIOUS SEGMENT IN ARTRIB IF SEGMENT >1.
      IF(ISTSG.GT.1) IFLG = ISTRM(4,LL)
      IF(ISTSG.GT.1)  ARTRIB(IFLG)=STRM(9,LL)
C
C8------IF SEGMENT IS A DIVERSION, COMPUTE FLOW OUT OF UPSTREAM REACH.
      IF(IDIVAR(ISTSG).LE.0) GO TO 50
      NDFLG=IDIVAR(ISTSG)
      DUM=ARTRIB(NDFLG)-FLOWIN
      IF(DUM.GE.0.0) ARTRIB(NDFLG)=DUM
      IF(DUM.GE.0.0) GO TO 50
      FLOWIN=0.
   50 IF(FLOWIN.GE.0.0) GO TO 300
C
C9-----SUM TRIBUTARY OUTFLOW AND USE AS INFLOW INTO DOWNSTREAM SEGMENT.
      FLOWIN =0.
      DO 100 ITRIB=1,NTRIB
      INODE=ITRBAR(ISTSG,ITRIB)
      IF(INODE.LE.0) GO TO 100
      FLOWIN=FLOWIN+ARTRIB(INODE)
  100 CONTINUE
C
C10-----IF REACH >1, SET INFLOW EQUAL TO OUTFLOW FROM UPSTREAM REACH.
  200 IF(NREACH.GT.1) FLOWIN=STRM(9,LL)
C
C11----COMPUTE STREAM STAGE IN REACH IF ICALC IS GREATER THAN 1.
  300 IF(ICALC.LE.0) GO TO 310
      XNUM=((FLOWIN+STRM(9,L))/2.0)*STRM(8,L)
      DNOM=CONST*STRM(6,L)*(SQRT(STRM(7,L)))
      DEPTH=(XNUM/DNOM)**0.6
      IF(DEPTH.LE.0.) DEPTH=0.
      STRM(2,L)=DEPTH+STRM(5,L)
  310 HSTR=STRM(2,L)
C
C12----DETERMINE LEAKAGE THROUGH STREAMBED.
      IF(FLOWIN.LE.0.) HSTR=STRM(5,L)
      CSTR=STRM(3,L)
      SBOT=STRM(4,L)
      H=HNEW(IC,IR,IL)
C-----ADDED FOR PARAMETER ESTIMATION
      IF(IU.GT.0.AND.IP.NE.0) H=SHNW(IC,IR,IL)
      T=HSTR-SBOT
C
C13----COMPUTE LEAKAGE AS A FUNCTION OF STREAM STAGE AND HEAD IN CELL.
      FLOBOT=CSTR*(HSTR-H)
C
C14----RECOMPUTE LEAKAGE IF HEAD IN CELL IS BELOW STREAMBED BOTTOM.
      IQFLG=0
      IF(H.GT.SBOT) GO TO 312
      IQFLG=1
      FLOBOT=CSTR*T
C
C15----SET LEAKAGE EQUAL TO STREAM INFLOW IF LEAKAGE MORE THAN INFLOW.
  312 IF(FLOBOT.LE.FLOWIN) GO TO 320
      IQFLG=1
      FLOBOT=FLOWIN
C
C16-----STREAMFLOW OUT EQUALS STREAMFLOW IN MINUS LEAKAGE.
  320 FLOWOT=FLOWIN-FLOBOT
      IF((ISTSG.GT.1).AND.(NREACH.EQ.1)) STRM(9,LL)=ARTRIB(IFLG)
C
C17----STORE STREAM INFLOW, OUTFLOW AND LEAKAGE FOR EACH REACH.
      STRM(9,L)=FLOWOT
      STRM(10,L)=FLOWIN
      STRM(11,L)=FLOBOT
C
C18----RETURN TO STEP 3 IF STREAM INFLOW IS LESS THAN OR EQUAL TO ZERO
C        AND LEAKAGE IS GREATER THAN OR EQUAL TO ZERO.
      IF((FLOWIN.LE.0.0).AND.(FLOBOT.GE.0.0)) GO TO 500
C
C19------IF HEAD > BOTTOM THEN ADD TERMS TO RHS AND HCOF.
      IF(IQFLG.GT.0) GO TO 400
C-------FOR ADJOINT STATES, ONLY CALCULATE CONTRIBUTION TO HCOF
      IF(IU.EQ.0.OR.IP.EQ.0.OR.ISN.LT.0)
     1RHS(IC,IR,IL)=RHS(IC,IR,IL)-CSTR*HSTR
      HCOF(IC,IR,IL)=HCOF(IC,IR,IL)-CSTR
      GO TO 500
C
C20------IF HEAD < BOTTOM THEN ADD TERM ONLY TO RHS.
C-------FOR ADJOINT STATES, ONLY CALCULATE CONTRIBUTION TO RHS
  400 IF(IU.EQ.0.OR.IP.EQ.0.OR.ISN.LT.0)
     1RHS(IC,IR,IL)=RHS(IC,IR,IL)-FLOBOT
```

```
  500 CONTINUE
C
C22-----RETURN.
      RETURN
      END


      SUBROUTINE GHB1FM(NBOUND,MXBND,BNDS,HCOF,RHS,IBOUND,
     1              NCOL,NROW,NLAY,
     2              IU,IP,ISN)
C
C-----VERSION 1037 10APR1985 GHB1FM
C-----VERSION MODIFIED FOR MODFLOWP 1000 01FEB1992
C
C     ******************************************************************
C     ADD GHB TERMS TO RHS AND HCOF
C     ******************************************************************
C
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION BNDS(5,MXBND),HCOF(NCOL,NROW,NLAY),
     1          RHS(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
C
C1------IF NBOUND<=0 THEN THERE ARE NO GENERAL HEAD BOUNDS. RETURN.
      IF(NBOUND.LE.0) RETURN
C
C2------PROCESS EACH ENTRY IN THE GENERAL HEAD BOUND LIST (BNDS)
      DO 100 L=1,NBOUND
C
C3------GET COLUMN, ROW AND LAYER OF CELL CONTAINING BOUNDARY
      IL=BNDS(1,L)
      IR=BNDS(2,L)
      IC=BNDS(3,L)
C
C4------IF THE CELL IS EXTERNAL THEN SKIP IT.
      IF(IBOUND(IC,IR,IL).LE.0) GO TO 100
C
C5------SINCE THE CELL IS INTERNAL GET THE BOUNDARY DATA.
      HB=BNDS(4,L)
      C=BNDS(5,L)
C
C6------ADD TERMS TO RHS AND HCOF
      HCOF(IC,IR,IL)=HCOF(IC,IR,IL)-C
C-------FOR ADJOINT STATES, ONLY CALCULATE CONTRIBUTION TO RHS
      IF(IU.EQ.0.OR.IP.EQ.0.OR.ISN.LT.0)
     1 RHS(IC,IR,IL)=RHS(IC,IR,IL)-C*HB
  100 CONTINUE
C
C7------RETURN
      RETURN
      END
```

New SEN Primary Modules

Listed in order of first appearance in MAIN

```
C=======================================================================
      SUBROUTINE SEN1AL (ISUM,LENX,NCOL,NROW,NLAY,NPER,IOUT,LCSHNW,
     1             LCNLL,LCSMAT,LCCELS,IUHEAD,NP,NSM,NSN,ISN,LCTS,LCATS,
     2             LCST,NH,LCNDER,LCSHOL,LCSTP,NRWD,LCIRWD,IU,LCBSTP,IPAR,
     3             LCB,LCH,LCX,LCRINT,LCJOFF,LCIOFF,LCCONV,LCCOFF,LCROFF,
     4             LCMLAY,LCPR,MOBS,LCLN,NQ,NQC,NQT,LCIBT,LCNQOB,LCNQCL,
     5             LCIQOB,LCQCLS,IUBD,ISCALS,LCTOFF,MPR,LCXX,MAXM,LCLZ,
     6             NLLI1,NLLIT,NMM,NZM,LZI1,LCSFAC,LCLM,LCMATZ,LCHOBS,
     7             LCWT,IOUB,LCSV,IPRNP,IUNHEA,NUMR,ICHECK,NPNG,LCIPNG,
     8             NZER,LCIZER,LCIWPG,NOPT,LCXD)
C-----VERSION 1000 01FEB1992 SEN1AL
C     ******************************************************************
C     ALLOCATE ARRAY STORAGE FOR SENSITIVITIES
C     ******************************************************************
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 HEADNG(20)
      COMMON /FLWCOM/LAYCON(80)
C     ------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE AND PRINT HEADING
      WRITE(IOUT,1) IU
    1 FORMAT(1H0,'SEN1 -- SENSITIVITY PACKAGE,  ,
     1  'VERSION 1, 12/01/91 INPUT READ FROM UNIT',I3)
      READ(IU,2) HEADNG
```

```
    2 FORMAT(20A4)                                                          SEN0026
      WRITE(IOUT,3) HEADNG                                                  SEN0027
    3 FORMAT(1H0,20A4)                                                      SEN0028
      READ(IU,2) HEADNG                                                     SEN0029
      WRITE(IOUT,3) HEADNG                                                  SEN0030
C                                                                           SEN0031
C-------READ & PRINT LINES 3 THROUGH 8 OF THE INPUT FILE                    SEN0032
      READ(IU,10) NP,NSM,NSN,NLLI1,LZI1,NMM,NZM                             SEN0033
      READ(IU,10) IPRNP,ICHECK,NUMR                                         SEN0034
      READ(IU,10) NH,MOBS,MAXM,NQ,NQC,NQT,MPR                               SEN0035
      READ(IU,10) IPAR,ISN,ISCALS,NOPT                                      SEN0036
      READ(IU,10) IUHEAD,IOUB,IUBD,IUNHEA                                   SEN0037
      READ(IU,10) NRWD,NZER,NPNG                                            SEN0038
   10 FORMAT(15I5)                                                          SEN0039
      WRITE(IOUT,20) NP,NSM,NSN,NLLI1,LZI1,NMM,NZM                          SEN0040
   20 FORMAT(                                                               SEN0041
     1    ' NUMBER OF PARAMETERS..............................................:',I5,/,  SEN0042
     2    ' NUMBER NLL>0 IN ALL DATA SETS 2 FOR PID=T,KV,',/,               SEN0043
     3    '  TKV, AND S1, +1 FOR NLL(1)>0 FOR PID=RCH OR ETM...:',I5,/,     SEN0044
     4    ' SUM ALL |NLL(1)| FOR PID=Q,KRB,KDR,KST,GHB,AND CH..:',I5,/,     SEN0045
     5    ' LARGEST NLL VECTOR IN ALL DATA SETS 2..............:',I5,/,     SEN0046
     6    ' LARGEST LZ VECTOR IN ALL DATA SETS 2B..............:',I5,/,     SEN0047
     7    ' NUMBER OF MULTIPLICATION MATRICES IN DATA SET 3....:',I5,/,     SEN0048
     8    ' NUMBER OF ZONE MATRICES IN DATA SET 4..............:',I5,/)     SEN0049
      WRITE(IOUT,30) IPRNP,ICHECK,NUMR                                      SEN0050
   30 FORMAT(                                                               SEN0051
     3    ' PRINT MATRICES PRODUCED BY INIT. PARAM. EST. (>0)..:',I5,/,     SEN0052
     4    ' CHECK CELL REPETITIONS FOR MATRIX PARAMETERS (>0)..:',I5,/,     SEN0053
     5    '    NUMBER OF CELL REPETITIONS ALLOWED..............:',I5,/)     SEN0054
      WRITE(IOUT,40) NH,MOBS,MAXM,NQ,NQC,NQT,MPR                            SEN0055
   40 FORMAT(                                                               SEN0056
     1    ' NUMBER OF HEADS....................................:',I5,/,     SEN0057
     2    '    NUMBER OF MULTILAYER HEADS......................:',I5,/,     SEN0058
     3    '    MAXIMUM NUMBER OF LAYERS FOR MULTILAYER HEADS....:',I5,/,    SEN0059
     4    ' NUMBER OF FLUX CELL GROUPS.........................:',I5,/,     SEN0060
     5    '    NUMBER OF CELLS IN FLUX CELL GROUPS.............:',I5,/,     SEN0061
     6    '    NUMBER OF FLUXES................................:',I5,/,     SEN0062
     7    ' NUMBER OF PRIORS WITH MULTIPLE PARAMETERS..........:',I5,/)     SEN0063
      WRITE(IOUT,50) IPAR,ISN,ISCALS,NOPT                                   SEN0064
   50 FORMAT(                                                               SEN0065
     1    ' HEADS (<0), HEADS AND SENS. (0), OR PAR. EST. (1)..:',I5,/,     SEN0066
     2    ' ADJOINT (+) OR SEN. EQ. (-) CALCULATIONS...........:',I5,/,     SEN0067
     3    '    (GRADIENT SEARCH (+) OR GAUSS-NEWTON (-) REGRESSION)',/,     SEN0068
     4    ' SCALE PRINTED SENSITIVITIES (>0)...................:',I5,/,     SEN0069
     5    ' ADJUST GAUSS-NEWTON MATRIX WITH NEWTON UPDATES.....:',I5,/)     SEN0070
      WRITE(IOUT,60) IUHEAD,IOUB,IUBD,IUNHEA                                SEN0071
   60 FORMAT(                                                               SEN0072
     1    ' OUTPUT/INPUT UNIT FOR HEADS OR SENSITIVITIES.......:',I5,/,     SEN0073
     2    ' OUTPUT/INPUT UNIT NUMBER FOR INFO FROM ITERATIONS..:',I5,/,     SEN0074
     3    ' INPUT UNIT FOR SS/TR IBOUND ARRAYS.................:',I5,/,     SEN0075
     4    ' INPUT UNIT FOR KNOWN HEADS AT CONSTANT-HEAD BNDRYS.:',I5,/)     SEN0076
      WRITE(IOUT,70) NRWD,NZER,NPNG                                         SEN0077
   70 FORMAT(                                                               SEN0078
     1    ' NUMBER OF INPUT UNITS IDENTIFIED IN RCH AND EVT....:',I5,/,     SEN0079
     2    ' NUMBER OF TIME STEPS>0 AT WHICH HEAD=0.............:',I5,/,     SEN0080
     3    ' NUMBER OF USUALLY POS. PARAMETERS THAT MAY BE NEG..:',I5,/)     SEN0081
      NLLIT=(NLLI1-1)/2                                                     SEN0082
      LZI1=LZI1+1                                                           SEN0083
      IF(IPAR.LT.1) MPR=0                                                   SEN0084
      IF(MAXM.EQ.1) THEN                                                    SEN0085
        WRITE(IOUT,80)                                                      SEN0086
        STOP                                                                SEN0087
      ENDIF                                                                 SEN0088
   80 FORMAT(/,' MAXM CAN NOT EQUAL 1 -- STOP EXECUTION')                   SEN0089
C-------STORE, IN ISOLD, LOCATION OF FIRST UNALLOCATED SPACE IN X.          SEN0090
      ISOLD=ISUM                                                            SEN0091
C                                                                           SEN0092
C2------ALLOCATE SPACE FOR ARRAYS                                           SEN0093
C----------TIME STEP ARRAYS                                                 SEN0094
      NODES=NROW*NCOL*NLAY                                                  SEN0095
      LCTS=ISUM                                                             SEN0096
      IF(NPER.NE.0) ISUM=ISUM+NPER                                          SEN0097
      IF(NPER.EQ.0) ISUM=ISUM+1                                             SEN0098
      LCATS=ISUM                                                            SEN0099
      IF(ISN.GT.0) ISUM=ISUM+(2*(NPER+1))                                   SEN0100
      LCSTP=ISUM                                                            SEN0101
      ISUM=ISUM+NPER+1                                                      SEN0102
      LCBSTP=ISUM                                                           SEN0103
      IF(ISN.GT.0) ISUM=ISUM+NPER+1                                         SEN0104
C----------FULL GRID ARRAYS                                                 SEN0105
      LCSHNW=ISUM                                                           SEN0106
      ISUM=ISUM+NODES                                                       SEN0107
      LCST=ISUM                                                             SEN0108
```

```
      ISUM=ISUM+NODES
      LCSV=ISUM
      IF(LAYCON(1).EQ.1.AND.NLAY.GT.1.AND.IPAR.GT.0) ISUM=ISUM+NROW*NCOL
      LCSHOL=ISUM
      IF((ISN.LT.0.OR.ISN.EQ.10).AND.NPER.NE.0) ISUM=ISUM+NODES
C----------ARRAYS FOR PARAMETER DEFINITION
      LCLN=ISUM
      ISUM=ISUM+NP
      LCNLL=ISUM
      ISUM=ISUM+NP*NLLI1
      LCSFAC=ISUM
      ISUM=ISUM+NSM
      LCLM=ISUM
      ISUM=ISUM+NSM
      LCLZ=ISUM
      ISUM=ISUM+NSM*LZI1
      LCSMAT=ISUM
      ISUM=ISUM+NMM*NROW*NCOL
      LCMATZ=ISUM
      ISUM=ISUM+NZM*NROW*NCOL
      LCCELS=ISUM
      ISUM=ISUM+(NSN*4)
      LCB=ISUM
      ISUM=ISUM+NP
      LCIWPG=ISUM
      ISUM=ISUM+NP
C----------HEAD DATA ARRAYS
      LCNDER=ISUM
      ISUM=ISUM+5*NH
      LCCOFF=ISUM
      ISUM=ISUM+NH
      LCROFF=ISUM
      ISUM=ISUM+NH
      LCIOFF=ISUM
      ISUM=ISUM+NH
      LCJOFF=ISUM
      ISUM=ISUM+NH
      LCRINT=ISUM
      ISUM=ISUM+4*NH
      LCMLAY=ISUM
      ISUM=ISUM+MAXM*MOBS
      LCPR=ISUM
      ISUM=ISUM+MAXM*MOBS
C----------FLOW DATA ARRAYS
      LCIBT=ISUM
      ISUM=ISUM+NQ*2
      LCNQOB=ISUM
      ISUM=ISUM+NQ
      LCNQCL=ISUM
      ISUM=ISUM+NQ
      LCIQOB=ISUM
      ISUM=ISUM+NQT
      LCQCLS=ISUM
      ISUM=ISUM+5*NQC
C----------ARRAYS USED FOR ALL DEPENDENT-VARIABLE DATA
      LCTOFF=ISUM
      ISUM=ISUM+NH+NQT
      LCH=ISUM
      ISUM=ISUM+NH+NQT
      LCHOBS=ISUM
      ISUM=ISUM+NH+NQT
      LCWT=ISUM
      ISUM=ISUM+NH+NQT
      IF(MOD(ISUM,2).EQ.0) ISUM=ISUM+1
      LCX=ISUM
      ISUM=ISUM+NP*(NH+NQT)
      LCXD=ISUM
      IF(IPAR.GT.0.AND.NOPT.EQ.1) ISUM=ISUM+NP*(NH+NQT)
      LCXX=ISUM
      IF(ISN.GT.0) ISUM=ISUM+2*NP
C----------MISCELLANEOUS ARRAYS
      LCIRWD=ISUM
      ISUM=ISUM+NRWD
      LCIPNG=ISUM
      ISUM=ISUM+NPNG
      LCIZER=ISUM
      ISUM=ISUM+NZER
      LCCONV=ISUM
      IF(ISN.LT.0.OR.ISN.GE.4) ISUM=ISUM+2*NP+2
C
C8------PRINT AMOUNT OF SPACE USED BY SENSITIVITY PACKAGE.
      ISP=ISUM-ISOLD
      WRITE (IOUT,4)ISP
```

```
    4 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED FOR SENSITIVITIES')
      ISUM1=ISUM-1
      WRITE(IOUT,5)ISUM1,LENX
    5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
    6 FORMAT(1X,'    ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C9------RETURN
      RETURN
      END


C=======================================================================
      SUBROUTINE SEN1RP (
     1       NCOL,NROW,NLAY,NSN,NSM,MXBND,MXWELL,MXRIVR,NPER,IU,IOUT,NP,
     2       PID,DID,NLL,CELS,SMAT,NH,NDER,JT,NRWD,IRWD,IBOUND,ISN,JOFF,
     3       IOFF,HOBS,WT,B,B1,IPAR,CONV,HCLOSE,RCLOSE,TOP,BOT,DELR,DELC,
     4       RINT,COFF,ROFF,MLAY,PR,MOBS,LN,NQ,NQC,NQT,IBT,NQOB,NQCL,
     6       IQOB,QCLS,IUBD,BUFF,TOFF,MXDRN,MXSTRM,MAXM,NMM,NZM,LZI1,
     7       SFAC,LZ,LM,MATZ,NLLI1,IUNIT,NLLIT,EV,NUMR,ICHECK,NPNG,IPNG,
     8       NZER,IZER,IWPG,NPO)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     READ, CHECK AND STORE DATA FOR SENSITIVITY CALCULATIONS AND
C     PARAMETER ESTIMATION, AND INITIALIZE SOME VARIABLES.
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4   PID(NPO),DID(NH+NQT),PIDTMP
      INTEGER       NLL(NLLI1,NPO),IUNIT(24)
      DIMENSION NDER(5,NH),CELS(4,NSN),SMAT(NCOL,NROW,NMM),IWPG(NPO),
     1     IRWD(NRWD),IBOUND(NCOL,NROW,NLAY),IOFF(NH),JOFF(NH),
     2     WT(NH+NQT),B(NPO),B1(NPO),CONV(2*NPO+2),TOP(NCOL,NROW,NLAY),
     3     BOT(NCOL,NROW,NLAY),DELR(NCOL),DELC(NROW),RINT(4,NH),
     4     COFF(NH),ROFF(NH),MLAY(MAXM,MOBS),PR(MAXM,MOBS),LN(NPO),
     5     NQOB(NQ),NQCL(NQ),IQOB(NQT),QCLS(4,NQC),HOBS(NH+NQT),
     6     BUFF(NCOL,NROW,NLAY),TOFF(NH,NQT),IBT(2,NQ),LZ(LZI1,NSM),
     7     LM(NSM),SFAC(NSM),MATZ(NCOL,NROW,NZM),IPNG(NPNG),IZER(NZER)
      COMMON /FLWCOM/LAYCON(80)
C     ------------------------------------------------------------------
503   FORMAT(8F10.0)
504   FORMAT(2F10.0,2I5,F10.0)
505   FORMAT(16I5)
510   FORMAT(8I10)
530   FORMAT(//,' DEFINITION OF TEMPORAL AND SPATIAL EXTENT OF ',
     3          'PARAMETERS (PARAMETERIZATION)')
531   FORMAT(/,' INPUT UNIT NUMBERS LISTED IN RCH AND EVT PACKAGE '
     2  ,'INPUT FILES :',9I5)
535   FORMAT(/,' PARAMETER NUMBERS OF AQUIFER PROPERTIES DEFINED ',
     1  'WITH KRIGING OPTION 2 :',9I5)
540   FORMAT(/,' TIME STEPS AT WHICH HEADS ARE SET TO ZERO :',9I5)
550   FORMAT(/,' IPNG(',I3,')=',I3,'; PID(',I3,')= ',A4,/,
     1  ' INCLUSION OF THIS PARAMETER IN IPNG IS WRONG OR UNNECESSARY',
     2  ' -- STOP EXECUTION (SEN1RP)')
575   FORMAT(/,' INPUT ERROR VARIANCES FOR HEADS AND FLOWS: ',2G15.4,
     1/,' INPUT UNIT NUMBERS FOR HEADS AND FLOWS :',2I5,
     2/,' ESTIMATED COMMON ERROR VARIANCE :',G15.4)
580   FORMAT(/,' PARAMETER INFORMATION',/,
     1 '  (CONVERGENCE CRITERIA LISTED HERE ARE USED TO SOLVE FOR ',
     1 'SENSITIVITY-EQUATION SENSITIVITIES)',/,'     #  ID   ',
     1 'INITIAL VALUE  LN      CONVERGENCE CRITERIA  GROUP #')
582   FORMAT(/,' PARAMETER INFORMATION',/,'      #  ID   ',
     1 'INITIAL VALUE  LN  GROUP #')
585   FORMAT(' ',I5,2X,A4,G14.4,I3,3X,2G13.3,I5)
587   FORMAT(' ',I5,2X,A4,G14.4,I3,3X,I5)
590   FORMAT(/,' LAYCON OF THE BLOCK-CENTERED FLOW PACKAGE MAY',
     1    ' ONLY EQUAL 0 OR 1 -- STOP EXECUTION (SEN1RP)')
595   FORMAT(/,' CONJUGATE-DIRECTION OPTIMIZATION NOT PROGRAMMED ',
     1       'FOR FLOW DATA -- STOP EXECUTION (SEN1RP)')
600   FORMAT(//,' EXECUTION TERMINATED DUE TO ERRORS IN',
     1    ' PARAMETER-ESTIMATION PACKAGE INPUT FILE',/,
     1    ' -- SEARCH ABOVE FOR STOP EXECUTION TO FIND',
     1    '  ERROR MESSAGES (SEN1RP)')
610   FORMAT(/,' CONJUGATE-DIRECTION OPTIMIZATION NOT PROGRAMMED ',
     1   'FOR NZER>0 -- STOP EXECUTION (SEN1RP)')
C
      DO 10 I=1,NP
        PID(I)='    '
10    CONTINUE
C-------BEGIN READING INPUT FILE
      IF(NRWD.GT.0) THEN
        READ(IU,505) (IRWD(I),I=1,NRWD)
        WRITE(IOUT,531) (IRWD(I),I=1,NRWD)
```

```
      ENDIF
      IF(NZER.GT.0) THEN
        READ(IU,505) (IZER(I),I=1,NZER)
        WRITE(IOUT,540) (IZER(I),I=1,NZER)
      ENDIF
      IF(NPNG.GT.0) THEN
        READ(IU,505) (IPNG(I),I=1,NPNG)
        WRITE(IOUT,535) (IPNG(I),I=1,NPNG)
      ENDIF
      WRITE(IOUT,530)
C-------READ PARAMETER DEFINITIONS
      CALL SSEN1G(NCOL,NROW,NLAY,NSN,NSM,MXBND,MXWELL,MXRIVR,NPER,IU,
     1    IOUT,NP,PID,NLL,CELS,SMAT,IBOUND,TOP,BOT,LN,IERR,IUBD,BUFF,
     2    DELR,DELC,MXDRN,MXSTRM,NMM,NZM,LZI1,SFAC,LZ,LM,MATZ,
     3    NLLI1,IUNIT)
C-------CHECK NPNG PARAMETER NUMBERS
      IF(NPNG.GT.0) THEN
        DO 50 I=1,NPNG
          PIDTMP=PID(IPNG(I))
          IF(PIDTMP.EQ.'ANI '.OR.PIDTMP.EQ.'ANIV'.OR.PIDTMP.EQ.'Q   '
     1        .OR.PIDTMP.EQ.'CH  '.OR.PIDTMP.EQ.'RCH ') THEN
            IERR=1
            WRITE(IOUT,550) I,IPNG(I),IPNG(I),PIDTMP
          ENDIF
   50   CONTINUE
      ENDIF
C-------NZER>0 OKAY ONLY FOR ISN<0
      IF(NZER.GT.0.AND.ISN.GT.0) THEN
        WRITE(IOUT,610)
        IERR=1
      ENDIF
C-------READ MULTIPLICATIVE FACTOR FOR VARIANCES OF HEADS AND FLOWS
      READ(IU,504) EVH,EVF,IUH,IUF,EV
      WRITE(IOUT,575) EVH,EVF,IUH,IUF,EV
C
C-------READ HEAD TIMES, LOCATIONS, OBSERVED VALUES, AND STATS FOR WTS
C       CALL TIMER(6)
      IF(IUH.EQ.0) IUH=IU
      IF(NH.GT.0)
     1 CALL SSEN1H(NCOL,NROW,NLAY,NPER,IUH,IOUT,DID,NH,NDER,JT,IBOUND,
     1     JOFF,IOFF,HOBS,WT,DELR,DELC,RINT,COFF,ROFF,MLAY,PR,
     2     MOBS,IERR,TOFF,EVH,ISN,MAXM)
C
C-------READ HEAD-DEPENDENT BOUNDARY FLOW DATA
C        CALL TIMER(7)
       IF(IUF.EQ.0) IUF=IU
       IF(NQ.GT.0)
     1  CALL SSEN1F(NCOL,NROW,NLAY,MXBND,MXRIVR,NPER,IUF,IOUT,
     2     DID,NH,JT,IBOUND,NQ,NQC,NQT,IBT,NQOB,NQCL,IQOB,QCLS,IERR,
     3     PID,NP,NLL,CELS,NSN,WT,HOBS,TOFF,EVF,MXDRN,MXSTRM,ISN,
     4     NLLI1,NLLIT)
C      CALL TIMER(8)
C------MAKE SURE THERE ARE NO CONVERTIBLE LAYERS
      DO 90 K=1,NLAY
        IF(LAYCON(K).GT.1) THEN
          WRITE(IOUT,590)
          IERR=1
        ENDIF
   90 CONTINUE
C------CONJUGATE-DIRECTION METHODS NOT PROGRAMMED FOR FLUX DATA
      IF(ISN.GT.0.AND.NQT.GT.0) THEN
        WRITE(IOUT,595)
        IERR=1
      ENDIF
C-------CHECK FOR MULTIPLE USE OF CELLS IN DEFINING MATRIX MODEL INPUTS
      IF(ICHECK.NE.0) CALL SSEN1A(PID,LM,LZ,NLL,SMAT,MATZ,BUFF,NCOL,
     1 NROW,NLAY,NSM,NP,NMM,NZM,LZI1,NLLI1,NLLIT,JT+1,IOUT,IERR,NUMR)
C------STOP IF THERE ARE ERRORS IN THE DATA SET
      IF(IERR.GT.0) THEN
        WRITE(IOUT,600)
        STOP
      ENDIF
C-------CROP TIME STEPS IN NLL FOR PARAMETERSS USED TO CALCULATE
C-------ADJOINT STATES.
      IF(ISN.GT.0) THEN
        DO 126 IP=1,NP
          PIDTMP=PID(IP)
          IF(PIDTMP.EQ.'KRB '.OR.PIDTMP.EQ.'GHB '.OR.
     1       PIDTMP.EQ.'KDR '.OR.PIDTMP.EQ.'KST ') THEN
            DO 128 NT=1,NLLIT
              IF(NLL(2*NT,IP).LE.JT.AND.NLL(2*NT+1,IP).GT.JT)
     1                 NLL(2*NT+1,IP)=JT
  128       CONTINUE
```

```
          ENDIF
  126   CONTINUE
        ENDIF
C-------READ INITIAL PARAMETER VALUES.
        READ(IU,503)(B(I),I=1,NP)
C-------PARAMETER GROUP NUMBERS
        READ(IU,510)(IWPG(IP),IP=1,NP)
C-------SAVE CONVERGENCE CRITERIA FOR HEAD.
        CONV(1)=HCLOSE
        CONV(NP+2)=RCLOSE
C-------CALCULATE CONVERGENCE CRITERIA FOR SENSITIVITY-EQ. SENSITIVITIES
        IF(ISN.LT.0.OR.ISN.GE.4) THEN
          DO 130 IP=1,NP
            CONV(1+IP)=0.
            CONV(NP+2+IP)=0.
            IF(IWPG(IP).GT.0) THEN
              BB=ABS(B(IP))
              IF(BB.LT.1.E-15) BB=1.0
              CONV(1+IP)=HCLOSE/(BB*100.)
              CONV(NP+2+IP)=RCLOSE/(BB*100.)
            ENDIF
  130   CONTINUE
        ENDIF
C-------CONVERT HEAD AND FLOW OBSERVATION VARIANCES TO WEIGHTS.
        ND=NH+NQT
        DO 135 N=1,ND
  135 WT(N)=EV/WT(N)
C-------IF THERE IS NO REGRESSION, PRINT PARAMETER INFORMATION
        IF(IPAR.LE.0) THEN
          IF(ISN.LT.0.OR.ISN.GE.4) WRITE(IOUT,580)
          IF(ISN.GT.0.AND.ISN.LT.4) WRITE(IOUT,582)
          DO 140 IP=1,NP
            IF(ISN.LT.0.OR.ISN.GE.4) WRITE(IOUT,585) IP,PID(IP),B(IP),
     1         LN(IP),CONV(1+IP),CONV(NP+2+IP),IWPG(IP)
            IF(ISN.GT.0.AND.ISN.LT.4) WRITE(IOUT,587) IP,PID(IP),B(IP),
     1         LN(IP)
  140   CONTINUE
        ENDIF
C-------OMIT PARAMETERS WITH NEGATIVE GROUP NUMBERS
        DO 150 IIP=1,NPO
          IF(IWPG(IIP).LT.0) THEN
            NP=IIP-1
            GO TO 155
          ENDIF
  150 CONTINUE
  155 CONTINUE
C-------IF THERE IS REGRESSION, SAVE INITIAL PARAMETER VALUES AND
C-------CONVERT INDICATED PARAMETERS TO NATURAL LOGS
        IF(IPAR.GT.0) THEN
          DO 160 IP=1,NP
            B1(IP)=B(IP)
  160       IF(LN(IP).GT.0) B1(IP)=ALOG(B(IP))
        ENDIF
        RETURN
        END


C=========================================================================
        SUBROUTINE SEN1ST(JT,INBAS,PERLEN,NSTP,TS,ATS,STP,ISN,BSTP)
C-----VERSION 1000 01FEB1992
C       ******************************************************************
C       PRODUCE TIME STEP FILES REQUIRED TO CALCULATE ADJOINT
C          STATES AND SENSITIVITIES
C       ******************************************************************
C       SPECIFICATIONS:
C       ------------------------------------------------------------------
        DIMENSION TS(JT),ATS((2*JT)+2),STP(JT+1),BSTP(JT+1)
C       ------------------------------------------------------------------
C
     1 FORMAT(F10.0,I10,F10.0)
        DUM=1.E22
        STP(1)=1.
C
C     FOR JT=0 (SENSITIVITIES CALCULATED AT STEADY STATE ONLY)
C
        IF(JT.EQ.0) THEN
          READ(INBAS,1,END=100) PERLEN,NSTP,TSMULT
          IF(TSMULT.NE.1.)
     #       TS(1)=PERLEN*(1.-TSMULT)/(1.-(TSMULT**NSTP))
          IF(TSMULT.EQ.1.) TS(1)=TS(1)/NSTP
          NSTP=1
          PERLEN=0.0
          IF(ISN.GT.0) THEN
```

```
          ATS(1)=DUM
          ATS(2)=TS(1)
        ENDIF
        RETURN
      ENDIF
C
C    FOR ALL OTHER JT
C
      JJ=0
      DO 80 K=1,1000
        JJ=JJ+1
        READ(INBAS,1,END=100) PERLEN,NSTP,TSMULT
        STP(1+K)=STP(K)+NSTP
        IF(PERLEN.EQ.0.0) GO TO 100
        IF(TSMULT.NE.1.)
     #     TS(JJ)=PERLEN*(1.-TSMULT)/(1.-(TSMULT**NSTP))
        IF(TSMULT.EQ.1.) TS(JJ)=PERLEN/NSTP
        IF(JJ.EQ.JT) GO TO 110
        DO 90 I=2,NSTP
          JJ=JJ+1
          TS(JJ)=TSMULT*TS(JJ-1)
          IF(JJ.EQ.JT) GO TO 110
   90   CONTINUE
   80 CONTINUE
  100 STOP 'TIME STEP EXCEEDS STRESS PERIOD DATA - FROM SEN1ST'
  110 CONTINUE
C
      NSTP=1
      PERLEN=0.0
      IF(ISN.GT.0) THEN
      DO 121 I=1,JT
        ATS((I*2)-1) = TS(JT-I+1)
121     ATS((I*2)+2)=TS(JT-I+1)
      ATS(2)=DUM
      ATS((JT*2)+1)=DUM
C
C   FOR ADJOINT STATES, SAVE THE LAST PUMPING PERIOD AND THE LAST
C    TIME STEP IN PRIOR PUMPING PERIODS
      BSTP(1)=K
      DO 130 I=1,K
  130   BSTP(I+1)=STP(K-I+1)-1
      ENDIF
      RETURN
      END


C=====================================================================
      SUBROUTINE SEN1RW(IUNIT,NRWD,IRWD)
C-----VERSION 1000 01FEB1992
C     *******************************************************************
C     REWIND DATA FILES WHICH ARE TO BE REREAD
C     *******************************************************************
C        SPECIFICATIONS:
      DIMENSION IUNIT(24),IRWD(NRWD)
C     -------------------------------------------------------------------
    5 FORMAT(I10)
      IF(IUNIT(2).GT.0) THEN
        REWIND(IUNIT(2))
        READ(IUNIT(2),5)
      ENDIF
      IF(IUNIT(3).GT.0) THEN
        REWIND(IUNIT(3))
        READ(IUNIT(3),5)
      ENDIF
      IF(IUNIT(8).GT.0) THEN
        REWIND(IUNIT(8))
        READ(IUNIT(8),5)
      ENDIF
      IF(IUNIT(5).GT.0) THEN
        REWIND(IUNIT(5))
        READ(IUNIT(5),5)
      ENDIF
      IF(IUNIT(4).GT.0) THEN
        REWIND(IUNIT(4))
        READ(IUNIT(4),5)
      ENDIF
      IF(IUNIT(7).GT.0) THEN
        REWIND(IUNIT(7))
        READ(IUNIT(7),5)
      ENDIF
      IF(IUNIT(18).GT.0) THEN
        REWIND(IUNIT(18))
        READ(IUNIT(18),5)
```

```
      ENDIF
      IF(NRWD.GT.0) THEN
        DO 180 N=1,NRWD
  180   REWIND(IRWD(N))
      ENDIF
      REWIND(IUNIT(12))
      READ(IUNIT(12),5)
      RETURN
      END


C=======================================================================
      SUBROUTINE SEN1FM(
     1       KNST,NCOL,NROW,NLAY,KPER,IUHEAD,NPER,SHNW,SHOLD,ISN,IP,HNEW,
     2       NODES,PID,NP,KSTP,PERTIM,TOTIM,BUFF,HOLD,ITERP,ISN1,IOUT)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     PREPARE HEAD ARRAYS REQUIRED TO CALCULATE SENSITIVITIES,
C     INITIALIZE HNEW FOR SENSITIVITIES.
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 PID(NP),TEXT
      INTEGER*4 IUNIT4
      DOUBLE PRECISION HNEW(NCOL,NROW,NLAY)
      DIMENSION SHNW(NCOL,NROW,NLAY),SHOLD(NCOL,NROW,NLAY),
     1   BUFF(NCOL,NROW,NLAY),HOLD(NCOL,NROW,NLAY),TEXT(4,2)
      COMMON /FLWCOM/LAYCON(80)
      DATA TEXT(1,1),TEXT(2,1),TEXT(3,1),TEXT(4,1)
     #  /'    ','    ','    ','HEAD'/
      DATA TEXT(1,2),TEXT(2,2),TEXT(3,2),TEXT(4,2)
     #  /'    ',' SEN','SITI','VITY'/
C     ------------------------------------------------------------------
C
C-----OPEN FILES
      IF(IP.EQ.0.AND.ITERP.EQ.1.AND.KPER.EQ.1) THEN
        NFILES=NPER
        IF(ISN1.EQ.-1) NFILES=NP+1
        IF(ISN1.EQ.10.AND.NP+1.GT.NPER) NFILES=NP+1
        DO 20 I=1,NFILES
          IUNIT4=IUHEAD-1+I
          OPEN(UNIT=IUNIT4,STATUS='SCRATCH',FORM='UNFORMATTED',ERR=30)
   20   CONTINUE
        GO TO 40
   30   WRITE(IOUT,600) NFILES,IUHEAD
  600   FORMAT(' ERROR IN OPENING SCRATCH FILES.  MAKE SURE',I5,
     1      ' FORTRAN UNIT NUMBERS INCLUDING AND AFTER IUHEAD,',
     1      ' WHICH EQUALS',I3,' (LINE 7)',
     2      ' ARE UNSPECIFIED.  (SEN1FM)')
        STOP
   40   CONTINUE
      ENDIF
C-----ONLY NEED TO UPDATE HEAD VECTORS ONCE IF NPER=1
      IF(NPER.EQ.1.AND.IP.GT.1) GO TO 70
C-----USE SS HEADS FROM LAST PAR. EST. ITERATION AS NEXT INITIAL HEADS
C-------READ PREVIOUS STEADY-STATE HEADS
      IF(IP.EQ.0.AND.KPER.EQ.1.AND.
     1 ((ITERP.GT.1.AND.ISN1.NE.10).OR.ITERP.GT.2).AND.ISN.NE.-1) THEN
        REWIND(IUHEAD)
        CALL UREADF(BUFF,NCOL,NROW,NLAY,IUHEAD)
        CALL USAVEF(BUFF,DUM,DUM,HNEW,NODES,2)
      ENDIF
C-----CALCULATING ADJOINT STATES
      IF(ISN.GT.0.AND.IP.EQ.1) THEN
C-------HEADS FROM FINAL TIME STEP WERE PUT IN SHNW IN SEN1OT
        IF(KNST.EQ.1.AND.KPER.EQ.1) GO TO 70
C---------READ HEADS
        IF(KNST.EQ.-1.AND.KPER.NE.NPER) THEN
          IUNIT=IUHEAD+NPER-KPER
          REWIND(IUNIT)
          CALL UREADF(HOLD,NCOL,NROW,NLAY,IUNIT)
        ENDIF
C---------SAVE HEADS FROM HOLD IN SHNW
        IF(KNST.EQ.0.AND.NPER.GT.1)
     1     CALL USAVEF(HOLD,SHNW,DUM,DUM,NODES,1)
      ENDIF
C-----CALCULATING ITERATION-PARAMETER VECTOR
      IF(ISN.GT.0.AND.IP.EQ.2) THEN
C-------READ HEADS
        IUNIT=IUHEAD+KPER-1
        REWIND(IUNIT)
        CALL UREADF(SHNW,NCOL,NROW,NLAY,IUNIT)
      ENDIF
```

```
C-----CALCULATING SENSITIVITY-EQUATION SENSITIVITIES FOR ALL           SEN0599
C-----TIME STEPS BEFORE PROCEEDING TO THE NEXT PARAMETER               SEN0600
      IF(ISN.EQ.-2.AND.IP.GT.0) THEN                                   SEN0601
C-------SAVE HEADS FROM SHNW IN SHOLD                                   SEN0602
        IF(KPER.GT.1) CALL USAVEF(SHNW,SHOLD,DUM,DUM,NODES,1)          SEN0603
C-------READ HEADS                                                      SEN0604
        IUNIT=IUHEAD+KPER-1                                            SEN0605
        REWIND(IUNIT)                                                  SEN0606
        CALL UREADF(SHNW,NCOL,NROW,NLAY,IUNIT)                         SEN0607
      ENDIF                                                            SEN0608
C-----INITIALIZE HNEW FOR SENSITIVITY-EQUATION SENSITIVITIES (ISN=-2),  SEN0609
C-----ADJOINT STATES, AND ITERATION-PARAMETER VECTOR (ISN>0)            SEN0610
   70 CONTINUE                                                         SEN0611
      IF(ISN.NE.-1.AND.KNST.EQ.1.AND.KPER.EQ.1.AND.IP.GT.0) THEN       SEN0612
         DO 105 IL=1,NLAY                                              SEN0613
         DO 105 IR=1,NROW                                              SEN0614
         DO 105 IC=1,NCOL                                              SEN0615
  105    HNEW(IC,IR,IL)=0.D0                                           SEN0616
      ENDIF                                                            SEN0617
C-----CALCULATING SENSITIVITY-EQUATION SENSITIVITIES FOR ALL           SEN0618
C-----PARAMETERS BEFORE PROCEEDING TO NEXT TIME STEP                   SEN0619
      IF(ISN.EQ.-1) THEN                                               SEN0620
C-------HEADS                                                           SEN0621
        IF(IP.EQ.0) THEN                                               SEN0622
          IF(KPER.GT.1) CALL USAVEF(SHNW,SHOLD,DUM,DUM,NODES,1)        SEN0623
          CALL USAVEF(DUM,SHNW,HNEW,DUM,NODES,3)                       SEN0624
          IF(KPER.EQ.1) THEN                                           SEN0625
            DO 220 K=1,NLAY                                            SEN0626
              CALL ULASAV(SHNW(1,1,K),TEXT(1,1),KSTP,KPER,PERTIM,      SEN0627
     1                    TOTIM,NCOL,NROW,K,IUHEAD)                    SEN0628
  220       CONTINUE                                                   SEN0629
            REWIND(IUHEAD)                                             SEN0630
          ENDIF                                                        SEN0631
        ENDIF                                                          SEN0632
C-------SENSITIVITIES                                                   SEN0633
        IF((KPER.EQ.1.OR.IFLAG.EQ.1).AND.IP.LT.NP) THEN                SEN0634
          DO 230 IL=1,NLAY                                             SEN0635
          DO 230 IR=1,NROW                                             SEN0636
          DO 230 IC=1,NCOL                                             SEN0637
  230     HOLD(IC,IR,IL)=0.                                            SEN0638
        ENDIF                                                          SEN0639
        IUNIT=IUHEAD+IP                                                SEN0640
        IF(IP.GE.1.AND.NPER.NE.1)THEN                                  SEN0641
          CALL USAVEF(DUM,BUFF,HNEW,DUM,NODES,3)                       SEN0642
          DO 250 K=1,NLAY                                              SEN0643
            CALL ULASAV(BUFF(1,1,K),TEXT(1,2),KSTP,KPER,PERTIM,        SEN0644
     1        TOTIM,NCOL,NROW,K,IUNIT)                                 SEN0645
  250     CONTINUE                                                     SEN0646
          REWIND(IUNIT)                                                SEN0647
        ENDIF                                                          SEN0648
        IF(KPER.GT.1.AND.IP.LT.NP) THEN                                SEN0649
          CALL UREADF(HOLD,NCOL,NROW,NLAY,IUNIT+1)                     SEN0650
          REWIND(IUNIT+1)                                              SEN0651
        ENDIF                                                          SEN0652
        IF(IP.NE.NP) CALL USAVEF(HOLD,DUM,DUM,HNEW,NODES,2)            SEN0653
C------HEADS                                                            SEN0654
        IF(IP.EQ.NP) THEN                                              SEN0655
          IF(KPER.EQ.NPER) THEN                                        SEN0656
            CALL UREADF(SHNW,NCOL,NROW,NLAY,IUHEAD)                    SEN0657
            REWIND(IUHEAD)                                             SEN0658
          ENDIF                                                        SEN0659
          CALL USAVEF(SHNW,DUM,DUM,HNEW,NODES,2)                       SEN0660
        ENDIF                                                          SEN0661
      ENDIF                                                            SEN0662
C                                                                      SEN0663
      RETURN                                                           SEN0664
      END                                                              SEN0665


C======================================================================SEN0666
      SUBROUTINE SEN1RR (                                              SEN0667
     1       IUNIT,WELL,NWELLS,MXWELL,DRAI,NDRAIN,MXDRN,NRCHOP,IRCH,   SEN0668
     2       RECH,DELR,DELC,NROW,NCOL,NLAY,NEVTOP,IEVT,EVTR,EXDP,SURF, SEN0669
     3       RIVR,NRIVER,MXRIVR,BNDS,NBOUND,MXBND,IOUT,IP,ITERP,STRM,  SEN0670
     4       ISTRM,NSTREM,MXSTRM,ITRBAR,NDIV,NSS,NTRIB,IDIVAR,ICALC,NRP)SEN0671
C-----VERSION 1000 01FEB1992 SEN1RR                                    SEN0672
C     ****************************************************************SEN0673
C     READ THE INPUT FILES AFTER THE FIRST READING, WHICH IS DONE BY THESEN0674
C     ORIGINAL MODULE SUBROUTINES                                      SEN0675
C     ****************************************************************SEN0676
C     SPECIFICATIONS:                                                  SEN0677
C     ----------------------------------------------------------------SEN0678
      DIMENSION IUNIT(24),WELL(4,MXWELL),DRAI(5,MXDRN),IEVT(NCOL,NROW), SEN0679
```

```
     1           EVTR(NCOL,NROW),EXDP(NCOL,NROW),SURF(NCOL,NROW),
     1           DELR(NCOL),DELC(NROW),BNDS(5,MXBND),RIVR(6,MXRIVR),
     1           IRCH(NCOL,NROW),RECH(NCOL,NROW),STRM(11,MXSTRM),
     1           ISTRM(5,MXSTRM),ITRBAR(NSS,NTRIB),IDIVAR(NSS)
C     -----------------------------------------------------------------
C
  501 FORMAT(3I10,3F10.0)
  510 FORMAT(8I10)
  520 FORMAT(5I5,F15.0,4F10.0)
  530 FORMAT(3F10.0)
  540 FORMAT(10I5)
C-------REPEAT READ NRP TIMES (FOR ADJOINT STATES NEED RECORDS FOR
C-------LATTER TIME STEPS FIRST)
      DO 400 N=1,NRP
C-------WEL
      IF(IUNIT(2).GT.0) THEN
        IN=IUNIT(2)
        READ(IN,501) ITMP
        IF(ITMP.GE.0) NWELLS=ITMP
        IF(ITMP.GT.0) THEN
          DO 100 II=1,NWELLS
            READ(IN,501) K,I,J,WELL(4,II)
            WELL(1,II)=K
            WELL(2,II)=I
            WELL(3,II)=J
  100     CONTINUE
        ENDIF
      ENDIF
C-------DRN
      IF(IUNIT(3).GT.0) THEN
        IN=IUNIT(3)
        READ(IN,501) ITMP
        IF(ITMP.GE.0) NDRAIN=ITMP
        IF(ITMP.GT.0) THEN
          DO 110 II=1,NDRAIN
            READ(IN,501) K,I,J,DRAI(4,II),DRAI(5,II)
            DRAI(1,II)=K
            DRAI(2,II)=I
            DRAI(3,II)=J
  110     CONTINUE
        ENDIF
      ENDIF
C-------RCH
      IF(IUNIT(8).GT.0) THEN
        IN=IUNIT(8)
        READ(IN,501) INRECH,INIRCH
        IF(INRECH.GE.0) THEN
          CALL U2DREN(RECH,NROW,NCOL,IN,IOUT)
          DO 125 IR=1,NROW
          DO 125 IC=1,NCOL
            RECH(IC,IR)=RECH(IC,IR)*DELR(IC)*DELC(IR)
  125     CONTINUE
        ENDIF
        IF(NRCHOP.EQ.2.AND.INIRCH.GE.0)CALL U2DINN(IRCH,NROW,NCOL,IN)
      ENDIF
C-------EVT
      IF(IUNIT(5).GT.0) THEN
        IN=IUNIT(5)
        READ(IN,510) INSURF,INEVTR,INEXDP,INIEVT
        IF(INSURF.GE.0) CALL U2DREN(SURF,NROW,NCOL,IN,IOUT)
        IF(INEVTR.GE.0) THEN
          CALL U2DREN(EVTR,NROW,NCOL,IN,IOUT)
          DO 135 IR=1,NROW
          DO 135 IC=1,NCOL
            EVTR(IC,IR)=EVTR(IC,IR)*DELR(IC)*DELC(IR)
  135     CONTINUE
        ENDIF
        IF(INEXDP.GE.0) CALL U2DREN(EXDP,NROW,NCOL,IN,IOUT)
        IF(NEVTOP.EQ.2.AND.INIEVT.GE.0)CALL U2DINN(IEVT,NROW,NCOL,IN)
      ENDIF
C-------RIV
      IF(IUNIT(4).GT.0) THEN
        IN=IUNIT(4)
        READ(IN,501) ITMP
        IF(ITMP.GE.0) NRIVER=ITMP
        IF(ITMP.GT.0) THEN
          DO 150 II=1,NRIVER
            READ(IN,501) K,I,J,(RIVR(JJ,II),JJ=4,6)
            RIVR(1,II)=K
            RIVR(2,II)=I
  150       RIVR(3,II)=J
        ENDIF
      ENDIF
```

```
C-------GHB
      IF(IUNIT(7).GT.0) THEN
        IN=IUNIT(7)
        READ(IN,501) ITMP
        IF(ITMP.GE.0) NBOUND=ITMP
        IF(ITMP.GT.0) THEN
          DO 170 II=1,NBOUND
            READ(IN,501) K,I,J,(BNDS(JJ,II),JJ=4,5)
            BNDS(1,II)=K
            BNDS(2,II)=I
  170       BNDS(3,II)=J
        ENDIF
      ENDIF
C-------STR
      IF(IUNIT(18).NE.0) THEN
        IN=IUNIT(18)
        READ(IN,501) ITMP
        IF(ITMP.GE.0) NSTREM=ITMP
        IF(ITMP.GT.0) THEN
          DO 190 II=1,NSTREM
  190     READ(IN,520) (ISTRM(JJ,II),JJ=1,5),(STRM(JJ,II),JJ=1,5)
          IF(ICALC.GT.0) THEN
            DO 200 II=1,NSTREM
  200       READ(IN,530) (STRM(JJ,II),JJ=6,8)
          ENDIF
          DO 210 IK=1,NSS
            IDIVAR(IK)=0
            DO 210 JK=1,NTRIB
  210     ITRBAR(IK,JK)=0
          IF(NTRIB.GT.0) THEN
            DO 220 IK=1,NSS
  220       READ(IN,540) (ITRBAR(IK,JK),JK=1,NTRIB)
          ENDIF
          IF(NDIV.GT.0) THEN
            DO 230 IK=1,NSS
  230       READ(IN,540) IDIVAR(IK)
          ENDIF
          DO 240 II=1,NSTREM
          DO 240 JJ=9,11
  240     STRM(JJ,II)=0.
        ENDIF
      ENDIF
C
  400 CONTINUE
      RETURN
      END


C=======================================================================
      SUBROUTINE SEN1FN(
     1        CC,CR,CV,CELS,SMAT,NSN,NSM,SC1,ST,HNEW,BNDS,NBOUND,WELL,
     2        NWELLS,RIVR,NRIVER,RECH,DELR,DELC,NROW,NCOL,NLAY,NP,NLL,
     3        IBOUND,IP,B,KPER,NPER,MXBND,MXRIVR,MXWELL,TRPY,PID,ISN,
     4        ITERP,B1,IOUT,LN,HY,EVTR,DRAI,NDRAIN,MXDRN,STRM,MXSTRM,
     5        NSTREM,ISTRM,IUBD,NMM,NZM,LZI1,SFAC,LZ,LM,MATZ,NLLI1,
     6        NLLIT,SV,WT,ND,IPRNP,BUFF,IWPG,NP1,NPNG,IPNG,IOUB,NDER,
     7        COFF,ROFF,RINT,JOFF,IOFF,MLAY,NH,MAXM,MOBS,IDRY,JDRY)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     PUT NEW PARAMETERS IN MODEL ARRAYS.
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 PID(NP1),PIDTMP,ANAME(6,6),PID1
      DOUBLE PRECISION HNEW(NCOL,NROW,NLAY)
      DIMENSION SMAT(NCOL,NROW,NMM),NLL(NLLI1,NP1),DELR(NCOL),DELC(NROW)
     1           ,IBOUND(NCOL,NROW,NLAY),TRPY(NLAY),RECH(NCOL,NROW),
     2           BNDS(5,MXBND),WELL(4,MXWELL),RIVR(6,MXRIVR),LM(NSM),
     3           CELS(4,NSN),SC1(NCOL,NROW,NLAY),MATZ(NCOL,NROW,NZM),
     4           CC(NCOL,NROW,NLAY),CR(NCOL,NROW,NLAY),LN(NP1),SFAC(NSM)
     5           ,CV(NCOL,NROW,NLAY),ST(NCOL,NROW,NLAY),B(NP1),B1(NP1),
     6           HY(NCOL,NROW,NLAY),EVTR(NCOL,NROW),DRAI(5,MXDRN),
     7           STRM(11,MXSTRM),ISTRM(5,MXSTRM),LZ(LZI1,NSM),WT(ND),
     8           SV(NCOL,NROW),BUFF(NCOL,NROW),IWPG(NP1),IPNG(NPNG),
     9           COFF(NH),ROFF(NH),RINT(4,NH),JOFF(NH),IOFF(NH),
     X           MLAY(MAXM,MOBS),NDER(5,NH)
      COMMON /FLWCOM/LAYCON(80)
C
      DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
     1 ANAME(6,1) /'   H','YD. ','COND','. AL','ONG ','ROWS'/
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
     1 ANAME(6,2) /'    ','TRAN','SMIS','. AL','ONG ','ROWS'/
      DATA ANAME(1,3),ANAME(2,3),ANAME(3,3),ANAME(4,3),ANAME(5,3),
```

```
     1  ANAME(6,3) /' EVA','POTR','ANSP','IRAT','ION ','RATE'/
        DATA ANAME(1,4),ANAME(2,4),ANAME(3,4),ANAME(4,4),ANAME(5,4),
     1  ANAME(6,4) /'    ','   V','ERTI','CAL ','LEAK','ANCE'/
        DATA ANAME(1,5),ANAME(2,5),ANAME(3,5),ANAME(4,5),ANAME(5,5),
     1  ANAME(6,5) /'    ','PRIM','ARY ','STOR','AGE ','COEF'/
        DATA ANAME(1,6),ANAME(2,6),ANAME(3,6),ANAME(4,6),ANAME(5,6),
     1  ANAME(6,6) /'    ','    ','    ','    ','RECH','ARGE'/

C     ------------------------------------------------------------------
  500 FORMAT(/,'  PARAMETER #',I3,', ID ',A4,' < 0 : NOT ',
     1    'PHYSICALLY REASONABLE.  CHANGED TO ',G13.6,' (SEN1FN)')
  510 FORMAT(/,'  LN PARAMETER #',I3,', ID ',A4,' <= 0 : NOT ',
     1    'PHYSICALLY OR MATHEMATICALLY REASONABLE.  CHANGED TO ',G13.6,
     2    ' (SEN1FN)')
  515 FORMAT(/,' MODEL-INPUT ARRAYS CALCULATED WITH INITIAL ',
     1 'PARAMETER VALUES (DATA SET 8)')
  520 FORMAT(1H0,/30X,6A4,' LAYER',I3,/,30X,73('-'))
  530 FORMAT(/,10X,6A4,' FOR TIME STEP',I5,/,10X,43('-'))
  540 FORMAT(8G10.3)
C
C-------CHECK FOR PARAMETER VALUES <=0 THAT SHOULD BE >0
      IF(IP.EQ.0.AND.KPER.EQ.1.AND.ITERP.GT.1) THEN
        DO 20 IIP=1,NP
          PIDTMP=PID(IIP)
          IF(B(IIP).LT.0..AND.LN(IIP).EQ.0.AND.
     1      (PIDTMP.EQ.'T   '.OR.PIDTMP.EQ.'S1  '.OR.
     2       PIDTMP.EQ.'KV  '.OR.PIDTMP.EQ.'TKV '.OR.
     3       PIDTMP.EQ.'GHB '.OR.PIDTMP.EQ.'KRB '.OR.
     4       PIDTMP.EQ.'KST '.OR.PIDTMP.EQ.'KDR '.OR.
     5       PIDTMP.EQ.'ANI '.OR.PIDTMP.EQ.'ANIV'.OR.
     6       PIDTMP.EQ.'ETM ')) THEN
            IF(NPNG.GT.0) THEN
              DO 10 I=1,NPNG
   10         IF(IIP.EQ.IPNG(I)) GO TO 20
            ENDIF
            B(IIP)=B1(IIP)/100.
            WRITE(IOUT,500) IIP,PIDTMP,B(IIP)
          ENDIF
          IF(B(IIP).LT.1.E-14.AND.LN(IIP).GT.0) THEN
            B(IIP)=1.E-14
            WRITE(IOUT,510) IIP,PIDTMP,B(IIP)
          ENDIF
   20   CONTINUE
C-------WRITE PARAMETER VALUES TO IOUB
        WRITE(IOUB,540) (B(IIP),IIP=1,NP)
      ENDIF
C-------SET FLAG TO INDICATE UNCONFINED LAYERS
      KK=0
      DO 30 K=1,NLAY
   30 KK=KK+LAYCON(K)
C-------READ IBOUND ARRAYS
      IF(IUBD.GT.0.AND.(IP.EQ.0.OR.ISN.NE.-1).AND.
     1  (KPER.EQ.1.OR.(KPER.EQ.2.AND.KK.EQ.0))) THEN
        IF(KPER.EQ.1) REWIND(IUBD)
        DO 40 K=1,NLAY
   40   CALL U2DINN(IBOUND(1,1,K),NROW,NCOL,IUBD)
      ENDIF
C-------RECALCULATE INTERPOLATION COEFFICIENTS, IF NEEDED
      IF(IP.EQ.0.AND.ISN.EQ.-1.AND.KPER.EQ.1.AND.ITERP.GT.1) THEN
C---------SKIP IF ALL LAYERS ARE CONFINED
        IF(KK.EQ.0) GO TO 60
C---------HAVE ANY DRY CELLS AFFECTED THE INTERPOLATION?
        IF(IDRY+JDRY.GT.0) THEN
          ML=0
          DO 50 N=1,NH
            IF(NDER(1,N).LT.0) ML=ML+1
            IF(COFF(N).GT..5.OR.WT(N).LT.0.) THEN
              IF(COFF(N).GT..5) COFF(N)=COFF(N)-5.
              IF(WT(N).LT.0.) WT(N)=ABS(WT(N))
              CALL SSEN1I (NDER(1,N),COFF(N),ROFF(N),DELR,DELC,IBOUND,
     1          NCOL,NROW,NLAY,RINT(1,N),JOFF(N),IOFF(N),MLAY(1,ML))
            ENDIF
   50     CONTINUE
        ENDIF
      ENDIF
   60 CONTINUE
C-------INITIALIZE MODEL INPUTS TO ZERO
      IF(IP.EQ.0.AND.KPER.EQ.1) THEN
        DO 80 IIP=1,NP1
          IF(PID(IIP).EQ.'T   '.OR.PID(IIP).EQ.'TKV ') THEN
            DO 70 K=1,NLAY
            DO 70 I=1,NROW
            DO 70 J=1,NCOL
```

```
   70        ST(J,I,K)=0.
             GO TO 90
           ENDIF
   80    CONTINUE
   90    DO 110 IIP=1,NP1
           IF(PID(IIP).EQ.'KV  '.OR.PID(IIP).EQ.'TKV ') THEN
             DO 100 K=1,NLAY-1
             DO 100 I=1,NROW
             DO 100 J=1,NCOL
  100        CV(J,I,K)=0.
             GO TO 130
           ENDIF
  110    CONTINUE
C-------IF NO PID=KV OR TKV, REINITIALIZE CV BENEATH A WATER TABLE LAYER
         IF(ITERP.GT.1.AND.KPER.EQ.1.AND.NLAY.GT.1.AND.
     1       LAYCON(1).EQ.1) THEN
           DO 120 I=1,NROW
           DO 120 J=1,NCOL
             CV(J,I,1)=SV(J,I)
             IF(IBOUND(J,I,1).EQ.0) CV(J,I,1)=0.
  120      CONTINUE
         ENDIF
  130    DO 150 IIP=1,NP1
           IF(PID(IIP).EQ.'S1  ') THEN
             DO 140 K=1,NLAY
             DO 140 I=1,NROW
             DO 140 J=1,NCOL
  140        SC1(J,I,K)=0.
             GO TO 160
           ENDIF
  150    CONTINUE
       ENDIF
  160 DO 180 IIP=1,NP1
         IF(PID(IIP).EQ.'RCH ') THEN
           DO 170 I=1,NROW
           DO 170 J=1,NCOL
  170      RECH(J,I)=0.
           GO TO 190
         ENDIF
  180 CONTINUE
  190 DO 210 IIP=1,NP1
         IF(PID(IIP).EQ.'ETM ') THEN
           DO 200 I=1,NROW
           DO 200 J=1,NCOL
  200      EVTR(J,I)=0.
           GO TO 220
         ENDIF
  210 CONTINUE
  220 CONTINUE
C-------PUT NEW PARAMETERS IN MODEL ARRAYS.
C----------SET COUNTERS
      NS=0
      NM=0
      ICH=0
C----------LOOP THROUGH PARAMETERS
      DO 350 IIP=1,NP1
        PIDTMP=PID(IIP)
        BB=B(IIP)
C----------PARAMETERS DEFINED BY MATRICES
        IF(PIDTMP.EQ.'T   '.OR.PIDTMP.EQ.'TKV ') THEN
          IF(PIDTMP.EQ.'TKV '.AND.KPER.GT.1) GO TO 320
          NM1=NM
          DO 310 LL=1,NLLI1
            NLO=NLL(LL,IIP)
            IF(NLO.EQ.0) THEN
              IF(PIDTMP.EQ.'TKV ') GO TO 310
              GO TO 320
            ENDIF
            IF(NLO.GT.0) NM=NM+1
            IF(IP.EQ.0.AND.KPER.EQ.1) THEN
              IF(PIDTMP.EQ.'TKV '.AND.MOD(LL+1,3).NE.0) GO TO 310
              NL=IABS(NLO)
              IF(NLO.GT.0) THEN
                SF=SFAC(NM)*BB
                M=LM(NM)
                LZ1=LZ(1,NM)
                IF(M.EQ.0.AND.LZ1.EQ.0) THEN
                  DO 240 I=1,NROW
                  DO 240 J=1,NCOL
  240             ST(J,I,NL)=ST(J,I,NL)+SF
                ENDIF
                IF (M.NE.0.AND.LZ1.EQ.0) THEN
                  DO 250 I=1,NROW
```

```
                DO 250 J=1,NCOL
  250           IF(SMAT(J,I,M).NE.0.) ST(J,I,NL)=ST(J,I,NL)+
     1                                   SF*SMAT(J,I,M)
              ENDIF
              IF (M.EQ.0.AND.LZ1.NE.0) THEN
                DO 260 IZ=2,LZI1
                  NZ=LZ(IZ,NM)
                  IF(NZ.EQ.0) GO TO 270
                  DO 260 I=1,NROW
                  DO 260 J=1,NCOL
  260             IF(MATZ(J,I,LZ1).EQ.NZ) ST(J,I,NL)=ST(J,I,NL)+SF
  270           CONTINUE
              ENDIF
              IF (M.NE.0.AND.LZ1.NE.0) THEN
                DO 280 IZ=2,LZI1
                  NZ=LZ(IZ,NM)
                  IF(NZ.EQ.0) GO TO 290
                  DO 280 I=1,NROW
                  DO 280 J=1,NCOL
  280             IF(MATZ(J,I,LZ1).EQ.NZ.AND.SMAT(J,I,M).NE.0.)
     1                ST(J,I,NL)=ST(J,I,NL)+SF*SMAT(J,I,M)
  290           CONTINUE
              ENDIF
            ELSEIF(NLO.LT.0) THEN
              DO 300 I=1,NROW
                DO 300 J=1,NCOL
  300           ST(J,I,NL)=ST(J,I,NL)+BB
            ENDIF
          ENDIF
  310   CONTINUE
        IF(PIDTMP.EQ.'TKV ') NM=NM1
  320 CONTINUE
      ENDIF
      IF(PIDTMP.EQ.'KV  '.OR.PIDTMP.EQ.'TKV ')
     1  CALL SSEN1M(PIDTMP,CV,SMAT,NCOL,NROW,NLAY,NSM,NM,
     2        NLL(1,IIP),BB,KPER,IP,DELR,DELC,IBOUND,ISN,
     3        NMM,NZM,LZI1,SFAC,LZ,LM,MATZ,NLLI1,NLLIT)
      IF(PIDTMP.EQ.'S1  ')
     1  CALL SSEN1M(PIDTMP,SC1,SMAT,NCOL,NROW,NLAY,NSM,NM,
     2        NLL(1,IIP),BB,KPER,IP,DELR,DELC,IBOUND,ISN,
     3        NMM,NZM,LZI1,SFAC,LZ,LM,MATZ,NLLI1,NLLIT)
      IF(PIDTMP.EQ.'RCH ')
     1  CALL SSEN1M(PIDTMP,RECH,SMAT,NCOL,NROW,NLAY,NSM,NM,
     2       NLL(1,IIP),BB,KPER,IP,DELR,DELC,IBOUND,ISN,
     3        NMM,NZM,LZI1,SFAC,LZ,LM,MATZ,NLLI1,NLLIT)
      IF(PIDTMP.EQ.'ETM ')
     1  CALL SSEN1M(PIDTMP,EVTR,SMAT,NCOL,NROW,NLAY,NSM,NM,
     2       NLL(1,IIP),BB,KPER,IP,DELR,DELC,IBOUND,ISN,
     3        NMM,NZM,LZI1,SFAC,LZ,LM,MATZ,NLLI1,NLLIT)
C----------ANI PARAMETERS
      IF(PIDTMP.EQ.'ANI '.AND.KPER.EQ.1) THEN
        DO 330 LL=1,NLLI1
  330   IF(NLL(LL,IIP).GT.0) TRPY(NLL(LL,IIP))=BB
      ENDIF
C----------PARAMETERS DEFINED BY LISTS OF CELLS
      IF(PIDTMP.EQ.'CH  ')
     1  CALL SSEN1J(IIP,B,NLL(1,IIP),CELS,NS,NSN,HNEW,NCOL,NROW,NLAY,
     1         NP1,IP,KPER,ICH,NLLI1)
      IF(PIDTMP.EQ.'GHB ')
     1  CALL SSEN1K(PIDTMP,IIP,NLL(1,IIP),NBOUND,MXBND,BNDS,CELS,
     1     NSN,NS,NCOL,NROW,NLAY,5,IBOUND,NPER,KPER,BB,IP,ISN,
     1     ISTRM,NLLI1,NLLIT)
      IF(PIDTMP.EQ.'Q   ')
     1  CALL SSEN1K(PIDTMP,IIP,NLL(1,IIP),NWELLS,MXWELL,WELL,CELS,
     1     NSN,NS,NCOL,NROW,NLAY,4,IBOUND,NPER,KPER,BB,IP,ISN,
     1     ISTRM,NLLI1,NLLIT)
      IF(PIDTMP.EQ.'KRB ')
     1  CALL SSEN1K(PIDTMP,IIP,NLL(1,IIP),NRIVER,MXRIVR,RIVR,CELS,
     1     NSN,NS,NCOL,NROW,NLAY,6,IBOUND,NPER,KPER,BB,IP,ISN,
     1     ISTRM,NLLI1,NLLIT)
      IF(PIDTMP.EQ.'KDR ')
     1  CALL SSEN1K(PIDTMP,IIP,NLL(1,IIP),NDRAIN,MXDRN,DRAI,CELS,
     1     NSN,NS,NCOL,NROW,NLAY,5,IBOUND,NPER,KPER,BB,IP,ISN,
     1     ISTRM,NLLI1,NLLIT)
      IF(PIDTMP.EQ.'KST ')
     1  CALL SSEN1K(PIDTMP,IIP,NLL(1,IIP),NSTREM,MXSTRM,STRM,CELS,
     1     NSN,NS,NCOL,NROW,NLAY,11,IBOUND,NPER,KPER,BB,IP,ISN,
     1     ISTRM,NLLI1,NLLIT)
  350 CONTINUE
C-------ANIV PARAMETERS
     IF(KPER.EQ.1.AND.IP.EQ.0) THEN
       DO 380 IIP=1,NP
         IF(PID(IIP).EQ.'ANIV') THEN
```

```
          BB=B(IIP)
          DO 370 LL=1,NLLI1
            IF(NLL(LL,IIP).GT.0) THEN
              NL=NLL(LL,IIP)
              DO 360 I=1,NROW
              DO 360 J=1,NCOL
  360         CV(J,I,NL)=BB*CV(J,I,NL)
            ENDIF
  370     CONTINUE
        ENDIF
  380   CONTINUE
      ENDIF
C-------SET CV TO ZERO WHERE NOT USED
      DO 400 IIP=1,NP1
        PIDTMP=PID(IIP)
        IF(PIDTMP.EQ.'KV  '.OR.PIDTMP.EQ.'TKV ') THEN
          DO 390 NL=1,NLAY-1
          DO 390 I=1,NROW
          DO 390 J=1,NCOL
            IF(IBOUND(J,I,NL).EQ.0.OR.IBOUND(J,I,NL+1).EQ.0)
     1                CV(J,I,NL)=0.
  390     CONTINUE
          GO TO 410
        ENDIF
  400 CONTINUE
  410 CONTINUE
C--------PRINT MODEL INPUT ARRAYS WHICH ARE NROW X NCOL
      IF(ITERP.EQ.1.AND.IPRNP.GT.0.AND.(ISN.NE.-2.OR.IP.EQ.0)) THEN
        ITS=KPER-1
        IFLAG=1
        IF(ITS.EQ.0) IFLAG=0
        IF(ISN.GT.0.AND.IP.EQ.1) ITS=NPER-KPER
        IF(KPER.EQ.1) THEN
          DO 470 JJ=1,3
            IF(JJ.EQ.1) THEN
              PIDTMP='S1  '
              INAME=5
            ELSEIF(JJ.EQ.3) THEN
              PIDTMP='KV  '
              INAME=4
            ENDIF
            IF(JJ.EQ.2) PIDTMP='T
            DO 460 NL=1,NLAY
              DO 450 IIP=1,NP1
                PID1=PID(IIP)
                DO 440 L=1,NLLI1
                  ICOR=0
                  IF(PID1.EQ.'TKV '.AND.MOD(L+2,3).EQ.0) ICOR=1
                  IF((PID1.EQ.PIDTMP.OR.(PID1.EQ.'TKV '.AND.
     1                  ((JJ.EQ.2.AND.MOD(L+1,3).EQ.0).OR.
     1                  (JJ.EQ.3.AND.MOD(L+1,3).NE.0))))
     1                    .AND.NLL(L,IIP)-ICOR.EQ.NL) THEN
                    IF(JJ.EQ.2) THEN
                      INAME=2
                      IF(NL.EQ.1.AND.LAYCON(NL).EQ.1) INAME=1
                    ENDIF
                    IF(IFLAG.EQ.0) WRITE(IOUT,515)
                    IFLAG=1
                    WRITE(IOUT,520) (ANAME(I,INAME),I=1,6),NL-ICOR
                    IF(JJ.EQ.2) THEN
                      CALL ULAPRW(ST(1,1,NL),ANAME(1,INAME),0,
     1                  0,NCOL,NROW,0,IPRNP,IOUT)
                      GO TO 460
                    ELSEIF(JJ.EQ.1) THEN
                      DO 420 I=1,NROW
                      DO 420 J=1,NCOL
  420                 BUFF(J,I)=SC1(J,I,NL)/(DELR(J)*DELC(I))
                    ELSEIF(JJ.EQ.3) THEN
                      DO 430 I=1,NROW
                      DO 430 J=1,NCOL
  430                   BUFF(J,I)=CV(J,I,NL-ICOR)/(DELR(J)*DELC(I))
                    ENDIF
                    CALL ULAPRW(BUFF,ANAME(1,INAME),0,0,
     1                             NCOL,NROW,0,IPRNP,IOUT)
                    GO TO 460
                  ENDIF
  440           CONTINUE
  450         CONTINUE
  460       CONTINUE
  470     CONTINUE
        ENDIF
C----------RCH AND ETM
        DO 720 JJ=1,2
```

```
          PIDTMP='ETM '
          IF(JJ.EQ.1) PIDTMP='RCH '
          DO 710 IIP=1,NP1
            IF(PID(IIP).EQ.PIDTMP) THEN
              DO 700 LL=1,NLLIT
                IF((ISN.LT.0.OR.IP.NE.1).AND.NLL(2*LL,IIP).EQ.ITS
     1            .AND.NLL(1,IIP).GT.0) THEN
                  INAME=3
                  IF(JJ.EQ.1) INAME=6
                  WRITE(IOUT,530) (ANAME(I,INAME),I=1,6),NLL(2*LL,IIP)
                  IF(JJ.EQ.1) THEN
                     DO 480 I=1,NROW
                     DO 480 J=1,NCOL
  480                BUFF(J,I)=RECH(J,I)/(DELR(J)*DELC(I))
                  ELSE
                     DO 490 I=1,NROW
                     DO 490 J=1,NCOL
  490                BUFF(J,I)=EVTR(J,I)/(DELR(J)*DELC(I))
                  ENDIF
                  CALL ULAPRW(BUFF,ANAME(1,INAME),0,0,
     1                        NCOL,NROW,0,IPRNP,IOUT)
                  GO TO 720
                ENDIF
  700         CONTINUE
            ENDIF
  710     CONTINUE
  720   CONTINUE
      ENDIF
C   UPDATE CC AND CR
      IF(IP.EQ.0.AND.KPER.EQ.1) THEN
        KB=0
        DO 750 K=1,NLAY
          IF(LAYCON(K).EQ.0) THEN
            YX=TRPY(K)*2.
            DO 730 I=1,NROW
            DO 730 J=1,NCOL
            T1=ST(J,I,K)
            CR(J,I,K)=0.
            CC(J,I,K)=0.
            IF(IBOUND(J,I,K).NE.0.AND.T1.NE.0.) THEN
              IF(J.NE.NCOL.AND.IBOUND(J+1,I,K).NE.0) THEN
                T2=ST(J+1,I,K)
                IF(T2.NE.0.)
     1            CR(J,I,K)=2.*T2*T1*DELC(I)/(T1*DELR(J+1)+T2*DELR(J))
              ENDIF
              IF(I.NE.NROW.AND.IBOUND(J,I+1,K).NE.0) THEN
                T2=ST(J,I+1,K)
                IF(T2.NE.0.)
     1            CC(J,I,K)=YX*T2*T1*DELR(J)/(T1*DELC(I+1)+T2*DELC(I))
              ENDIF
            ENDIF
  730     CONTINUE
        ELSEIF (LAYCON(K).EQ.1) THEN
          KB=KB+1
          DO 740 I=1,NROW
          DO 740 J=1,NCOL
  740     HY(J,I,KB)=ST(J,I,K)
        ENDIF
  750   CONTINUE
      ENDIF
C
      RETURN
      END


C=======================================================================
      SUBROUTINE SEN1RH(IUNHEA,BUFF,HNEW,KPER,NROW,NCOL,NLAY,IBOUND)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     CHANGE HEADS AT CONSTANT-HEAD BOUNDARIES
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW(NCOL,NROW,NLAY)
      DIMENSION BUFF(NCOL,NROW,NLAY),IBOUND(NCOL,NROW,NLAY)
C     ------------------------------------------------------------------
  500 FORMAT(10F5.0)
C-----REWIND INPUT FILE
      REWIND(IUNHEA)
C-----READ TIME STEPS AT WHICH HYDRAULIC HEADS AT CONSTANT-HEAD
C-----BOUNDARIES CHANGE
      READ(IUNHEA,500) (BUFF(IN,1,1),IN=1,16)
C-----ARE THERE CHANGES FOR THE PRESENT TIME STEP?
```

```
      DO 100 IN=1,16                                                    SEN1257
        IF(BUFF(IN,1,1).EQ.0.) RETURN                                   SEN1258
        IF(INT(BUFF(IN,1,1)).EQ.KPER) GO TO 105                         SEN1259
  100 CONTINUE                                                          SEN1260
      RETURN                                                            SEN1261
C-----READ ASSOCIATED HYDRAULIC HEADS FROM IUNHEA                        SEN1262
  105 DO 120 IM=1,IN                                                    SEN1263
        DO 110 K=1,NLAY                                                 SEN1264
  110   CALL U2DREN(BUFF(1,1,K),NCOL,NROW,IUNHEA)                       SEN1265
  120 CONTINUE                                                          SEN1266
C-----REPLACE HYDRAULIC HEADS ALONG CONSTANT-HEAD BOUNDARIES             SEN1267
      DO 140 K=1,NLAY                                                   SEN1268
      DO 140 I=1,NROW                                                   SEN1269
      DO 140 J=1,NCOL                                                   SEN1270
  140   IF(IBOUND(J,I,K).LT.0) HNEW(J,I,K)=BUFF(J,I,K)                  SEN1271
      RETURN                                                            SEN1272
      END                                                               SEN1273


C=======================================================================SEN1274
      SUBROUTINE SEN1AR(                                                SEN1275
     1              IOUT,NROW,NCOL,NLAY,NP,ISN,NH,NDER,HNEW,PID,        SEN1276
     2              DID,IP,KPER,BUFF,KSTP,PERTIM,TOTIM,X,DELR,DELC,     SEN1277
     3              IBOUND,COFF,ROFF,H,WT,HOBS,IFO,ITERP,IPAR,          SEN1278
     4              RINT,JOFF,IOFF,MLAY,PR,MOBS,NPER,B,LN,NQ,NQC,       SEN1279
     5              NQT,NQOB,NQCL,IQOB,QCLS,IBT,MXBND,NBOUND,BNDS,      SEN1280
     6              MXRIVR,NRIVER,RIVR,SHNW,LASTX,ISCALS,TOFF,          SEN1281
     7              MXDRN,NDRAIN,DRAI,MXSTRM,NSTREM,STRM,ISTRM,RHS,     SEN1282
     8              MAXM)                                               SEN1283
C-----VERSION 1000 01FEB1992                                            SEN1284
C     ******************************************************************SEN1285
C     COMPUTE OBJ. FUNC. DERIVATIVE AND ADD TO RHS TO CALCULATE ADJOINT SEN1286
C     STATES                                                            SEN1287
C     ******************************************************************SEN1288
C        SPECIFICATIONS:                                                SEN1289
C     ------------------------------------------------------------------SEN1290
      CHARACTER*4 PID(NP),DID(NH+NQT)                                   SEN1291
      DOUBLE PRECISION  HNEW(NCOL,NROW,NLAY)                            SEN1292
      DIMENSION NDER(5,NH),BUFF(NCOL*NROW*NLAY),X(NP,1),                SEN1293
     1     DELR(NCOL),DELC(NROW),IBOUND(NCOL,NROW,NLAY),COFF(NH),       SEN1294
     1     ROFF(NH),H(NH+NQT),WT(NH+NQT),HOBS(NH+NQT),RINT(4,NH),       SEN1295
     1     JOFF(NH),IOFF(NH),MLAY(MAXM,MOBS),PR(MAXM,MOBS),B(NP),LN(NP),SEN1296
     1     RIVR(6,MXRIVR),BNDS(5,MXBND),IBT(2,NQ),NQOB(NQ),NQCL(NQ),    SEN1297
     1     IQOB(NQT),QCLS(4,NQC),TOFF(NH+NQT),DRAI(5,MXDRN),            SEN1298
     1     RHS(NCOL,NROW,NLAY),STRM(11,MXSTRM),ISTRM(5,MXSTRM)          SEN1299
      COMMON /FLWCOM/LAYCON(80)                                         SEN1300
C     ------------------------------------------------------------------SEN1301
C-------HEADS                                                           SEN1302
      IF(NH.GT.0) CALL SSEN1U(NH,NDER,IOFF,JOFF,KPER,MLAY,IBOUND,RINT,  SEN1303
     1    DID,COFF,ROFF,DELR,DELC,NCOL,NROW,NLAY,PR,H,HOBS,WT,HNEW,X,IP,SEN1304
     1    IDRY,NP,MOBS,B,LN,ISN,NQT,TOFF,NPER,RHS,PV,MAXM,              SEN1305
     3    ITERP,JDRY,IPAR,IOUT)                                         SEN1306
      RETURN                                                            SEN1307
      END                                                               SEN1308


C=======================================================================SEN1309
      SUBROUTINE SEN1AP(                                                SEN1310
     1                  PID,KPER,NPER,DELT,IP,NP,NCOL,NROW,NLAY,NODES,  SEN1311
     2                  HOLD,SHNW,HNEW,XX,SMAT,NLL,NM,NSM,DELR,DELC,ST, SEN1312
     3                  IBOUND,TRPY,CELS,NS,NSN,MXWELL,NWELLS,MXBND,    SEN1313
     4                  NBOUND,MXRIVR,NRIVER,BNDS,WELL,RIVR,NRCHOP,IRCH,SEN1314
     5                  ISN,RHS,SHOLD,SC1,CR,CC,CV,IFLCH,IOUT,HY,BOT,   SEN1315
     6                  SC2,KITER,NDRAIN,MXDRN,DRAI,SURF,EXDP,          SEN1316
     7                  NEVTOP,MXSTRM,NSTREM,STRM,ISTRM,DD,B,LN,BUFF,   SEN1317
     8                  H,HOBS,RINT,IOFF,JOFF,NDER,NH,TOFF,SCL,NMM,NZM, SEN1318
     9                  LZI1,SFAC,LZ,LM,MATZ,NLLI1,NLLIT)               SEN1319
C-----VERSION 1000 01FEB1992                                            SEN1320
C     ******************************************************************SEN1321
C     COMPUTE CONTRIBUTIONS TO THE ADJOINT SENSITIVITIES, OR COMPUTE    SEN1322
C        RHS FOR SENSITIVITY-EQUATION SENSITIVITIES OR THE DERIVATIVE   SEN1323
C        OF HYDRAULIC HEAD WITH RESPECT TO NEWTON'S STEP SIZE.          SEN1324
C     ******************************************************************SEN1325
C        SPECIFICATIONS:                                                SEN1326
C     ------------------------------------------------------------------SEN1327
      CHARACTER*4   PID(NP),PIDTMP                                      SEN1328
      DOUBLE PRECISION HNEW(NCOL,NROW,NLAY),DD(NP),XX(NP)               SEN1329
      DIMENSION  HOLD(NCOL,NROW,NLAY),SHNW(NCOL,NROW,NLAY),LM(NSM),     SEN1330
     1          SMAT(NCOL,NROW,NSM),NLL(NLLI1,NP),DELR(NCOL),DELC(NROW),SEN1331
     2          IBOUND(NCOL,NROW,NLAY),TRPY(NLAY),DRAI(5,MXDRN),        SEN1332
     3          BNDS(5,MXBND),WELL(4,MXWELL),RIVR(6,MXRIVR),SFAC(NSM),  SEN1333
     4          CELS(4,NSN),IRCH(NCOL,NROW),RHS(NCOL,NROW,NLAY),        SEN1334
     5          SHOLD(NCOL,NROW,NLAY),SC1(NCOL,NROW,NLAY),LZ(LZI1,NSM), SEN1335
```

```
     6          CC(NCOL,NROW,NLAY),CR(NCOL,NROW,NLAY),ST(NODES),
     7          CV(NCOL,NROW,NLAY),HY(NCOL,NROW,NLAY),B(NP),SCL(NP),
     8          BOT(NCOL,NROW,NLAY),LN(NP),MATZ(NCOL,NROW,NZM),
     9          SC2(NCOL,NROW,NLAY),SURF(NCOL,NROW),EXDP(NCOL,NROW),
     X          STRM(11,MXSTRM),ISTRM(5,MXSTRM),BUFF(NCOL,NROW,NLAY),
     1          H(NH),HOBS(NH),RINT(4,NH),IOFF(NH),JOFF(NH),NDER(5,NH)
      COMMON /FLWCOM/LAYCON(80)
C     ------------------------------------------------------------------
C
C-----INITIALIZE RHS FOR SENSITIVITY-EQUATION SENSITIVITIES OR
C-----ITERATION-PARAMETER VECTOR
      IF(ISN.LT.0.OR.IP.EQ.2) THEN
        DO 100 K=1,NLAY
        DO 100 I=1,NROW
        DO 100 J=1,NCOL
  100   RHS(J,I,K) =0.
      ENDIF
C-----INITIALIZE XX FOR ADJOINT-STATE SENSITIVITIES
      IF(ISN.GT.0.AND.IP.EQ.1.AND.KPER.EQ.1) THEN
        DO 105 IIP=1,NP
  105   XX(IIP)=0.D0
      ENDIF
C-----SET FLAG WHICH DISTINGUISHES BETWEEN CALCULATING ADJOINT-STATE
C-----SENSITIVITIES AND THE ITERATION-PARAMETER VECTOR
      IHESS=0
      IF(ISN.GT.0.AND.IP.EQ.2) IHESS=1
C-----IF CALCULATING ADJOINT-STATE SENSITIVITIES OR ITERATION PARAMETER
C-----VECTOR, ARRANGE TO LOOP THROUGH THE PARAMETERS
      IIP=IP
      NNP=1
      IF(ISN.GT.0) THEN
        NNP=NP
        NS=0
        NM=0
        IFLCH=0
      ENDIF
      DO 110 IPP=1,NNP
        IP=IPP
        IF(ISN.LT.0)IP=IIP
        PIDTMP=PID(IP)
        IF(PIDTMP.EQ.'GHB ') CALL SSEN1D(PIDTMP,NLL(1,IP),NBOUND,MXBND,
     1         BNDS,CELS,NSN,NS,SHNW,HNEW,NCOL,NROW,NLAY,IOUT,5,IBOUND,
     2         NPER,KPER,XX(IP),ISN,RHS,ISTRM,DD(IP),IHESS,B(IP),LN(IP),
     3         SCL(IP),NLLI1,NLLIT)
        IF(PIDTMP.EQ.'KDR ') CALL SSEN1D(PIDTMP,NLL(1,IP),NBOUND,MXDRN,
     1         DRAI,CELS,NSN,NS,SHNW,HNEW,NCOL,NROW,NLAY,IOUT,5,IBOUND,
     2         NPER,KPER,XX(IP),ISN,RHS,ISTRM,DD(IP),IHESS,B(IP),LN(IP),
     3         SCL(IP),NLLI1,NLLIT)
        IF(PIDTMP.EQ.'KRB ') CALL SSEN1D(PIDTMP,NLL(1,IP),NRIVER,MXRIVR,
     1         RIVR,CELS,NSN,NS,SHNW,HNEW,NCOL,NROW,NLAY,IOUT,6,IBOUND,
     2         NPER,KPER,XX(IP),ISN,RHS,ISTRM,DD(IP),IHESS,B(IP),LN(IP),
     3         SCL(IP),NLLI1,NLLIT)
        IF(PIDTMP.EQ.'KST ') CALL SSEN1D(PIDTMP,NLL(1,IP),NSTREM,MXSTRM,
     1         STRM,CELS,NSN,NS,SHNW,HNEW,NCOL,NROW,NLAY,IOUT,11,IBOUND,
     2         NPER,KPER,XX(IP),ISN,RHS,ISTRM,DD(IP),IHESS,B(IP),LN(IP),
     3         SCL(IP),NLLI1,NLLIT)
        IF(PIDTMP.EQ.'Q   ') CALL SSEN1D(PIDTMP,NLL(1,IP),NWELLS,MXWELL,
     1         WELL,CELS,NSN,NS,SHNW,HNEW,NCOL,NROW,NLAY,IOUT,4,IBOUND,
     2         NPER,KPER,XX(IP),ISN,RHS,ISTRM,DD(IP),IHESS,B(IP),LN(IP),
     3         SCL(IP),NLLI1,NLLIT)
        IF(PIDTMP.EQ.'T   '.OR.PIDTMP.EQ.'KV  '.OR.PIDTMP.EQ.'ANI '
     1    .OR.PIDTMP.EQ.'ANIV'.OR.PIDTMP.EQ.'TKV '
     2    .OR.(KPER.GT.1.AND.PIDTMP.EQ.'S1  '))
     3         CALL SSEN1P(SMAT,NLL(1,IP),NSM,NM,SHNW,HOLD,HNEW,NCOL,
     4            NROW,NLAY,KPER,NPER,PIDTMP,ST,DELR,DELC,IBOUND,TRPY,
     5            DELT,XX(IP),ISN,RHS,SHOLD,BOT,DD(IP),IHESS,B(IP),
     6            LN(IP),SCL(IP),NMM,NZM,LZI1,SFAC,LZ,LM,MATZ,NLLI1,CV)
        IF(PIDTMP.EQ.'RCH ') CALL SSEN1R(NLL(1,IP),KPER,NPER,NCOL,
     1       NROW,NLAY,DELR,DELC,SMAT,NSM,NM,IOUT,XX(IP),NRCHOP,IRCH,
     2       HNEW,IBOUND,ISN,RHS,SURF,EXDP,PIDTMP,SHNW,DD(IP),IHESS,
     3       SCL(IP),NMM,NZM,LZI1,SFAC,LZ,LM,MATZ,NLLI1,NLLIT)
        IF(PIDTMP.EQ.'ETM ') CALL SSEN1R(NLL(1,IP),KPER,NPER,NCOL,
     1       NROW,NLAY,DELR,DELC,SMAT,NSM,NM,IOUT,XX(IP),NEVTOP,IEVT,
     2       HNEW,IBOUND,ISN,RHS,SURF,EXDP,PIDTMP,SHNW,DD(IP),IHESS,
     3       SCL(IP),NMM,NZM,LZI1,SFAC,LZ,LM,MATZ,NLLI1,NLLIT)
        IF(PIDTMP.EQ.'CH  ') CALL SSEN1C(IP,NLL(1,IP),NCOL,NROW,NLAY,
     1       CR,CC,CV,IBOUND,XX,NP,CELS,NS,NSN,ISN,HNEW,IFLCH,DD,
     2       IHESS,RHS,H,HOBS,RINT,IOFF,JOFF,NDER,NH,KPER,NPER,TOFF,
     3       SCL(IP),NLLI1)
  110 CONTINUE
      IP=IIP
C-------TERMS FOR UNCONFINED AQUIFERS
      IF(KITER.GT.1) THEN
```

```
        KB=0
        KT=0
        DO 120 K=1,NLAY
          IF(LAYCON(K).EQ.1) THEN
            KB=KB+1
            CALL SSEN1N(SHNW,HOLD,HNEW,NCOL,NROW,NLAY,ST,DELR,DELC,
     1            IBOUND,TRPY,RHS,SHOLD,BOT,CR,CC,K,ISN,KB,XX,NP)
          ENDIF
  120   CONTINUE
      ENDIF
C-------B MATRIX TIMES SOLUTION FROM LAST TIME STEP FOR SENSITIVITY-
C-------EQUATION SENSITIVITIES AND ITERATION-PARAMETER VECTOR
      IF(ISN.LT.0.OR.IHESS.GT.0) THEN
        IF(KPER.GT.1) THEN
          DO 140 K=1,NLAY
          DO 140 I=1,NROW
          DO 140 J=1,NCOL
  140       RHS(J,I,K)=RHS(J,I,K)-HOLD(J,I,K)*SC1(J,I,K)/DELT
        ENDIF
      ENDIF
      RETURN
      END

C=======================================================================
      SUBROUTINE SEN1NZ(RHS,HNEW,NODES,ISA)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     CHECK FOR NONZERO VALUES IN HNEW OR RHS
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW(NODES)
      DIMENSION RHS(NODES)
C     ------------------------------------------------------------------
C
      ISA=0
      DO 100 I=1,NODES
        IF(RHS(I).NE.0..OR.HNEW(I).NE.0.D0) THEN
          ISA=1
          RETURN
        ENDIF
  100 CONTINUE
      RETURN
      END

C=======================================================================
      SUBROUTINE SEN1HZ(HNEW,KPER,NODES,NZER,IZER,RHS,ISA)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     SET ALL COMPONENTS OF HNEW AND RHS TO ZERO
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION HNEW(NODES)
      DIMENSION IZER(NZER),RHS(NODES)
C     ------------------------------------------------------------------
      ITS=KPER-1
      DO 50 I=1,NZER
        IF(IZER(I).EQ.ITS) THEN
          DO 70 J=1,NODES
          RHS(J)=0.
   70     HNEW(J)=0.D0
          ISA=0
          RETURN
        ENDIF
   50 CONTINUE
      RETURN
      END

C=======================================================================
      SUBROUTINE SEN1FD(IFLD,NDER,NH,KPER,TOFF,NQ,NQOB,IQOB,NQT)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     ARE THERE ANY OBSERVATIONS AT THIS TIME STEP?
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION TOFF(NH+NQT),NDER(5,NH),NQOB(NQ),IQOB(NQT)
C     ------------------------------------------------------------------
C
```

```
      IFLD=0
      ITS=KPER-1
      DO 100 N=1,NH
        NTS=NDER(4,N)
        IF(NTS.EQ.ITS.OR.(NTS.EQ.ITS-1.AND.TOFF(N).GT.0.)) THEN
          IFLD=1
          RETURN
        ENDIF
  100 CONTINUE
      NT1=1
      DO 200 IQ=1,NQ
        NT2=NT1+NQOB(IQ)-1
        DO 150 NT=NT1,NT2
          NTS=IQOB(NT)
          IF(NTS.EQ.ITS.OR.(NTS.EQ.ITS-1.AND.TOFF(NH+NT).GT.0.)) THEN
            IFLD=1
            RETURN
          ENDIF
  150   CONTINUE
        NT1=NT2+1
  200 CONTINUE
      RETURN
      END


C=======================================================================
      SUBROUTINE SEN1OT(
     1              IUHEAD,IOUT,NROW,NCOL,NLAY,NP,ISN,NH,NDER,HNEW,PID,
     2              DID,IP,KPER,BUFF,KSTP,PERTIM,TOTIM,X,DELR,DELC,
     3              IBOUND,COFF,ROFF,H,WT,HOBS,IPRINT,IFO,ITERP,IPAR,
     4              RINT,JOFF,IOFF,MLAY,PR,MOBS,NPER,B,LN,NQ,NQC,NQT,
     5              NQOB,NQCL,IQOB,QCLS,IBT,MXBND,NBOUND,BNDS,MXRIVR,
     6              NRIVER,RIVR,SHNW,LASTX,ISCALS,TOFF,MXDRN,NDRAIN,
     7              DRAI,MXSTRM,NSTREM,STRM,ISTRM,MAXM,KPRINT,JDRY,IDRY,
     8              NPR,WP,MPR,PRM,IWPG,B1,IOUB,RSQ,RSQP,RSQO,RSQOO,NPO,
     9              SOSC,SOSR)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     WRITE HEADS ON FILE IUHEAD.
C     INTERPOLATE AND SAVE HEADS AND SENSITIVITIES FOR OBSERVATIONS.
C     PRINT DATA FOR OBSERVED HEAD POINTS AND FOR OBSERVED FLOWS.
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 PID(NP),DID(NH+NQT),TEXT
      CHARACTER*8 MEAN,CVAR,CORR
      DOUBLE PRECISION  HNEW(NCOL,NROW,NLAY)
      DIMENSION NDER(5,NH),BUFF(NCOL*NROW*NLAY),TEXT(4),X(NP,NH+NQT),
     1     DELR(NCOL),DELC(NROW),IBOUND(NCOL,NROW,NLAY),COFF(NH),
     1     ROFF(NH),H(NH+NQT),WT(NH+NQT),HOBS(NH+NQT),RINT(4,NH),
     1     JOFF(NH),IOFF(NH),MLAY(MAXM,MOBS),PR(MAXM,MOBS),B(NPO),
     1     RIVR(6,MXRIVR),BNDS(5,MXBND),IBT(2,NQ),NQOB(NQ),NQCL(NQ),
     1     IQOB(NQT),QCLS(4,NQC),TOFF(NH+NQT),DRAI(5,MXDRN),
     1     STRM(11,MXSTRM),ISTRM(5,MXSTRM),LN(NP),WP(NPO+MPR),
     1     PRM(NPO+1,MPR),IWPG(NP),SHNW(NCOL,NROW,NLAY),B1(NPO)
      COMMON /FLWCOM/LAYCON(80)
C
      DATA TEXT(1),TEXT(2),TEXT(3),TEXT(4) /'    ','    ','    ',
     1        'HEAD'/
      DATA MEAN,CVAR,CORR/'    MEAN','COEF.VAR','CORRELAT'/
C     ------------------------------------------------------------------
C
530     FORMAT(8G10.3)
539     FORMAT(' ')
542     FORMAT(4X,'MULTIPLE LAYERS AND PROPORTIONS :',5(I5,',',F5.2,3X))
545     FORMAT(' SENSITIVITIES',//,' PARAMETER # :',3X,I6,9I10,/
     1     ,12(14X,3X,I6,9I10,/))
546     FORMAT(' SCALED SENSITIVITIES (SCALED BY B*(WT**.5))',
     1 //,' PARAMETER # :',3X,I6,9I10,/,12(14X,3X,I6,9I10,/))
547     FORMAT(' PARAMETER ID :',7X,A4,6X,A4,6X,A4,6X,A4,6X,A4,6X,A4,
     1     6X,A4,6X,A4,6X,A4,6X,A4,/,12(15X,7X,A4,6X,A4,6X,A4,6X,A4,
     1     6X,A4,6X,A4,6X,A4,6X,A4,6X,A4,6X,A4,/))
548     FORMAT(' OBS# & ID')
550     FORMAT(I5,2X,A4,6X,10G10.3,12(/,17X,10G10.3))
590     FORMAT(/,7X,'((SUM OF THE SQUARED VALUES)**.5)/ND'
     1     ,13(/,17X,10E10.3))
605     FORMAT(/,' PARAMETER ESTIMATION CONVERGED BECAUSE SUM OF',
     1  ' SQUARED, WEIGHTED RESIDUALS HAS NOT CHANGED',F7.4,
     1  ' PERCENT IN 2 ITERATIONS',/)
610     FORMAT(/,' ADD R MATRIX OF EQUATION (51) TO EQUATION',
     1  '  (50) OF TEXT FOR ALL SUBSEQUENT ITERATIONS BECAUSE THE ',/,
     2  'SUM OF SQUARED, WEIGHTED RESIDUALS HAS NOT CHANGED MORE THAN',
     3  F7.4,' PERCENT IN 2 ITERATIONS')
```

```
C
C-------WRITE HEADS ON SCRATCH FILE OPENED IN SEN1FM
      IF(IP.EQ.0.AND.(ISN.EQ.-2.OR.ISN.GT.0)) THEN
        IUNIT=IUHEAD+KPER-1
        NRC=NROW*NCOL
        DO 10 K=1,NLAY
          CALL USAVEF(DUM,BUFF,HNEW(1,1,K),DUM,NROW*NCOL,3)
          CALL ULASAV(BUFF,TEXT,KSTP,KPER,PERTIM,TOTIM,NCOL,NROW,K,
     1       IUNIT)
   10   CONTINUE
        REWIND(IUNIT)
C------PUT HEADS FROM FINAL TIME STEP IN SHNW
        IF(ISN.GT.0.AND.KPER.EQ.NPER)
     1        CALL USAVEF(DUM,SHNW,HNEW,DUM,NROW*NCOL*NLAY,3)
      ENDIF
C------WRITE INITIAL PARAMETER VALUES ON FILE IOUB
      IF(ITERP.EQ.1.AND.IP.EQ.0.AND.KPER.EQ.1.AND.IOUB.GT.0)
     1          WRITE(IOUB,530) (B(I),I=1,NP)
C-------INTERPOLATE TO HEAD LOCATIONS
      IF(NH.GT.0) CALL SSEN1U(NH,NDER,IOFF,JOFF,KPER,MLAY,IBOUND,RINT,
     1    DID,COFF,ROFF,DELR,DELC,NCOL,NROW,NLAY,PR,H,HOBS,WT,HNEW,
     1    X,IP,IDRY,NP,MOBS,B,LN,ISN,NQT,TOFF,NPER,RHS,PV,
     1    MAXM,ITERP,JDRY,IPAR,IOUT)
C
      IF(NQ.GT.0) THEN
C-------INITIALIZE CALCULATED FLOW AND SENSITIVITY ARRAYS.
        ITS=KPER-1
        IF(IP.EQ.0.AND.ITS.EQ.0) THEN
          DO 30 NT=1,NQT
   30     H(NH+NT)=0.
        ENDIF
        IF(IP.EQ.1.AND.ITS.EQ.0) THEN
          DO 40 NT=1,NQT
          DO 40 IIP=1,NP
   40       X(IIP,NH+NT)=0.
        ENDIF
C-------CALCULATE FLOWS AND SENSITIVITIES FOR FLOW BOUNDARIES
        CALL SSEN1V(NQ,NQC,NQT,NQOB,NQCL,IQOB,QCLS,IBT,MXBND,
     1    NBOUND,BNDS,MXRIVR,NRIVER,RIVR,SHNW,IP,HNEW,NCOL,
     1    NROW,NLAY,IOUT,IBOUND,NPER,KPER,NH,X,DID,NP,H,
     1    B,LN,TOFF,MXDRN,NDRAIN,DRAI,MXSTRM,NSTREM,STRM,ISTRM,ISN,
     1    RHS,WT,HOBS)
      ENDIF
C
C-----CALCULATE THE SUM OF SQUARED RESIDUALS AND PRINT RESIDUALS FOR
C-----OBSERVATIONS AND PRIOR INFORMATION
C-------SET FLAG WHICH CONTROLS PRINTING OF RESIDUALS
      IO=0
      IF(KPER.EQ.NPER.AND.(IPAR.LT.0.OR.
     1     (IP.EQ.0.AND.(IPRINT.NE.0.OR.ITERP.EQ.1.OR.IFO.NE.0))))
     1       IO=1
      IF(KPER.EQ.NPER.AND.IP.EQ.0) THEN
        CALL SSEN1O(NP,B,B1,IWPG,PID,WP,IOUT,NH,H,HOBS,WT,DID,NDER,
     2                 ROFF,COFF,TOFF,NQT,IQOB,ND,IPAR,NPR,MPR,PRM,
     3                 RSQ,RSQP,IO,LN,RSQH,NPO)
        IF(IOUB.GT.0) WRITE(IOUB,530) RSQH,RSQ,RSQP
      ENDIF
C-----TEST FOR REDUCTION IN SUM OF SQUARED, WEIGHTED RESIDUALS
      IF(IPAR.GT.0.AND.IP.EQ.NP.AND.KPER.EQ.NPER.AND.ISN.LT.0.AND.
     1       (SOSC.GT.0..OR.SOSR.GT.0.)) THEN
        IF(ITERP.GT.2) THEN
          TEMP=(ABS(RSQP-RSQO)/RSQP)+(ABS(RSQO-RSQOO)/RSQO)
          IF(IFO.EQ.0.AND.TEMP.LT.SOSC) THEN
            IFO=2
            WRITE(IOUT,605) SOSC*100.
            IF(IO.EQ.0) CALL SSEN1O(NP,B,B1,IWPG,PID,WP,IOUT,NH,H,HOBS,
     1            WT,DID,NDER,ROFF,COFF,TOFF,NQT,IQOB,ND,IPAR,NPR,MPR,
     3                  PRM,RSQ,RSQP,1,LN,RSQH,NPO)
          ENDIF
          IF(IFO.EQ.0.AND.TEMP.LT.SOSR) THEN
            WRITE(IOUT,610) SOSR*100.
            SOSR=-SOSR
          ENDIF
        ENDIF
        IF(ITERP.GE.2) RSQOO=RSQO
        IF(ITERP.GE.1) RSQO=RSQP
      ENDIF
C-----IF REQUESTED, PRINT UNSCALED OR SCALED SENSITIVITIES
      ND=NH+NQT
      IF(KPER.EQ.NPER.AND.
     1 ((IP.EQ.NP.AND.(IPRINT.NE.0.OR.ITERP.EQ.1.OR.IFO.NE.0)).OR.
     1  (IP.EQ.0.AND.IFO.NE.0.AND.LASTX.EQ.0))) THEN
        WRITE(IOUT,539)
```

```
        IF(ISCALS.EQ.0.AND.KPRINT.LE.1)WRITE(IOUT,545)(IIP,IIP=1,NP)
        IF(ISCALS.GT.0.OR.KPRINT.GT.1)WRITE(IOUT,546)(IIP,IIP=1,NP)
        WRITE(IOUT,547) (PID(IIP),IIP=1,NP)
        WRITE(IOUT,548)
        IF(ISCALS.EQ.0.AND.KPRINT.LE.1) THEN
          DO 95 N=1,ND
            WRITE(IOUT,550)  N,DID(N),(X(IIP,N),IIP=1,NP)
   95     CONTINUE
        ELSE
          DO 98 IIP=1,NP
   98     BUFF(NP+IIP)=0.
          DO 110 N=1,ND
            DO 100 IIP=1,NP
              IF(WT(N).LT.0.) THEN
                BUFF(IIP)=0.
                GO TO 100
              ENDIF
              BB=B(IIP)
              IF(LN(IIP).GT.0) BB=ALOG(BB)
              IF(ABS(BB).LT.1.E-25) BB=1.0
              BUFF(IIP)=BB*X(IIP,N)*(WT(N)**.5)
              BUFF(NP+IIP)=BUFF(NP+IIP)+(BUFF(IIP)*BUFF(IIP))
  100       CONTINUE
            WRITE(IOUT,550)  N,DID(N),(BUFF(IIP),IIP=1,NP)
  110     CONTINUE
          DO 112 IIP=1,NP
  112       BUFF(NP+IIP)=(BUFF(NP+IIP)**.5)/ND
          WRITE(IOUT,590) (BUFF(NP+IIP),IIP=1,NP)
        ENDIF
      ENDIF
C
      RETURN
      END
```

New SEN Submodules

Listed in alphabetical order

```
      SUBROUTINE SSEN1A(
     1              PID,LM,LZ,NLL,SMAT,MATZ,BUFF,NCOL,NROW,NLAY,
     2              NSM,NP,NMM,NZM,LZI1,NLLI1,NLLIT,NPER,IOUT,IERR,NUMR)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     CHECK FOR MULTIPLE USE OF A SINGLE CELL IN PARAMETER DEFINITIONS.
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 PID(NP),PID1,PID2,PIDTMP,TEXT(4)
      DIMENSION BUFF(NCOL,NROW,NLAY),LM(NSM),LZ(LZI1,NSM),
     1          SMAT(NCOL,NROW,NMM),MATZ(NCOL,NROW,NZM),NLL(NLLI1,NP)
C     ------------------------------------------------------------------
500   FORMAT(' AT LEAST ONE GRID CELL IS USED MORE THAN NUMR TIMES IN',
     1 ' DEFINING THE FOLLOWING MODEL INPUTS -- STOP EXECUTION',/,
     2 ' THE NUMBERS EQUAL THE NUMBER OF TIMES THE CELL IS REPEATED')
C
C-----INITIALIZE VARIABLES
      TEXT(1)=' INP'
      TEXT(2)='UT T'
      TEXT(3)='YPE '
      ITS=KPER-1
C-----MODEL INPUT TYPE LOOP
      DO 350 II=1,5
        KP2=1
        IF(II.GE.4) KP2=NPER
        DO 300 KPER=1,KP2
          IFLAG=0
          DO 5 NL=1,NLAY
          DO 5 I=1,NROW
          DO 5 J=1,NCOL
    5     BUFF(J,I,NL)=0.
          IF(II.EQ.1) THEN
            PID1='T   '
            PID2='TKV '
            GO TO 50
          ELSEIF(II.EQ.2) THEN
            PID1='KV  '
            PID2='TKV '
            GO TO 50
          ELSEIF(II.EQ.3) THEN
            PID1='S1  '
            PID2=PID1
```

```
            GO TO 50
          ELSEIF(II.EQ.4) THEN
            PID1='RCH '
            PID2=PID1
            GO TO 50
          ELSEIF(II.EQ.5) THEN
            PID1='ETM '
            PID2=PID1
            GO TO 50
          ENDIF
C---------LOOP THROUGH PARAMETERS
   50     NM=0
          DO 200 IP=1,NP
            PIDTMP=PID(IP)
            IF(PIDTMP.NE.'T   '.AND.PIDTMP.NE.'TKV '.AND.PIDTMP.NE.
     1       'KV  '.AND.PIDTMP.NE.'S1  '.AND.PIDTMP.NE.'RCH '.AND.
     2       PIDTMP.NE.'ETM ') GO TO 200
            NL2=NLLI1
            NT=1
            IF(PIDTMP.EQ.'RCH '.OR.PIDTMP.EQ.'ETM ') THEN
              NL2=1
              NT=NLLIT
            ENDIF
C-------LOOP THROUGH LAYER NUMBERS
            DO 90 LL=1,NL2
              NLO=NLL(LL,IP)
              IF(NLO.GT.0) NM=NM+1
C-------MATCH?
              IF(PIDTMP.NE.PID1.AND.PIDTMP.NE.PID2) GO TO 90
C-------IN SOME CASES JUST KEEP TRACK OF NM
              IF(PIDTMP.EQ.'TKV '.AND.((II.EQ.1.AND.MOD(LL+1,3).NE.0)
     1          .OR.(II.EQ.2.AND.MOD(LL+1,3).EQ.0))) GO TO 90
              IF(PIDTMP.NE.'RCH '.AND.PIDTMP.NE.'ETM ') THEN
                IF(KPER.GT.1) GO TO 90
                IF(NLO.EQ.0.AND.PIDTMP.EQ.'TKV ') GO TO 90
                IF(NLO.EQ.0) GO TO 200
              ENDIF
              IFLAG=1
C-------INITIALIZE NL
              NL=IABS(NLO)
              IF(PIDTMP.EQ.'RCH '.OR.PIDTMP.EQ.'ETM ') NL=1
C-------LOOP THROUGH TIMES FOR RCH AND ETM; NT=1 FOR OTHER PID'S
              DO 85 K=1,NT
                IF((PIDTMP.EQ.'RCH '.OR.PIDTMP.EQ.'ETM ').AND.
     1          (ITS.LT.NLL(2*K,IP).OR.ITS.GT.NLL(2*K+1,IP))) GO TO 85
                IF(NLO.LT.0)GO TO 20
                M=LM(NM)
                LZ1=LZ(1,NM)
                LZN=2
                IF(LZ1.NE.0) LZN=LZI1
                DO 15 IZ=2,LZN
                  IF(LZ1.NE.0)THEN
                    NZ=LZ(IZ,NM)
                    IF(NZ.EQ.0) GO TO 20
                  ENDIF
                  DO 10 I=1,NROW
                  DO 10 J=1,NCOL
                    ICO=0
                    IF(LZ1.EQ.0) THEN
                      IF(M.EQ.0) ICO=1
                      IF(M.NE.0.AND.SMAT(J,I,M).NE.0.)
     1                  ICO=1
                    ELSE
                      IF(MATZ(J,I,LZ1).EQ.NZ) THEN
                        IF(M.EQ.0) ICO=1
                        IF(M.NE.0.AND.SMAT(J,I,M).NE.0.)
     1                    ICO=1
                      ENDIF
                    ENDIF
                    IF(ICO.NE.0) THEN
                      KK=NL
                      IF(PIDTMP.EQ.'TKV '.AND.MOD(LL+2,3).EQ.0) KK=NL-1
                      BUFF(J,I,KK)=BUFF(J,I,KK)+1.
                    ENDIF
   10             CONTINUE
   15           CONTINUE
   20           CONTINUE
                IF(NLO.LT.0) THEN
                  DO 80 I=1,NROW
                  DO 80 J=1,NCOL
                    KK=NL
                    IF(PIDTMP.EQ.'TKV '.AND.MOD(LL+2,3).EQ.0) KK=NL-1
                    BUFF(J,I,KK)=BUFF(J,I,KK)+1
```

```
 80             CONTINUE
              ENDIF
 85         CONTINUE
 90       CONTINUE
200     CONTINUE
        IF(IFLAG.EQ.0) GO TO 350
        NL2=NLAY
        IF(II.EQ.2) NL2=NLAY-1
        IF(II.EQ.4.OR.II.EQ.5) NL2=1
        DO 260 NL=1,NL2
          DO 240 I=1,NROW
          DO 240 J=1,NCOL
240       IF(BUFF(J,I,NL).GT.REAL(NUMR)) GO TO 250
          GO TO 260
250       TEXT(4)=PID1
          IERR=1
          WRITE(IOUT,500)
          CALL ULAPRS(BUFF(1,1,NL),TEXT,ITS,0,NCOL,NROW,NL,8,IOUT)
260     CONTINUE
300   CONTINUE
350 CONTINUE
    RETURN
    END


C===========================================================================
      SUBROUTINE SSEN1B(
     1          NLL,NCOL,NROW,CELS,NS,NSN,DELR,DELC,IOUT,IERR,NLLI1)
C-----VERSION 1000 01FEB1992
C     ***********************************************************************
C     CALCULATE COEFFICIENTS FOR CONSTANT-HEAD BOUNDARIES.
C     ONLY INTERPOLATE BACKWARDS FROM EACH ENDPOINT.
C     ***********************************************************************
C        SPECIFICATIONS:
C     -----------------------------------------------------------------------
      DIMENSION CELS(4,NSN),NLL(NLLI1),DELR(NCOL),DELC(NROW)
C     -----------------------------------------------------------------------
580   FORMAT(/,' LINEAR INTERPOLATION FOR CONSTANT HEADS REQUESTED '
     1  'ALONG A DIAGONAL OR AROUND A CORNER -- STOP EXECUTION ',
     2  '(SSEN1B)',/)
C
      NL1=NLL(1)
      IICH=0
      N1=NS-NL1+1
      DO 10 I=N1,NS
   10 CELS(4,I)=0.
C-------FIND THE NEXT PARAMETER OR KNOWN HEAD LOCATION.
   22 NN2=NL1-IICH
      DO 25 NN=1,NN2
        N=N1-1+IICH+NN
        CE=CELS(3,N)
        IF(CE.NE.0.) THEN
          CELS(4,N)=CE
          IICH=IICH+NN
          GO TO 26
        ENDIF
   25 CONTINUE
   26 CONTINUE
C-------COUNT HOW FAR BACK TO INTERPOLATE
        N0=0
        IF(IICH.GT.2.AND.CELS(3,N1-1+IICH-1).EQ.0.) THEN
          I1=IICH-1
          DO 40 I=I1,1,-1
            IF(CELS(3,N1-1+I).NE.0.) GO TO 41
   40     CONTINUE
   41     CONTINUE
          N0=IABS(I-IICH)-1
          IF(N0.EQ.1) GO TO 60
C-------CALCULATE TOTAL INTERPOLATION DISTANCE
          TX=0.
          TY=0.
          DO 45 NN=1,N0+2
            N=N1-1+IICH-(NN-1)
            FAC=1.0
            IF(NN.EQ.1.OR.NN.EQ.N0+2) FAC=0.5
            IC=INT(CELS(1,N))
            JC=INT(CELS(2,N))
C----------ALONG A ROW
            IF(N.LT.NS.AND.IC.EQ.INT(CELS(1,N+1)))
     1            TX=TX+FAC*DELR(JC)
            IF(N.EQ.NS.AND.INT(CELS(1,N-1)).EQ.IC)
     1            TX=TX+FAC*DELR(JC)
C----------ALONG A COLUMN
```

```
          IF(N.LT.NS.AND.JC.EQ.INT(CELS(2,N+1)))
     1          TY=TY+FAC*DELC(IC)
          IF(N.EQ.NS.AND.INT(CELS(2,N-1)).EQ.JC)
     1          TY=TY+FAC*DELC(IC)
   45     CONTINUE
          IF(TX.GT.0..AND.TY.GT.0.) THEN
            WRITE(IOUT,580)
            IERR=1
          ENDIF
C-------CALCULATE PARTIAL DISTANCES AND COEFFICIENTS
          DX=0.
          DY=0.
          DO 50 NN=1,NO
            N=N1-1+IICH-NN
C----------ALONG ROWS
            IF(TX.GT.0.) THEN
              DX = DX + 0.5*(DELR(INT(CELS(2,N+1)))
     1                     +DELR(INT(CELS(2,N))))
              CELS(4,N)=1.-DX/TX
            ENDIF
C----------ALONG COLUMNS
            IF(TY.GT.0) THEN
              DY = DY + .5*(DELC(INT(CELS(1,N+1)))
     1                     +DELC(INT(CELS(1,N))))
              CELS(4,N)=1.-DY/TY
            ENDIF
   50     CONTINUE
        ENDIF
C----------IS THE NEXT PARAMETER THE SAME?
   60 NNN2=NL1-IICH
      DO 72 NNN=1,NNN2
        N=N1-1+IICH+NNN
        CE=CELS(3,N)
        CEN=CELS(3,N1-1+IICH)
        IF(CE.GT.1..AND.CEN.GT.1.) THEN
          IF(CE.EQ.CEN) THEN
            DO 70 NNNN=1,NNN
   70       CELS(4,N1-1+IICH+NNNN)=1.
            IICH=IICH+NNN
          ENDIF
        ENDIF
   72 CONTINUE
      IF(IICH.LT.NL1) GO TO 22
      RETURN
      END


C=====================================================================
      SUBROUTINE SSEN1C(
     1                  IP,NLL,NCOL,NROW,NLAY,CR,CC,CV,IBOUND,XX,NP,
     2                  CELS,NS,NSN,ISN,HNEW,IFLCH,DD,IHESS,RHS,H,HOBS,
     3                  RINT,IOFF,JOFF,NDER,NH,KPER,NPER,TOFF,SCL,NLLI1)
C-----VERSION 1000 01FEB1992
C     ****************************************************************
C     FOR CONSTANT HEAD NODES - CALCULATE ADJOINT SENSITIVITY, OR SET
C         VALUES IN HNEW FOR THE SENSITIVITY EQUATIONS OR ITERATION
C         PARAMETER VECTOR
C     ****************************************************************
C        SPECIFICATIONS:
C     ----------------------------------------------------------------
      DOUBLE PRECISION HNEW(NCOL,NROW,NLAY),DD(NP),XX(NP)
      DIMENSION CR(NCOL*NROW*NLAY),CC(NCOL*NROW*NLAY),CELS(4,NSN),
     1      CV(NCOL*NROW*NLAY),IBOUND(NCOL,NROW,NLAY),NLL(NLLI1),
     2      RHS(NCOL,NROW,NLAY),H(NH),HOBS(NH),RINT(4,NH),IOFF(NH),
     3      JOFF(NH),NDER(5,NH),TOFF(NH)
C     ----------------------------------------------------------------
C
      ITS=KPER-1
      IF(ISN.GT.0.AND.IHESS.EQ.0) ITS=NPER-KPER
      NL1=NLL(1)
      NL2=NLL(2)
      NL3=NLL(3)
      IF(IFLCH.EQ.0) NS=NS+NL1
      IFLCH=IFLCH+1
      IF((NL3.EQ.0.AND.ITS.NE.0).OR.(NL3.EQ.1.AND.ITS.EQ.0)) GO TO 150
      N1=NS-NL1+1
C-----FIND PARAMETER AND THE PRECEDING SECTION OF THE BOUNDARY.
      II=0
      I=0
      DO 30 INCH=N1,NS
        CE=CELS(4,INCH)
        IF(II.EQ.IFLCH-1.AND.INCH.GT.N1) THEN
          CEM1=CELS(4,INCH-1)
```

```
        IF((CE.GT.0..AND.CE.LT.1.).AND.
     1      (CEM1.LT.0..OR.CEM1.GE.1.)) I=INCH
        IF(CE.GT.1.) GO TO 32
      ENDIF
      IF(CE.GT.1.) II=II+1
      IF(INCH.EQ.N1.AND.II.EQ.IFLCH) GO TO 32
   30 CONTINUE
   32 CONTINUE
      IF(I.EQ.0) I=INCH
      IEND=INCH
C-----DOES THE PARAMETER EQUAL THE HEAD AT SUBSEQUENT CELLS OR IS THIS
C-----PARAMETER USED TO INTERPOLATE FURTHER?
      IF=0
      IF(IEND.LT.NS) THEN
        I1=IEND+1
        DO 40 INCH=I1,NS
          CE=CELS(4,INCH)
          IF(CE.GT.1..OR.CE.LT.0.) THEN
            IEND=INCH-1
            GO TO 42
          ENDIF
          IF(CE.NE.1..AND.IF.EQ.0) IF=INCH
   40   CONTINUE
   42   CONTINUE
      ENDIF
      IF(IF.EQ.0) IF=NS+1
C-----LOOP THROUGH THIS SECTION OF BOUNDARY
      DO 100 INCH=I,IEND
        FAC=CELS(4,INCH)
        IF(FAC.GT.1) FAC=1.
        IF(INCH.GE.IF) FAC=1.-FAC
        II=CELS(1,INCH)
        JJ=CELS(2,INCH)
C--------LOOP THROUGH LAYERS
        DO 50 NL=4,NLLI1
          KK=NLL(NL)
          IF(KK.EQ.0) GO TO 100
          IND=JJ+NCOL*(II-1)+NRC*(KK-1)
C-----------ADJOINT-STATE SENSITIVITIES
          IF(ISN.GT.0.AND.IHESS.EQ.0) THEN
C--------------CV
            IF(KK.GT.1.AND.IBOUND(JJ,II,KK-1).GT.0)
     #        XX(IP)=XX(IP)+FAC*CV(IND-NRC)*HNEW(JJ,II,KK-1)
            IF(KK.LT.NLAY.AND.IBOUND(JJ,II,KK+1).GT.0)
     #        XX(IP)=XX(IP)+FAC*CV(IND)*HNEW(JJ,II,KK+1)
C--------------CC
            IF(II.GT.1.AND.IBOUND(JJ,II-1,KK).GT.0)
     #        XX(IP)=XX(IP)+FAC*CC(IND-NCOL)*HNEW(JJ,II-1,KK)
            IF(II.LT.NROW.AND.IBOUND(JJ,II+1,KK).GT.0)
     #        XX(IP)=XX(IP)+FAC*CC(IND)*HNEW(JJ,II+1,KK)
C--------------CR
            IF(JJ.GT.1.AND.IBOUND(JJ-1,II,KK).GT.0)
     #        XX(IP)=XX(IP)+FAC*CR(IND-1)*HNEW(JJ-1,II,KK)
            IF(JJ.LT.NCOL.AND.IBOUND(JJ+1,II,KK).GT.0)
     #        XX(IP)=XX(IP)+FAC*CR(IND)*HNEW(JJ+1,II,KK)
C--------------DOES THIS CH DIRECTLY AFFECT A HEAD OBSERVATION?
            DO 46 N=1,NH
              IF(KK.EQ.NDER(1,N).AND.(NDER(4,N).EQ.ITS.OR.
     1          (NDER(4,N)-1.EQ.ITS.AND.TOFF(N).GT.0.0))) THEN
                FACT=1.0
                IF(TOFF(N).GT.0.0) THEN
                  IF(NDER(4,N).EQ.ITS) FACT=TOFF(N)
                  IF(NDER(4,N)-1.EQ.ITS)  FACT=1.-TOFF(N)
                ENDIF
                IH=NDER(2,N)
                JH=NDER(3,N)
                IO=IOFF(N)
                JO=JOFF(N)
                FACH=0.
                IF(II.EQ.IH.AND.JJ.EQ.JH) FACH=RINT(1,N)
                IF(II.EQ.IH.AND.JJ.EQ.JH+JO) FACH=FACH+RINT(2,N)
                IF(II.EQ.IH+IO.AND.JJ.EQ.JH) FACH=FACH+RINT(3,N)
                IF(II.EQ.IH+IO.AND.JJ.EQ.JH+JO) FACH=FACH+RINT(4,N)
                IF(FACH.EQ.0.) GO TO 46
                XX(IP)=XX(IP)+2.*(H(N)-HOBS(N))*FAC*FACH*FACT
              ENDIF
   46       CONTINUE
          ENDIF
C
C-----------FOR SENSITIVITY-EQUATION SENSITIVITIES
          IF(ISN.LT.0) HNEW(JJ,II,KK)=FAC
C-----------FOR ITERATION PARAMETER VECTOR
          IF(ISN.GT.0.AND.IHESS.NE.0) THEN
```

```
            IF(IFLCH.EQ.1) HNEW(JJ,II,KK)=0.
            HNEW(JJ,II,KK)=HNEW(JJ,II,KK)+FAC*DD(IP)*SCL
          ENDIF
   50   CONTINUE
  100 CONTINUE
C
C-----IS THIS THE LAST PARAMETER OF THIS CH IDENTIFIER?
  150 IF(IFLCH.EQ.NL2) IFLCH=0
      RETURN
      END


C=====================================================================
      SUBROUTINE SSEN1D(
     1                  PID,NLL,NNB,MXBN,BN,CELS,NSN,NS,SHNW,HNEW,NCOL,
     2                  NROW,NLAY,IOUT,I1BN,IBOUND,NPER,KPER,XX,ISN,
     3                  RHS,ISTRM,DD,IHESS,B,LN,SCL,NLLI1,NLLIT)
C-----VERSION 1000 01FEB1992
C     *****************************************************************
C     FOR PUMPING AND HEAD-DEPENDANT BOUNDARIES : CALCULATE MATRIX AND
C     VECTOR DERIVATIVES, MULTIPLY BY HEADS AND ADJOINTS, AND ADD
C     COMPONENTS TO THE SENSITIVITY OR RHS.
C     *****************************************************************
C        SPECIFICATIONS:
C     -----------------------------------------------------------------
      CHARACTER*4 PID
      DOUBLE PRECISION HNEW(NCOL,NROW,NLAY),DD,XX,DF
      DIMENSION BN(I1BN,MXBN),CELS(4,NSN),SHNW(NCOL,NROW,NLAY),
     1     IBOUND(NCOL,NROW,NLAY),NLL(NLLI1),
     I     RHS(NCOL,NROW,NLAY),ISTRM(5,MXBN)
C     -----------------------------------------------------------------
C
C-------CALCULATE TIME STEP
      ITS=NPER-KPER
      IF(ISN.LT.0.OR.IHESS.GT.0) ITS=KPER-1
      NS1=NS
C-------LOOP THROUGH PARAMETER TIME DEFINITION
      DO 250 KT=1,NLLIT
        IF(NLL(2*KT).LT.0) GO TO 255
        IF( ITS.GE.NLL(2*KT) .AND. ITS.LE.NLL(2*KT+1) ) THEN
          NL1=NLL(1)
          NL1A=IABS(NL1)
C-------LOOP THROUGH PARAMETER CELLS
          DO 200 II=1,NL1A
            DF=0.
            NS=NS+1
            K=CELS(1,NS)
            I=CELS(2,NS)
            J=CELS(3,NS)
            IF(PID.NE.'KST '.AND.IBOUND(J,I,K).LE.0) GO TO 200
C-------LOOP THROUGH TO SEE IF THIS CELL IS BEING USED.
C-------IF SO, CALCULATE CONTRIBUTION TO SENSITIVITY.
            DO 100 JJ=1,NNB
              IF(PID.NE.'KST ') THEN
                KB=BN(1,JJ)
                IB=BN(2,JJ)
                JB=BN(3,JJ)
              ELSE
                IS=ISTRM(4,JJ)
                IR=ISTRM(5,JJ)
              ENDIF
              IF((PID.NE.'KST '.AND.KB.EQ.K .AND. IB.EQ.I .AND. JB.EQ.J)
     1           .OR.(PID.EQ.'KST '.AND.IS.EQ.K.AND.IR.EQ.I))THEN
                IF(PID.EQ.'KST ') THEN
                  IF(BN(10,JJ).LE.0.0.AND.BN(11,JJ).GE.0.0) GO TO 200
                  K=ISTRM(1,JJ)
                  I=ISTRM(2,JJ)
                  J=ISTRM(3,JJ)
                  IF(IBOUND(J,I,K).LE.0) GO TO 200
                ENDIF
                FACTOR=1.
                IF(NL1.GT.0) FACTOR=CELS(4,NS)
                IF(PID.EQ.'GHB ') DF=FACTOR*(BN(4,JJ)-SHNW(J,I,K))
                IF(PID.EQ.'Q   ') DF=FACTOR
                IF(PID.EQ.'KRB '.OR.PID.EQ.'KST '.OR.PID.EQ.'KDR ')
     1            THEN
                  IN=4
                  IF(PID.EQ.'KRB ') IN=6
                  IF(SHNW(J,I,K).GT.BN(IN,JJ)) THEN
                    IM=4
                    IF(PID.EQ.'KST ') IM=2
                    DF=FACTOR*(BN(IM,JJ)-SHNW(J,I,K))
                  ELSE
```

```
                    IF(PID.EQ.'KRB ') DF=FACTOR*(BN(4,JJ)-BN(6,JJ))
                    IF(PID.EQ.'KST ') THEN
                      IM=2
                      IF(BN(10,JJ).LE.0.) IM=5
                      DF=FACTOR*(BN(IM,JJ)-BN(4,JJ))
                    ENDIF
                  ENDIF
                ENDIF
C-------FOR ADJOINT SENSITIVITIES
                IF(ISN.GT.0.AND.IHESS.EQ.0)
     1              XX = XX + HNEW(J,I,K) * DF
C-------FOR ITERATION-PARAMETER VECTOR FOR CONJUGATE-DIRECTION OPT.
                IF(ISN.GT.0.AND.IHESS.GT.0) THEN
                  BB=1.
                  IF(LN.GT.0) BB=B/SCL
                  RHS(J,I,K)=RHS(J,I,K)-DF*DD*BB*SCL
                ENDIF
C-------FOR SENSITIVITY-EQUATION SENSITIVITIES
                IF(ISN.LT.0) RHS(J,I,K) = RHS(J,I,K) - DF
                GO TO 200
              ENDIF
100         CONTINUE
200       CONTINUE
        ENDIF
250   CONTINUE
255   CONTINUE
C
      IF(NS.EQ.NS1) NS=NS+IABS(NLL(1))
      RETURN
      END


C=========================================================================
      SUBROUTINE SSEN1F (
     1      NCOL,NROW,NLAY,MXBND,MXRIVR,NPER,IU,IOUT,DID,NH,JT,
     2      IBOUND,NQ,NQC,NQT,IBT,NQOB,NQCL,IQOB,QCLS,IERR,PID,NP,NLL,
     3      CELS,NSN,WT,HOBS,TOFF,EVF,MXDRN,MXSTRM,ISN,NLLI1,NLLIT)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     READ, CHECK AND STORE DATA FOR HEAD-DEPENDENT FLOW BOUNDARIES.
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4   DID(NH+NQT),PID(NP),PIDTMP
      DIMENSION IBOUND(NCOL,NROW,NLAY),IBT(2,NQ),NQOB(NQ),NQCL(NQ),
     1   IQOB(NQT),QCLS(5,NQC),NLL(NLLI1,NP),CELS(4,NSN),WT(NH+NQT),
     1   HOBS(NH+NQT),TOFF(NH+NQT)
C     ------------------------------------------------------------------
501   FORMAT(15X,2F5.0,F10.0)
503   FORMAT(8F10.0)
510   FORMAT(A4,1X,I5,F8.0,2F10.0,I5)
515   FORMAT(3I5)
517   FORMAT(/,' HEAD-DEPENDENT FLOW DATA')
518   FORMAT(/,'   GROUP#',I3,', BOUNDARY TYPE',I3,
     1 ', # OF FLOWS',I5,', # OF CELLS IN GROUP',I5,/,
     1 40X,'OBSERVED STREAMFLOW',/,
     1 6X,'OBS# ID    TIME STEP AND OFFSET',
     1 '    GAIN(-) OR LOSS(+)  VARIANCE')
520   FORMAT(4X,I5,1X,A4,2X,I5,F10.2,12X,G10.3,3X,G14.3)
522   FORMAT(/,'          LAYER  ROW  COLUMN    FACTOR')
523   FORMAT(/,'        SEGMENT REACH    FACTOR')
525   FORMAT(4X,F8.0,F6.0,F7.0,F9.2)
530   FORMAT(4X,F8.0,F6.0,F9.2)
540   FORMAT(/,' STATISTIC RELATED TO WEIGHT < 0 -- STOP EXECUTION',
     1     ' (SSEN1F)',/)
560   FORMAT(/,' PACKAGE RELATED TO HEAD-DEPENDENT BOUNDARY DATA',
     1     ' IS NOT OPEN -- STOP EXECUTION (SSEN1F)')
570   FORMAT(/,' TIME STEP (',I5,') LARGER THAN NPER (',I5,
     1  ') OF BASIC PACKAGE INPUT FILE -- STOP EXECUTION (SSEN1F)',/)
575   FORMAT(/,' IBT MUST BE 1, 2, 3 OR 4 -- STOP EXECUTION',
     1     ' (SSEN1F)',/)
580   FORMAT(/,' NQC CAN BE REDUCED FROM',I5,' TO',I5)
585   FORMAT(/,' NQT CAN BE REDUCED FROM',I5,' TO',I5)
587   FORMAT(/,' ROW OR COLUMN NUMBER INVALID -- STOP EXECUTION',
     1      ' (SSEN1F)',/)
588   FORMAT(/,' NUMBER OF NODES IN THE NQ CELL GROUPS, ',
     1      I5,' > NQC (',I5,')  OR  NUMBER FLOW OBSERVATIONS',I5,
     1      ' > NQT (',I5,') -- STOP EXECUTION (SSEN1F)',/)
590   FORMAT(/,' LAYER INVALID OR CELL DESIGNATED AS INACTIVE IN '
     1 ,'IBOUND ARRAY OF BASIC PACKAGE INPUT -- STOP EXECUTION '
     2   '(SSEN1F)',/)
C
C-------INITIALIZE VARIABLES
```

```
      DO 10 I=1,NQC
        QCLS(5,I)=0.
   10 CONTINUE
      DO 20 IQ=1,NQ
   20 IBT(2,IQ)=0
      NC=0
      NT=0
C-------WRITE TITLE AND LOOP THROUGH REACHES
      WRITE(IOUT,517)
      DO 200 IQ=1,NQ
        READ(IU,515) IBT(1,IQ),NQOB(IQ),NQCL(IQ)
        WRITE(IOUT,518) IQ,IBT(1,IQ),NQOB(IQ),NQCL(IQ)
C----------SET FLAG FOR SETTING ALL FACTORS TO 1
        I4=0
        IF(NQCL(IQ).LT.0) THEN
          I4=1
          NQCL(IQ)=-NQCL(IQ)
        ENDIF
C----------READ TIME STEPS, MEASURED FLOWS, AND WEIGHTS.
        NT1=NT+1
        NT2=NT+NQOB(IQ)
        DO 115 J=NT1,NT2
          N=NH+J
          READ(IU,510) DID(N),IQOB(J),TOFF(N),HOBS(N),
     1                                WT(N),IWT
          IF(IWT.EQ.2) WT(N)=WT(N)*HOBS(N)
          IF(IWT.GT.0) WT(N)=WT(N)*WT(N)
          WT(N)=WT(N)*EVF
          WRITE(IOUT,520) N,DID(N),IQOB(J),TOFF(N),HOBS(N),WT(N)
          IF(WT(N).LT.0.) THEN
            WRITE(IOUT,540)
            IERR=1
          ENDIF
C----------ERROR CHECKING
          IF(IQOB(J).GT.JT) JT=IQOB(J)
          IF(IQOB(J).GT.NPER) THEN
            WRITE(IOUT,570) IQOB(J),NPER
            IERR=1
          ENDIF
          IBT1=IBT(1,IQ)
          IF(IBT1.LT.1.OR.IBT1.GT.4) THEN
            WRITE(IOUT,575)
            IERR=1
          ELSEIF(IBT1.EQ.1.AND.MXRIVR.EQ.0.OR.
     1           IBT1.EQ.2.AND.MXBND.EQ.0.OR.
     2           IBT1.EQ.3.AND.MXSTRM.EQ.0.OR.
     3           IBT1.EQ.4.AND.MXDRN.EQ.0) THEN
            WRITE(IOUT,560)
            IERR=1
          ENDIF
  115   CONTINUE
C----------READ LAYER, ROW AND COLUMN, OR SEGMENT AND REACH.
        IBT1=IBT(1,IQ)
        NC1=NC+1
        NC2=NC+NQCL(IQ)
        IF(IBT1.NE.3) WRITE(IOUT,522)
        IF(IBT1.EQ.3) WRITE(IOUT,523)
        NNUM=3
        IF(IBT1.EQ.3) NNUM=2
        DO 118 L=NC1,NC2
          IF(IBT1.NE.3) THEN
            READ(IU,503) (QCLS(I,L),I=1,4)
            IF(QCLS(4,L).EQ.0..OR.I4.EQ.1) QCLS(4,L)=1.
            WRITE(IOUT,525) (QCLS(I,L),I=1,4)
          ELSE
            READ(IU,501) (QCLS(I,L),I=1,2),QCLS(4,L)
            IF(QCLS(4,L).EQ.0..OR.I4.EQ.1) QCLS(4,L)=1.
            WRITE(IOUT,530) (QCLS(I,L),I=1,2),QCLS(4,L)
            QCLS(3,L)=0.
            GO TO 118
          ENDIF
          K=QCLS(1,L)
          I=QCLS(2,L)
          J=QCLS(3,L)
          IF(J.LE.0.OR.J.GT.NCOL.OR.I.LE.0.OR.I.GT.NROW) THEN
            WRITE(IOUT,587)
            IERR=1
          ENDIF
          IF(K.LE.0.OR.K.GT.NLAY.OR.IBOUND(J,I,K).EQ.0)THEN
            WRITE(IOUT,590)
            IERR=1
          ENDIF
  118   CONTINUE
```

```
C-------ARE ANY PARAMETERS INVOLVED IN THIS BOUNDARY?
        NS=0
        IFLAG=0
        IFL=0
        ICH=0
        DO 180 IP=1,NP
          PIDTMP=PID(IP)
          INS=NS
          IF((IBT(1,IQ).EQ.1.AND.PIDTMP.EQ.'KRB ').OR.
     1       (IBT(1,IQ).EQ.2.AND.PIDTMP.EQ.'GHB ').OR.
     1       (IBT(1,IQ).EQ.3.AND.PIDTMP.EQ.'KST ').OR.
     1       (IBT(1,IQ).EQ.4.AND.PIDTMP.EQ.'KDR ')) THEN
            DO 170 IT=1,NLLIT
              IF(NLL(2*IT,IP).LT.0) GO TO 175
              DO 165 N=NT1,NT2
                IF(IQOB(N).GE.NLL(2*IT,IP).AND.
     1             IQOB(N).LE.NLL(2*IT+1,IP)) THEN
                  NL1=NLL(1,IP)
                  NL1A=IABS(NL1)
C-------LOOP THROUGH PARAMETER CELLS
                  DO 160 II=1,NL1A
                    INS=INS+1
                    K=CELS(1,INS)
                    I=CELS(2,INS)
                    J=CELS(3,INS)
                    IF(IBOUND(J,I,K).LE.0) GO TO 160
C-------LOOP THROUGH TO SEE IF THIS CELL IS ON THE BOUNDARY
                    DO 150 JJ=NC1,NC2
                      K1=QCLS(1,JJ)
                      I1=QCLS(2,JJ)
                      J1=QCLS(3,JJ)
C-------IF SO, PUT PARAMETER FACTOR IN QCLS(5)
                      IF(QCLS(1,JJ).EQ.K .AND. QCLS(2,JJ).EQ.I .AND.
     #                   QCLS(3,JJ).EQ.J) THEN
                        IBT(2,IQ)=IP
                        IFLAG=1
                        QCLS(5,JJ)=1.
                        IF(NL1.GT.0) QCLS(5,JJ)=CELS(4,INS)
                      ENDIF
  150               CONTINUE
  160             CONTINUE
                  GO TO 175
                ENDIF
  165         CONTINUE
  170       CONTINUE
          ENDIF
  175     CONTINUE
          IF(IFLAG.EQ.1) GO TO 185
          IF(PIDTMP.EQ.'CH  '.AND.ICH.EQ.0) THEN
            ICH=ICH+1
            IF(ICH.EQ.NLL(2,IP)) ICH=0
            GO TO 180
          ENDIF
          IF(PIDTMP.EQ.'KRB '.OR.PIDTMP.EQ.'GHB '.OR.
     1       PIDTMP.EQ.'Q   '.OR.PIDTMP.EQ.'CH  '.OR.
     1       PIDTMP.EQ.'KDR '.OR.PIDTMP.EQ.'KST ')
     1       NS=NS+IABS(NLL(1,IP))
  180   CONTINUE
        IF(IFLAG.EQ.0) IBT(2,IQ)=0
  185   CONTINUE
C-------UPDATE COUNTERS
        NC=NC2
        NT=NT2
  200 CONTINUE
C-------EXCEEDED STORAGE RESERVED FOR FLOW DATA?
      IF(NC.GT.NQC.OR.NT.GT.NQT) THEN
        WRITE(IOUT,588) NC,NQC,NT,NQT
        IERR=1
      ENDIF
C-------ALLOCATED TOO MUCH STORAGE?
      IF(NQC.GT.NC) THEN
        WRITE(IOUT,580) NQC,NC
        NQC=NC
      ENDIF
      IF(NQT.GT.NT) THEN
        WRITE(IOUT,585) NQT,NT
        NQT=NT
      ENDIF
      RETURN
      END
```

```
C=====================================================================
      SUBROUTINE SSEN1G (
     1       NCOL,NROW,NLAY,NSN,NSM,MXBND,MXWELL,MXRIVR,NPER,IU,IOUT,NP,
     2       PID,NLL,CELS,SMAT,IBOUND,TOP,BOT,LN,IERR,IUBD,BUFF,DELR,
     3       DELC,MXDRN,MXSTRM,NMM,NZM,LZI1,SFAC,LZ,LM,MATZ,NLLI1,IUNIT)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     READ, CHECK AND STORE DATA FOR PARAMETER DEFINITION.
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4   PID(NP),PIDTMP,ANAME
      INTEGER       NLL(NLLI1,NP)
      DIMENSION CELS(4,NSN),SMAT(NCOL,NROW,NMM),ANAME(6,2),SFAC(NSM),
     1      IBOUND(NCOL,NROW,NLAY),TOP(NCOL,NROW,NLAY),LZ(LZI1,NSM),
     2      BOT(NCOL,NROW,NLAY),LN(NP),LM(NSM),MATZ(NCOL,NROW,NZM),
     3      BUFF(NCOL,NROW,NLAY),DELR(NCOL),DELC(NROW),IUNIT(24)
C
      DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1),ANAME(5,1),
     # ANAME(6,1) /'   M','ULTI','PLIC','ATIO','N MA','TRIX'/
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2),ANAME(5,2),
     # ANAME(6,2) /'    ','    ','    ',' ZON','E MA','TRIX'/
C     ------------------------------------------------------------------
 501  FORMAT(15X,2F5.0,25X,F10.0)
 503  FORMAT(8F10.0)
 504  FORMAT(3F10.0,10X,F10.0)
 506  FORMAT(24F3.0)
 508  FORMAT(F9.0,F11.0,F9.0,3X,E10.3)
 510  FORMAT(F10.0,15I5)
 520  FORMAT(16I5)
 525  FORMAT(A4,I1,11I5)
 527  FORMAT(/,' # FACTORS >1. (',I4,') DOES NOT EQUAL NLL(2) (',I4,')')
 528  FORMAT(/,' LN>0 IS NOT VALID FOR THIS PID -- STOP EXECUTION ',
     1      '(SSEN1G)')
 530  FORMAT(' SFAC=',G11.3,', MULT. MATRIX',I5,
     1 ', ZONE MATRIX',I5)
 531  FORMAT(' ZONES =',20I5,/,20(8X,20I5,/))
 532  FORMAT(/,' PARAMETER NUMBER:',I5,', PARAMETER IDENTIFIER: ',1X,A4,
     #    /,' LN:',I3,', VECTOR NLL:',11I5,/,20(11I5))
 533  FORMAT(/' NLL VALUE',I5,' NOT VALID -- STOP EXECUTION (SSEN1G)',/)
 534  FORMAT(//,1H ,A5,' FOR PARAMETER NUMBER',I5,' IS NOT A VALID '
     #     'PARAMETER IDENTIFIER --  STOP EXECUTION (SSEN1G)')
 536  FORMAT(/,' NLL(1)',I5,', =0 OR >'
     #  ' MAXIMUM SPECIFIED BY MXWELL, MXRIVR, MXBND, MXSTRM, OR MXDRN'
     #  ' -- STOP EXECUTION (SSEN1G)',/)
 538  FORMAT(/,' MODULAR MODEL PACKAGE RELATED TO THIS PARAMETER IS ',
     1 'NOT OPEN -- STOP EXECUTION (SSEN1G)')
 539  FORMAT(/,' NEED NLLI1>=3 FOR TIMING INFORMATION -- STOP ',
     1 'EXECUTION')
 540  FORMAT(/,' FOR CH, FACTOR MUST BE =0., <0., OR >1. -- '
     1   'STOP EXECUTION',/)
 550  FORMAT(/,' NUMBER OF NODES USED FOR PARAMETER DEFINITION, ',
     1      I5,' >NSN (',I5,')  OR  NUMBER OF MATRICES',I5,' >NSM ('
     1      ,I5,') -- STOP EXECUTION (SSEN1G)',/)
 560  FORMAT(4X,'SEGMENT',5X,'REACH    ZERO      FACTOR')
 565  FORMAT(5X,'LAYER',7X,'ROW    COLUMN     FACTOR')
 570  FORMAT(/,' CELL NOT DESIGNATED AS CONSTANT HEAD IN IBOUND '
     1  'ARRAY OF BASIC PACKAGE UNIT -- STOP EXECUTION (SSEN1G)',/)
 575  FORMAT(/,' ENDS OF CONSTANT HEAD BOUNDARIES MUST HAVE '
     1  'NONZERO VALUES -- STOP EXECUTION (SSEN1G)',/)
 580  FORMAT(/,' NSN CAN BE REDUCED FROM',I5,' TO',I5)
 585  FORMAT(/,' NSM CAN BE REDUCED FROM',I5,' TO',I5)
 590  FORMAT(/,' ERROR IN CH DATA : FOURTH COLUMN ALL 0 OR FIRST OR',
     1  ' LAST VALUE IS 0 -- STOP EXECUTION (SSEN1G)',/)
 600  FORMAT(/,' SPECIFIED ZONATION INCONSISTENT WITH LZI1 AND/OR NZM',
     1  ' (LINE 4) -- STOP EXECUTION')
 610  FORMAT(/,' ERROR IN ANIV DATA : LAYER NUMBER LISTED HERE',
     1 ' APPEARS FOR KV PARAMETER #',I5,' -- STOP EXECUTION')
 620  FORMAT(/,' ERROR IN ANIV DATA : ALL LAYERS MUST APPEAR ',
     1 'IN A PREVIOUS TKV PARAMETER -- STOP EXECUTION')
 630  FORMAT(/,' MULTIPLICATION MATRIX NUMBER INCONSISTENT WITH NMM',
     1 ' (LINE 4) -- STOP EXECUTION')
C
C-------SET COUNTERS
      NS=0
      NM=0
      ICH=0
      IFL=0
C-------SET ERROR INDICATOR
      IERR=0
C-------READ PARAMETER IDENTIFICATION DATA
      DO 100 IP=1,NP
C----------ACCOUNT FOR MULTIPLE CONSTANT-HEAD PARAMETERS UNDER ONE PID
```

```
      IF(IP.GT.1.AND.PID(IP-1).EQ.'CH  ') THEN
        IF(ICH.EQ.NLL(2,IP-1)) THEN
          ICH=0
          GO TO 20
        ENDIF
        PID(IP)='CH  '
        DO 15 NL=1,NLLI1
15        NLL(NL,IP)=NLL(NL,IP-1)
        LN(IP)=LN(IP-1)
        ICH=ICH+1
        GO TO 100
      ENDIF
C----------FIRST CARD FOR EACH PARAMETER
20    IF(NLLI1.LE.11) THEN
        READ(IU,525) PID(IP),LN(IP),(NLL(LL,IP),LL=1,NLLI1)
      ELSE
        READ(IU,525) PID(IP),LN(IP),(NLL(LL,IP),LL=1,11)
        READ(IU,520) (NLL(LL,IP),LL=12,NLLI1)
      ENDIF
      PIDTMP=PID(IP)
      WRITE(IOUT,532)IP,PIDTMP,LN(IP),(NLL(LL,IP),LL=1,NLLI1)
C----------CHECK PARAMETER IDENTIFIER
      INAME=0
      IF(PIDTMP.EQ.'T   '.OR.PIDTMP.EQ.'S1  '.OR.
     1 PIDTMP.EQ.'RCH '.OR.PIDTMP.EQ.'ETM '.OR.PIDTMP.EQ.'GHB '
     2 .OR.PIDTMP.EQ.'Q   '.OR.PIDTMP.EQ.'KV  '.OR.
     3 PIDTMP.EQ.'ANI '.OR.PIDTMP.EQ.'CH  '.OR.PIDTMP.EQ.'KDR '
     4 .OR.PIDTMP.EQ.'KRB '.OR.PIDTMP.EQ.'KST '.OR.
     5 PIDTMP.EQ.'TKV '.OR.PIDTMP.EQ.'ANIV') INAME=1
      IF(INAME.EQ.0) THEN
        WRITE(IOUT,534) PIDTMP,IP
        STOP
      ENDIF
C----------CHECK USE OF NATURAL LOG OF PARAMETER
      IF(LN(IP).NE.0.AND.(PIDTMP.EQ.'RCH '.OR.PIDTMP.EQ.'Q
     1       .OR.PIDTMP.EQ.'CH  '.OR.PIDTMP.EQ.'ETM ')) THEN
        WRITE(IOUT,528)
        IERR=1
      ENDIF
C----------READ AND CHECK FOR PARAMETERS DEFINED BY LISTS OF CELLS
      IF(PIDTMP.EQ.'Q   '.OR.PIDTMP.EQ.'GHB '.OR.PIDTMP.EQ.'KRB '.OR.
     1  PIDTMP.EQ.'KDR '.OR.PIDTMP.EQ.'KST ')THEN
        NL1=NLL(1,IP)
        NLLA=IABS(NL1)
        IF((NLLA.EQ.0).OR.
     1    (PIDTMP.EQ.'Q   '.AND.NLLA.GT.MXWELL).OR.
     2    (PIDTMP.EQ.'GHB '.AND.NLLA.GT.MXBND).OR.
     3    (PIDTMP.EQ.'KDR '.AND.NLLA.GT.MXDRN).OR.
     4    (PIDTMP.EQ.'KRB '.AND.NLLA.GT.MXRIVR).OR.
     5    (PIDTMP.EQ.'KST '.AND.NLLA.GT.MXSTRM)) THEN
          WRITE(IOUT,536) NL1
          IERR=1
        ENDIF
        IF(NLLI1.LT.3) THEN
          WRITE(IOUT,539)
          IERR=1
        ENDIF
        IF(PIDTMP.EQ.'Q   '.AND.IUNIT(2).EQ.0.OR.
     1     PIDTMP.EQ.'GHB '.AND.IUNIT(7).EQ.0.OR.
     1     PIDTMP.EQ.'KDR '.AND.IUNIT(3).EQ.0.OR.
     1     PIDTMP.EQ.'KRB '.AND.IUNIT(4).EQ.0) THEN
           WRITE(IOUT,538)
           IERR=1
        ENDIF
        IF(PIDTMP.NE.'KST ') WRITE(IOUT,565)
        IF(PIDTMP.EQ.'KST ') WRITE(IOUT,560)
        FAC=1.
        DO 30 J=1,NLLA
          NS=NS+1
          IF(PIDTMP.NE.'Q   '.AND.PIDTMP.NE.'KST ')
     1           READ(IU,504) (CELS(I,NS),I=1,4)
          IF(PIDTMP.EQ.'KST ') THEN
            READ(IU,501) (CELS(I,NS),I=1,2),CELS(4,NS)
            CELS(3,NS)=0.
          ENDIF
          IF(PIDTMP.EQ.'Q   ') READ(IU,503) (CELS(I,NS),I=1,4)
          IF(NL1.GT.0) WRITE(IOUT,508)(CELS(I,NS),I=1,4)
          IF(NL1.LT.0) WRITE(IOUT,508)(CELS(I,NS),I=1,3),FAC
30      CONTINUE
      ENDIF
C-------------CH
      IF(PIDTMP.EQ.'CH  ') THEN
        NL1=NLL(1,IP)
```

```
          IF(NL1.LE.0) THEN
            WRITE(IOUT,536) NL1
            STOP
          ENDIF
          ICH=1
C---------------READ IBOUND ARRAY(S)
   33     IF(IUBD.NE.0.AND.IFL.EQ.0) THEN
            REWIND(IUBD)
            IFL=1
            DO 35 K=1,NLAY
   35       CALL U2DINN(IBOUND(1,1,K),NROW,NCOL,IUBD)
            IF(NPER.GT.1) THEN
              DO 40 K=1,NLAY
                DO 38 I=1,NROW
                DO 38 J=1,NCOL
   38             BUFF(J,I,K)=IBOUND(J,I,K)
   40           CALL U2DINN(IBOUND(1,1,K),NROW,NCOL,IUBD)
            ENDIF
          ENDIF
          WRITE(IOUT,565)
          NPCH=0
          RPCH=0.
          NL3=NLL(3,IP)
          DO 50 IICH=1,NL1
            NS=NS+1
            READ(IU,503) DUM,(CELS(I,NS),I=1,3)
            DUM=0.
            WRITE(IOUT,508) DUM,(CELS(I,NS),I=1,3)
            CE3=CELS(3,NS)
            IF(CE3.GT.1..AND.CE3.NE.RPCH) THEN
              NPCH=NPCH+1
              RPCH=CE3
            ENDIF
            IF(CE3.GT.0..AND.CE3.LE.1.) THEN
              WRITE(IOUT,540)
              IERR=1
            ENDIF
            DO 45 NL=4,NLLI1
              K=NLL(NL,IP)
              IF(K.EQ.0) GO TO 50
              IF((IUBD.EQ.0.OR.NL3.NE.0.OR.NPER.EQ.1).AND.
     1          IBOUND(INT(CELS(2,NS)),INT(CELS(1,NS)),K).GT.0) THEN
                WRITE(IOUT,570)
                IERR=1
              ENDIF
              IF(IUBD.NE.0.AND.NL3.NE.1.AND.NPER.GT.1.AND.
     1          INT(BUFF(INT(CELS(2,NS)),INT(CELS(1,NS)),K)).GE.0)
     2          THEN
                WRITE(IOUT,570)
                IERR=1
              ENDIF
   45       CONTINUE
   50     CONTINUE
          IF(NPCH.NE.NLL(2,IP)) THEN
            WRITE(IOUT,527) NPCH,NLL(2,IP)
            IERR=1
          ENDIF
          IF(IUBD.NE.0) REWIND(IUBD)
C-------CHECK TO BE SURE THAT THE FIRST AND LAST FACTORS ARE NOT ZERO.
          IF(CELS(3,(NS-NL1+1)).EQ.0..OR.CE3.EQ.0.) THEN
            WRITE(IOUT,590)
            IERR=1
          ENDIF
C-------------CALCULATE COEFFICIENTS FOR CONSTANT-HEAD BOUNDARIES
          CALL SSEN1B(NLL(1,IP),NCOL,NROW,CELS,NS,NSN,DELR,
     1         DELC,IOUT,IERR,NLLI1)
        ENDIF
C----------READ AND CHECK FOR PARAMETERS DEFINED BY ONE NCOL X NROW ARRA
        IF(PIDTMP.EQ.'RCH '.OR.PIDTMP.EQ.'ETM ')THEN
          NL1=NLL(1,IP)
          IF(NLLI1.LT.3) THEN
            WRITE(IOUT,539)
            IERR=1
          ENDIF
          IF(NL1.EQ.0) THEN
            IERR=1
            WRITE(IOUT,536) NL1
          ELSEIF (NL1.GT.0) THEN
            NM=NM+1
            IF(LZI1.LE.14) THEN
              READ(IU,510) SFAC(NM),LM(NM),(LZ(IZ,NM),IZ=1,LZI1)
            ELSE
              READ(IU,510) SFAC(NM),LM(NM),(LZ(IZ,NM),IZ=1,14)
```

```
            READ(IU,520) (LZ(IZ,NM),IZ=15,LZI1)
          ENDIF
          WRITE(IOUT,530) SFAC(NM),LM(NM),LZ(1,NM)
          WRITE(IOUT,531) (LZ(IZ,NM),IZ=2,LZI1)
          IF(LZ(1,NM).NE.0.AND.(LZI1.LE.1.OR.NZM.LT.LZ(1,NM))) THEN
            WRITE(IOUT,600)
            IERR=1
          ENDIF
          IF(NMM.LT.LM(NM)) THEN
            WRITE(IOUT,630)
            IERR=1
          ENDIF
        ENDIF
        IF(PIDTMP.EQ.'RCH '.AND.IUNIT(8).EQ.0.OR.
     1     PIDTMP.EQ.'ETM '.AND.IUNIT(5).EQ.0) THEN
           WRITE(IOUT,538)
           IERR=1
        ENDIF
      ENDIF
C----------READ AND CHECK FOR PARAMETERS WHICH MAY APPLY TO MANY LAYERS
      IF(PIDTMP.EQ.'S1  '.OR.PIDTMP.EQ.'T   '.OR.PIDTMP.EQ.'KV  '
     1    .OR.PIDTMP.EQ.'TKV ') THEN
        DO 70 LL=1,NLLI1
          NL=NLL(LL,IP)
          NLLA=IABS(NL)
          IF(NL.EQ.0) GO TO 70

          IF(NLLA.GT.NLAY. OR .
     1      (PIDTMP.EQ.'KV  '.AND.NLAY.EQ.1).OR.
     1      (PIDTMP.EQ.'TKV '.AND.MOD(LL+1,3).NE.0.AND.
     1      (NLAY.EQ.1.OR.(MOD(LL+2,3).EQ.0.AND.NLLA.EQ.1)))) THEN
            WRITE(IOUT,533) NL
            IERR=1
          ENDIF
          IF(PIDTMP.EQ.'TKV '.AND.MOD(LL+2,3).EQ.0) NLLA=NLLA-1
          IF(NL.GT.0) THEN
            NM=NM+1
            IF(LZI1.LE.14) THEN
              READ(IU,510) SFAC(NM),LM(NM),(LZ(IZ,NM),IZ=1,LZI1)
            ELSE
              READ(IU,510) SFAC(NM),LM(NM),(LZ(IZ,NM),IZ=1,14)
              READ(IU,520) (LZ(IZ,NM),IZ=15,LZI1)
            ENDIF
            WRITE(IOUT,530) SFAC(NM),LM(NM),LZ(1,NM)
            WRITE(IOUT,531) (LZ(IZ,NM),IZ=2,LZI1)
            IF(LZ(1,NM).NE.0.AND.(LZI1.LE.1.OR.NZM.LT.LZ(1,NM))) THEN
              WRITE(IOUT,600)
              IERR=1
            ENDIF
            IF(NMM.LT.LM(NM)) THEN
              WRITE(IOUT,630)
              IERR=1
            ENDIF
          ENDIF
   70   CONTINUE
      ENDIF
C----------READ AND CHECK FOR PARAMETER WITH NO ADDITIONAL DATA
      IF(PIDTMP.EQ.'ANI '.OR.PIDTMP.EQ.'ANIV') THEN
        DO 75 LL=1,NLLI1
          IF(NLL(LL,IP).EQ.0) GO TO 77
          IF(NLL(LL,IP).GT.NLAY.OR.NLL(LL,IP).LT.0) THEN
            WRITE(IOUT,533) NLL(LL,IP)
            IERR=1
          ENDIF
   75   CONTINUE
   77   CONTINUE
      ENDIF
      IF(PIDTMP.EQ.'ANIV') THEN
        I1=0
        I2=0
        DO 90 LL=1,NLLI1
          NLP=NLL(LL,IP)
          IF(NLP.EQ.0) GO TO 95
          IF(NLP.GT.0) THEN
            I1=I1+1
            DO 85 IIP=1,IP-1
              IF(PID(IIP).EQ.'TKV ') THEN
                DO 80 LV=1,NLLI1
                  IF(MOD(LV,2).NE.0.AND.
     1               NLP.EQ.ABS(NLL(LV,IIP))) I2=I2+1
   80           CONTINUE
              ENDIF
              IF(PID(IIP).EQ.'KV  ') THEN
```

```
                DO 82 LV=1,NLLI1
                  IF(ABS(NLL(LV,IIP)).EQ.NLP) THEN
                    WRITE(IOUT,610) IIP
                    IERR=1
                  ENDIF
   82           CONTINUE
               ENDIF
   85        CONTINUE
           ENDIF
   90    CONTINUE
         IF(I1.NE.I2) THEN
           WRITE(IOUT,620)
           IERR=1
         ENDIF
   95    CONTINUE
       ENDIF
  100 CONTINUE
C-----READ MULTIPLICATION AND ZONE MATRICES
      IF(NMM.GT.0) THEN
        DO 110 M=1,NMM
  110     CALL U2DREL(SMAT(1,1,M),ANAME(1,1),NROW,NCOL,-M,IU,IOUT)
      ENDIF
      IF(NZM.GT.0) THEN
        DO 120 MZ=1,NZM
  120     CALL U2DINT(MATZ(1,1,MZ),ANAME(1,2),NROW,NCOL,-MZ,IU,IOUT)
      ENDIF
C----------CHECK WHETHER ITEMS STORED EXCEED RESERVED STORAGE SPACE
      IF(NS.GT.NSN.OR.NM.GT.NSM) THEN
        WRITE(IOUT,550) NS,NSN,NM,NSM
        IERR=1
      ELSE
        IF(NSN.GT.NS) THEN
          WRITE(IOUT,580) NSN,NS
          NSN=NS
        ENDIF
        IF(NSM.GT.NM) THEN
          WRITE(IOUT,585) NSM,NM
          NSM=NM
        ENDIF
      ENDIF
      RETURN
      END


C=======================================================================
      SUBROUTINE SSEN1H (
     1    NCOL,NROW,NLAY,NPER,IU,IOUT,DID,NH,NDER,JT,IBOUND,JOFF,IOFF,
     2    HOBS,WT,DELR,DELC,RINT,COFF,ROFF,MLAY,PR,MOBS,IERR,
     3    TOFF,EVH,ISN,MAXM)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     READ, CHECK AND STORE DATA FOR HYDRAULIC HEAD LOCATIONS.
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4  DID(NH)
      DIMENSION NDER(5,NH),IBOUND(NCOL,NROW,NLAY),IOFF(NH),JOFF(NH),
     2     HOBS(NH),WT(NH),DELR(NCOL),DELC(NROW),RINT(4,NH),
     4     COFF(NH),ROFF(NH),MLAY(MAXM,MOBS),PR(MAXM,MOBS),TOFF(NH)
C     ------------------------------------------------------------------
506   FORMAT(A4,1X,4I5,3F8.0,F10.0,F8.0,I5)
509   FORMAT(8(I5,F5.0))
515   FORMAT(A4,1X,I5,F8.0,3F10.0,I5)
520   FORMAT(2X,'TRANSIENT DATA AT THIS LOCATION, ITT =',I4)
525   FORMAT(1X,I5,2X,A4,21X,I5,19X,F8.2,4X,2G12.5)
540   FORMAT(/,' STATISTIC RELATED TO WEIGHT < 0 -- STOP EXECUTION',
     1 ' (SSEN1H)',/)
555   FORMAT(//,' OBSERVED HEAD DATA',//,3X,'OBS#  ID   LAYER ROW',
     P ' COLUMN   TIME STEP  ROW/COLUMN/TIME OFFSETS  OBSERVATION ',
     P ' VARIANCE')
560   FORMAT(1X,I5,2X,A4,I5,1X,2I5,5X,I5,3X,3F8.2,4X,2G12.5)
562   FORMAT(5X,'MULTIPLE LAYERS AND PROPORTIONS :',5(I5,',',F5.2,3X))
570   FORMAT(' TIME STEP (',I5,') LARGER THAN NPER (',I5,
     1  ') OF BASIC PACKAGE INPUT FILE -- STOP EXECUTION (SSEN1H)',/)
587   FORMAT(' ROW OR COLUMN NUMBER INVALID -- STOP EXECUTION',
     1      ' (SSEN1H)',/)
590   FORMAT(' LAYER INVALID OR CELL DESIGNATED AS INACTIVE IN '
     1 ,'IBOUND ARRAY OF BASIC PACKAGE INPUT -- STOP EXECUTION',
     2  ' (SSEN1H)',/)
592   FORMAT(/,' MULTILAYER PROPORTIONS DO NOT SUM ',
     1  'TO 1.0 -- STOP EXECUTION (SSEN1H)',/)
593   FORMAT(/,' NUMBER OF MULTILAYER OBSERVATIONS EXCEEDS MOBS -- '
     1 ,'STOP EXECUTION (SSEN1H)',/)
```

```
600   FORMAT(' ')
610   FORMAT(' ITT MUST = 1 OR 2 -- STOP EXECUTION (SSEN1H)',/)
C
C-------WRITE INTRODUCTORY LINES
      WRITE(IOUT,555)
C-------INITIALIZE VARIABLES
      JT=0
      IDRY=-1
      ML=0
      NT=0
      NTC=0
      IF(MAXM.EQ.0) THEN
        DO 20 MM=1,MOBS
        DO 20 M=1,MAXM
          MLAY(M,MM)=0
          PR(M,MM)=0.
   20   CONTINUE
      ENDIF
      DO 50 N=1,NH
   50 NDER(5,N)=0
C-------LOOP THROUGH HEAD OBSERVATIONS
      DO 110 N=1,NH
C----------READ FIRST TRANSIENT OBSERVATIONS AT ONE LOCATION
        IF(N.GT.1.AND.(NDER(4,N-1).LT.0.OR.NTC.LT.NT)) THEN
          IF(NDER(4,N-1).LT.0) THEN
            N1=N-1
            NT=-NDER(4,N1)
            NTC=1
            READ(IU,515) DID(N1),NDER(4,N1),TOFF(N1),HOBS(N1),WT(N1),
     1                   DUM,IWT
            IF(IWT.EQ.2) WT(N1)=WT(N1)*HOBS(N1)
            IF(IWT.GT.0) WT(N1)=WT(N1)*WT(N1)
            WT(N1)=WT(N1)*EVH
            WRITE(IOUT,525) N1,DID(N1),NDER(4,N1),TOFF(N1),
     1                     HOBS(N1),WT(N1)
            IF(WT(N1).LT.0.) THEN
              WRITE(IOUT,540)
              IERR=1
            ENDIF
          ENDIF
C-------SUBSEQUENT OBSERVATIONS AT ONE LOCATION
          NTC=NTC+1
C----------ASSIGN INFORMATION WHICH STAYS THE SAME
          DO 80 I=1,3
   80     NDER(I,N)=NDER(I,N1)
          ROFF(N)=ROFF(N1)
          COFF(N)=COFF(N1)
          IOFF(N)=IOFF(N1)
          JOFF(N)=JOFF(N1)
          DO 90 I=1,4
   90     RINT(I,N)=RINT(I,N1)
          IF(NDER(1,N1).LT.0) THEN
            ML1=ML-1
            ML=ML+1
            DO 95 M=1,MAXM
            PR(M,ML)=PR(N,ML1)
   95       MLAY(M,ML)=MLAY(M,ML1)
          ENDIF
C----------READ INFORMATION UNIQUE TO THIS OBSERVATION
          READ(IU,515) DID(N),NDER(4,N),TOFF(N),HOBS(N),WT1,WT2,IWT
          IF(ITT.EQ.1) WT(N)=WT1
          IF(ITT.EQ.2) THEN
            WT(N)=WT2
            HOBS(N)=HOBS(N)-HOBS(N1)
            NDER(5,N)=N1
          ENDIF
          IF(IWT.EQ.2) WT(N)=WT(N)*HOBS(N)
          IF(IWT.GT.0) WT(N)=WT(N)*WT(N)
          WT(N)=WT(N)*EVH
          WRITE(IOUT,525) N,DID(N),NDER(4,N),TOFF(N),HOBS(N),WT(N)
          IF(WT(N).LT.0.) THEN
            WRITE(IOUT,540)
            IERR=1
          ENDIF
          IF(NTC.EQ.NT) WRITE(IOUT,600)
          GO TO 107
        ENDIF
C----------READ A DATA SET 6
        READ(IU,506) DID(N),(NDER(I,N),I=1,4),
     1             ROFF(N),COFF(N),TOFF(N),HOBS(N),WT(N),IWT
        IF(IWT.EQ.2) WT(N)=WT(N)*HOBS(N)
        IF(IWT.GT.0) WT(N)=WT(N)*WT(N)
        WT(N)=WT(N)*EVH
```

```
        IF(NDER(4,N).LT.0) THEN
          READ(IU,509) ITT
          WRITE(IOUT,520) ITT
          IF(ITT.NE.1.AND.ITT.NE.2) THEN
            WRITE(IOUT,610)
            STOP
          ENDIF
        ENDIF
        WRITE(IOUT,560) N,DID(N),(NDER(I,N),I=1,4),
     P              ROFF(N),COFF(N),TOFF(N),HOBS(N),WT(N)
        IF(WT(N).LT.0.) THEN
          WRITE(IOUT,540)
          IERR=1
        ENDIF
C-------ERROR CHECKING
        K=NDER(1,N)
        I=NDER(2,N)
        J=NDER(3,N)
        IF(J.LE.0.OR.J.GT.NCOL.OR.I.LE.0.OR.I.GT.NROW)THEN
          WRITE(IOUT,587)
          IERR=1
        ENDIF
C-------INITIALIZE SOME VARIABLES
        MM=1
        TPR=1.
C-------READ INFORMATION FOR MULTILAYER OBSERVATIONS (DATA SET 6A)
        IF(K.LT.0) THEN
          ML=ML+1
          READ(IU,509)(MLAY(M,ML),PR(M,ML),M=1,-K)
          WRITE(IOUT,562)(MLAY(M,ML),PR(M,ML),M=1,-K)
          MM=-K
          TPR=0.
        ENDIF
C-------ERROR CHECKING
        DO 105 M=1,MM
          KK=K
C----------ASSIGN LAYER NUMBERS AND ADD PROPORTIONS FOR MULTILAYER
C----------OBSERVATION WELLS
          IF(K.LT.0) THEN
            KK=MLAY(M,ML)
            IF(KK.EQ.0) GO TO 105
            TPR=TPR+PR(M,ML)
          ENDIF
C----------CHECK LAYER NUMBER AND WHETHER THE CELL IS ACTIVE
          IF(KK.LE.0.OR.KK.GT.NLAY.OR.IBOUND(J,I,KK).EQ.0)THEN
            WRITE(IOUT,590)
            IERR=1
          ENDIF
  105   CONTINUE
C---------CHECK SUM OF PROPORTIONS FOR MULTILAYER OBS WELLS
        IF(K.LT.0.AND.(TPR.LT..98.OR.TPR.GT.1.02)) THEN
          WRITE(IOUT,592)
          IERR=1
        ENDIF
C-------CALCULATE INTERPOLATION COEFFICIENTS
        CALL SSEN1I(NDER(1,N),COFF(N),ROFF(N),DELR,DELC,IBOUND,NCOL,
     1       NROW,NLAY,RINT(1,N),JOFF(N),IOFF(N),MLAY(1,ML))
C-------KEEP TRACK OF LATEST MEASUREMENT
  107   IF(NDER(4,N).GT.JT) THEN
          JT=NDER(4,N)
          IF(TOFF(N).GT.0.) JT=JT+1
        ENDIF
C-------ERROR CHECKING - MAKE SURE LAST MEASUREMENT IS NOT TOO LATE
        IF(JT.GT.NPER) THEN
          WRITE(IOUT,570) JT,NPER
          IERR=1
          JT=NPER
        ENDIF
  110 CONTINUE
C-------ERROR CHECKING -
C-------EXCEEDED STORAGE RESERVED FOR MULTILAYER OBSERVATIONS?
      IF(ML.GT.MOBS) THEN
        WRITE(IOUT,593)
        IERR=1
      ENDIF
      RETURN
      END
```

```
C=========================================================================
      SUBROUTINE SSEN1I(
     1                  NDER,COFF,ROFF,DELR,DELC,IBOUND,
     2                  NCOL,NROW,NLAY,RINT,JOFF,IOFF,MLAY)
C-----VERSION 1000 01FEB1992
C     ********************************************************************
C     CALCULATE INTERPOLATION COEFFICIENTS FOR LOCATING OBSERVED HEADS.
C     ********************************************************************
C        SPECIFICATIONS:
C     --------------------------------------------------------------------
      DIMENSION NDER(5),DELR(NCOL),DELC(NROW),IBOUND(NCOL,NROW,NLAY),
     1       RINT(4)
C     --------------------------------------------------------------------
C
      K=NDER(1)
      IF(K.LT.0) K=MLAY
      I=NDER(2)
      J=NDER(3)
      I1=I+1
      J1=J+1
      IOFF=1
      JOFF=1
      IF(ROFF.LT.0.) THEN
        I1=I-1
        IOFF=-1
      ENDIF
      IF(COFF.LT.0.) THEN
        J1=J-1
        JOFF=-1
      ENDIF
      IF(I1.LT.1.OR.I1.GT.NROW) ROFF=0.
      IF(J1.LT.1.OR.J1.GT.NCOL) COFF=0.
C
      IF((ABS(ROFF).LT..001.AND.ABS(COFF).LT..001).OR.
     1   (ABS(ROFF).LT..001.AND.IBOUND(J1,I,K).EQ.0).OR.
     1   (ABS(COFF).LT..001.AND.IBOUND(J,I1,K).EQ.0).OR.
     1   (IBOUND(J,I1,K).EQ.0.AND.IBOUND(J1,I,K).EQ.0)) THEN
         IOFF=0
         JOFF=0
         DO 100 IR=1,4
  100    RINT(IR)=.25
        RETURN
      ENDIF
C
C---CALCULATE CONSTANTS
      IF(ABS(ROFF).GT..001) THEN
        DC=(DELC(I)+DELC(I1))/2.
        DCF=ABS(ROFF)*DELC(I)
      ENDIF
      IF(ABS(COFF).GT..001) THEN
        DR=(DELR(J)+DELR(J1))/2.
        DRF=ABS(COFF)*DELR(J)
      ENDIF
      IF(ABS(ROFF).GT..001.AND.ABS(COFF).GT..001) A=1/(DC*DR)
C
C---LINEAR INTERPOLATION
      IF(ABS(ROFF).LT..001.OR.
     1  (IBOUND(J,I1,K).EQ.0.AND.IBOUND(J1,I1,K).EQ.0)) THEN
        IOFF=0
        RINT(1)=0.5*(1.-DRF/DR)
        RINT(2)=0.5*DRF/DR
        RINT(3)=RINT(1)
        RINT(4)=RINT(2)
        RETURN
      ENDIF
C
      IF(ABS(COFF).LT..001.OR.
     1  (IBOUND(J1,I,K).EQ.0.AND.IBOUND(J1,I1,K).EQ.0)) THEN
        JOFF=0
        RINT(1)=0.5*(1.-DCF/DC)
        RINT(2)=RINT(1)
        RINT(3)=0.5*DCF/DC
        RINT(4)=RINT(3)
        RETURN
      ENDIF
C
C---CALCULATE BASIS FUNCTIONS FOR INTERPOLATION ON A RECTANGLE
      IF(IBOUND(J1,I,K).NE.0.AND.IBOUND(J,I1,K).NE.0.AND.
     1   IBOUND(J1,I1,K).NE.0) THEN
        RINT(3)=A*(DR-DRF)*DCF
        RINT(4)=A*DRF*DCF
        RINT(2)=A*DRF*(DC-DCF)
        RINT(1)=A*(DR-DRF)*(DC-DCF)
```

```
        RETURN
      ENDIF
C
C---CALCULATE BASIS FUNCTIONS FOR INTERPOLATION ON A TRIANGLE
      IF(IBOUND(J1,I,K).EQ.0) THEN
        RINT(1)=A*(DR*DC - DR*DCF)
        RINT(2)=0.
        RINT(3)=A*(DR*DCF - DC*DRF)
        RINT(4)=A*(DC*DRF)
        RETURN
      ENDIF
C
      IF(IBOUND(J,I1,K).EQ.0) THEN
        RINT(1)=A*(DR*DC - DC*DRF)
        RINT(4)=A*(DR*DCF)
        RINT(2)=A*(DC*DRF - DR*DCF)
        RINT(3)=0.
        RETURN
      ENDIF
C
      IF(IBOUND(J1,I1,K).EQ.0) THEN
        RINT(1)=A*(DR*DC - DC*DRF - DR*DCF)
        RINT(3)=A*(DR*DCF)
        RINT(2)=A*(DC*DRF)
        RINT(4)=0.
        RETURN
      ENDIF
      END


C======================================================================
      SUBROUTINE SSEN1J(
     1                 IIP,B,NLL,CELS,NS,NSN,HNEW,NCOL,NROW,NLAY,
     2                 NP,IP,KPER,ICH,NLLI1)
C-----VERSION 1000 01FEB1992
C     *****************************************************************
C     FOR PID=CH : PUT NEW CONSTANT HEAD VALUES IN HEAD ARRAY
C     *****************************************************************
C        SPECIFICATIONS:
C     -----------------------------------------------------------------
      DOUBLE PRECISION HNEW(NCOL,NROW,NLAY)
      DIMENSION B(NP),CELS(4,NSN),NLL(NLLI1)
C     -----------------------------------------------------------------
C
      NL1=NLL(1)
      NL2=NLL(2)
C-------DO NOT DO CALCULATIONS IF THIS ISN'T THE FIRST CALL FOR THIS
C-------CONSTANT-HEAD SEQUENCE.
      IF(ICH.NE.0.OR.IP.GT.0) THEN
        IF(IP.GT.0.AND.ICH.EQ.0) NS=NS+NL1
        ICH=ICH+1
        IF(ICH.EQ.NL2) ICH=0
        RETURN
      ENDIF
C-------CHECK TIMING
      IT=1
      NL3=NLL(3)
      IF((NL3.EQ.1.AND.KPER.NE.2).OR.
     1 ((NL3.EQ.0.OR.NL3.EQ.2).AND.KPER.NE.1)) THEN
        IF(ICH.EQ.0) NS=NS+NL1
        ICH=ICH+1
        IF(ICH.EQ.NL2) ICH=0
        RETURN
      ENDIF
C-------INITIALIZE VARIABLES
      II=0
      N1=NS+1
      NS=NS+NL1
      IICH=0
C-------GET LIMIT OF BACKWARD INTERP. (I) AND NEXT END-POINT
C-------LOCATION (INCH)
      I1=N1
   15 I=0
      DO 20 INCH=I1,NS
        CE=CELS(4,INCH)
        IF(INCH.GT.N1) THEN
          CEM1=CELS(4,INCH-1)
          IF((CE.GT.0..AND.CE.LE.1.).AND.(CEM1.LT.0..OR.CEM1.GE.1.))
     1       I=INCH
        ENDIF
        IF(CE.GT.1..OR.CE.LT.0.) GO TO 21
   20 CONTINUE
   21 CONTINUE
```

```
C-------SAVE ROW AND COLUMN OF END POINT
      JC1=INT(CELS(2,INCH))
      IC1=INT(CELS(1,INCH))
C-------PUT PARAMETER VALUE IN HNEW
      IF(CE.GT.1.) THEN
        II=II+1
        DO 25 NL=4,NLLI1
          K=NLL(NL)
          IF(K.EQ.0) GO TO 26
          HNEW(JC1,IC1,K)=B(IIP-1+II)
   25   CONTINUE
   26   CONTINUE
      ENDIF
C-------INTERPOLATE BACKWARDS IF THERE ARE NONZERO COEFFICIENTS
      IF(I.NE.0.AND.I.LT.INCH) THEN
        JC2=INT(CELS(2,I-1))
        IC2=INT(CELS(1,I-1))
        DO 30 N=I,INCH-1
          P=CELS(4,N)
          JC=INT(CELS(2,N))
          IC=INT(CELS(1,N))
          DO 29 NL=4,NLLI1
            K=NLL(NL)
            IF(K.EQ.0) GO TO 30
            HNEW(JC,IC,K)=P*HNEW(JC1,IC1,K)+(1.-P)*HNEW(JC2,IC2,K)
   29     CONTINUE
   30   CONTINUE
      ENDIF
C-------DOES THE HEAD APPLY DIRECTLY TO A SERIES OF CELLS?
      IF(INCH.EQ.NS) GO TO 80
      II1=INCH+1
      DO 50 INCH=II1,NS
        IF(CELS(4,INCH).NE.1.) GO TO 51
        JC=INT(CELS(2,INCH))
        IC=INT(CELS(1,INCH))
        DO 35 NL=4,NLLI1
          K=NLL(NL)
          IF(K.EQ.0) GO TO 50
          HNEW(JC,IC,K)=HNEW(JC1,IC1,K)
   35   CONTINUE
   50 CONTINUE
C-------CHECK FOR LAST CH PARAMETER OF THIS GROUP
   51 IF(INCH.EQ.NS.AND.II.EQ.NLL(2)) GO TO 80
      I1=INCH
      IICH=INCH-(NS-NL1)
      GO TO 15
   80 ICH=1
      IF(NLL(2).EQ.1) ICH=0
      RETURN
      END


C=======================================================================
      SUBROUTINE SSEN1K(
     1        PID,IIP,NLL,NNB,MXBN,BN,CELS,NSN,NS,NCOL,NROW,NLAY,I1BN,
     2        IBOUND,NPER,KPER,B,IP,ISN,ISTRM,NLLI1,NLLIT)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     FOR PUMPING AND HEAD-DEPENDANT BOUNDARIES : PUT NEW PARAMETER
C     VALUES IN MODEL ARRAYS
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 PID
      DIMENSION BN(I1BN,MXBN),CELS(4,NSN),ISTRM(5,MXBN),
     1     IBOUND(NCOL,NROW,NLAY),NLL(NLLI1)
C     ------------------------------------------------------------------
C
      NL1=NLL(1)
      NL1A=IABS(NL1)
      NS1=NS+1
      NS=NS+NL1A
      ITS=KPER-1
      IF(ISN.GT.0.AND.IP.EQ.1) ITS=NPER-KPER
C-------LOOP THROUGH TIMES.
      DO 300 KT=1,NLLIT
C----------ONLY DO CALCULATIONS IF THE VALUE HAS CHANGED.
        IF((ISN.LT.0.OR.IP.NE.1).AND.NLL(2*KT).NE.ITS) GO TO 300
        IF((ISN.GT.0.AND.IP.EQ.1).AND.NLL(2*KT+1).NE.ITS) GO TO 300
        DO 200 II=NS1,NS
          K=CELS(1,II)
          I=CELS(2,II)
          J=CELS(3,II)
```

```
        IF(PID.NE.'KST '.AND.IBOUND(J,I,K).LE.0) GO TO 200
        DO 100 JJ=1,NNB
          K1=BN(1,JJ)
          I1=BN(2,JJ)
          J1=BN(3,JJ)
          IF((PID.NE.'KST '.AND.
     1         K1.EQ.K .AND. I1.EQ.I .AND. J1.EQ.J)
     2         .OR.(PID.EQ.'KST '.AND.
     3         ISTRM(4,JJ).EQ.K .AND. ISTRM(5,JJ).EQ.I))THEN
            FACTOR=1.
            IF(NL1.GT.0) FACTOR=CELS(4,II)
            IF(PID.NE.'Q   '.AND.PID.NE.'KST ') BN(5,JJ)=FACTOR*B
            IF(PID.EQ.'KST ') BN(3,JJ)=FACTOR*B
            IF(PID.EQ.'Q   '.AND.IP.EQ.0) BN(4,JJ)=FACTOR*B
          ENDIF
  100   CONTINUE
  200 CONTINUE
  300 CONTINUE
      RETURN
      END


C=====================================================================
      SUBROUTINE SSEN1M(
     1       PID,C,SMAT,NCOL,NROW,NLAY,NSM,NM,NLL,B,KPER,IP,DELR,DELC,
     2       IBOUND,ISN,NMM,NZM,LZI1,SFAC,LZ,LM,MATZ,NLLI1,NLLIT)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     FOR PID=KV,S1,RCH,ETM,TKV : PUT NEW PARAMETER VALUES IN MODEL
C                                 ARRAYS
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 PID
      DIMENSION C(NCOL,NROW,NLAY),SMAT(NCOL,NROW,NMM),NLL(NLLI1),
     1          DELR(NCOL),DELC(NROW),IBOUND(NCOL,NROW,NLAY),LM(NSM),
     2          MATZ(NCOL,NROW,NZM),SFAC(NSM),LZ(LZI1,NSM)
C     ------------------------------------------------------------------
      NL2=NLLI1
      NT=1
      IF(PID.EQ.'RCH '.OR.PID.EQ.'ETM ') THEN
        NL2=1
        NT=NLLIT
        ITS=KPER-1
        IF(ISN.GT.0.AND.IP.EQ.1) ITS=NPER-KPER
      ENDIF
C-------LOOP THROUGH LAYERS
      DO 90 LL=1,NL2
        NLO=NLL(LL)
        IF(NLO.GT.0) NM=NM+1
        IF(PID.EQ.'TKV '.AND.MOD(LL+1,3).EQ.0) GO TO 90
C----------TIME INVARIANT PARAMETERS: FOR KPER>1, JUST KEEP TRACK OF NM.
        IF(PID.EQ.'KV  '.OR.PID.EQ.'TKV '.OR.PID.EQ.'S1  ') THEN
          IF(KPER.GT.1.OR.IP.GT.0) GO TO 90
          IF(NLO.EQ.0.AND.PID.EQ.'TKV ') GO TO 90
          IF(NLO.EQ.0) GO TO 95
        ENDIF
        NL=IABS(NLO)
        IF(PID.EQ.'RCH '.OR.PID.EQ.'ETM ') NL=1
C-------LOOP THROUGH TIMES FOR RCH AND ETM; NT=1 FOR OTHER PID'S
        DO 85 K=1,NT
C----------APPLICABLE THIS TIME STEP?
          IF((PID.EQ.'RCH '.OR.PID.EQ.'ETM ').AND.
     1      (ITS.GT.NLL(2*K+1).OR.ITS.LT.NLL(2*K))) GO TO 85
          IF(NLO.GT.0) THEN
            SF=SFAC(NM)*B
            M=LM(NM)
            LZ1=LZ(1,NM)
            LZN=2
            IF(LZ1.NE.0) LZN=LZI1
            DO 15 IZ=2,LZN
              IF(LZ1.NE.0)THEN
                NZ=LZ(IZ,NM)
                IF(NZ.EQ.0) GO TO 20
              ENDIF
              DO 10 I=1,NROW
              DO 10 J=1,NCOL
                CO=0.0
                IF(LZ1.EQ.0) THEN
                  IF(M.EQ.0) CO=SF*DELR(J)*DELC(I)
                  IF(M.NE.0.AND.SMAT(J,I,M).NE.0.)
     1              CO=SF*SMAT(J,I,M)*DELR(J)*DELC(I)
                ELSE
```

```
                  IF(MATZ(J,I,LZ1).EQ.NZ) THEN                                 SSJ0216
                    IF(M.EQ.0) CO=SF*DELR(J)*DELC(I)                           SSJ0217
                    IF(M.NE.0.AND.SMAT(J,I,M).NE.0.)                           SSJ0218
     1                CO=SF*SMAT(J,I,M)*DELR(J)*DELC(I)                        SSJ0219
                  ENDIF                                                        SSJ0220
                ENDIF                                                          SSJ0221
                IF(CO.NE.0..AND.(PID.EQ.'KV  '.OR.                             SSJ0222
     1             (PID.EQ.'TKV '.AND.MOD(LL+1,3).NE.0))) THEN                 SSJ0223
                  IF(PID.EQ.'KV  '.OR.MOD(LL,3).EQ.0) THEN                     SSJ0224
                    CNL=C(J,I,NL)                                              SSJ0225
                    IF(CNL.EQ.0.) C(J,I,NL)=CO                                 SSJ0226
                    IF(CNL.NE.0.) C(J,I,NL)=1./( (1./CNL) + 1./CO)             SSJ0227
                  ELSEIF (MOD(LL+2,3).EQ.0) THEN                               SSJ0228
                    CNLM=C(J,I,NL-1)                                           SSJ0229
                    IF(CNLM.EQ.0.) C(J,I,NL-1)=CO                              SSJ0230
                    IF(CNLM.NE.0.) C(J,I,NL-1)=1./( (1./CNLM) + 1./CO)         SSJ0231
                  ENDIF                                                        SSJ0232
                ELSEIF (CO.NE.0.) THEN                                         SSJ0233
                  C(J,I,NL)=C(J,I,NL)+CO                                       SSJ0234
                ENDIF                                                          SSJ0235
   10         CONTINUE                                                         SSJ0236
   15       CONTINUE                                                           SSJ0237
   20       CONTINUE                                                           SSJ0238
          ENDIF                                                                SSJ0239
          IF(NLO.LT.0) THEN                                                    SSJ0240
            DO 80 I=1,NROW                                                     SSJ0241
            DO 80 J=1,NCOL                                                     SSJ0242
              IF(IBOUND(J,I,NL).GT.0.OR.PID.EQ.'RCH '.OR.PID.EQ.'ETM ')        SSJ0243
     1          THEN                                                           SSJ0244
                IF((PID.EQ.'KV  '.OR.PID.EQ.'TKV ').AND.                       SSJ0245
     1              CO.NE.0..AND.(C(J,I,NL).NE.0..OR.                          SSJ0246
     1                (PID.EQ.'TKV '.AND.MOD(LL+2,3).EQ.0.AND.                 SSJ0247
     1                  C(J,I,NL-1).NE.0.))) THEN                              SSJ0248
                  IF(PID.EQ.'TKV '.AND.MOD(LL+2,3).EQ.0) THEN                  SSJ0249
                    C(J,I,NL-1)=1./                                            SSJ0250
     1                ((1./C(J,I,NL-1)) + 1./(B*DELR(J)*DELC(I)))              SSJ0251
                  ELSE                                                         SSJ0252
                    C(J,I,NL)=1./                                              SSJ0253
     1                ((1./C(J,I,NL)) + 1./(B*DELR(J)*DELC(I)))                SSJ0254
                  ENDIF                                                        SSJ0255
                ELSE                                                           SSJ0256
                  C(J,I,NL)=C(J,I,NL)+(B*DELR(J)*DELC(I))                      SSJ0257
                ENDIF                                                          SSJ0258
              ENDIF                                                            SSJ0259
   80       CONTINUE                                                           SSJ0260
          ENDIF                                                                SSJ0261
   85   CONTINUE                                                               SSJ0262
   90 CONTINUE                                                                 SSJ0263
   95 CONTINUE                                                                 SSJ0264
      RETURN                                                                   SSJ0265
      END                                                                      SSJ0266
C=======================================================================SSJ0267
      SUBROUTINE SSEN1N(                                                       SSJ0268
     1                 H,HOLD,A,NC,NR,NL,ST,DELR,DELC,IBOUND,TRPY,             SSJ0269
     2                 RHS,SHOLD,BOT,CR,CC,K,ISN,KB,X,NP)                      SSJ0270
C-----VERSION 1000 01FEB1992                                                   SSJ0271
C     ******************************************************************SSJ0272
C     ADD NONLINEAR TERMS FOR SENSITIVITY EQUATION CALCULATIONS                SSJ0273
C     ******************************************************************SSJ0274
C        SPECIFICATIONS:                                                       SSJ0275
C     ------------------------------------------------------------------SSJ0276
      DOUBLE PRECISION A(NC*NR*NL),AO,AP                                       SSJ0277
      DIMENSION H(NC*NR*NL),CR(NC,NR,NL),CC(NC,NR,NL),ST(NC,NR,NL),            SSJ0278
     1       DELR(NC),DELC(NR),RHS(NC,NR,NL),IBOUND(NC,NR,NL),TRPY(NL),SSJ0279
     2       HOLD(NC*NR*NL),SHOLD(NC*NR*NL),BOT(NC,NR,NL),X(NP,1)              SSJ0280
      COMMON /FLWCOM/LAYCON(80)                                                SSJ0281
C     ------------------------------------------------------------------SSJ0282
C                                                                              SSJ0283
      NRC = NR*NC                                                              SSJ0284
      NRCL = NR*NC*NL                                                          SSJ0285
      NR1=NR-1                                                                 SSJ0286
      NC1=NC-1                                                                 SSJ0287
C                                                                              SSJ0288
      YX=TRPY(K)*2.                                                            SSJ0289
      LT=LAYCON(K)                                                             SSJ0290
C                                                                              SSJ0291
C     CR                                                                       SSJ0292
C                                                                              SSJ0293
      IF(NC.GT.1) THEN                                                         SSJ0294
        DO 300 I=1,NR                                                          SSJ0295
        DO 300 J=1,NC1                                                         SSJ0296
          IF(IBOUND(J,I,K).EQ.0.OR.IBOUND(J+1,I,K).EQ.0) GO TO 300             SSJ0297
          IND=J+NC*(I-1)+NRC*(K-1)                                             SSJ0298
```

```
          IF(IBOUND(J,I,K).EQ.0.OR.IBOUND(J+1,I,K).EQ.0) GO TO 300
          IND=J+NC*(I-1)+NRC*(K-1)
          AO=A(IND)
          AP=A(IND+1)
          HO=H(IND)
          HP=H(IND+1)
          BO=BOT(J,I,KB)
          BP=BOT(J+1,I,KB)
          TH1=HO-BO
          TH2=HP-BP
C-------MATRIX DERIVATIVES
          D1CR=(CR(J,I,K)**2)*DELR(J)/
     1                       (2.*DELC(I)*ST(J,I,K)*(TH1**2))
          D2CR=(CR(J,I,K)**2)*DELR(J+1)/
     1                       (2.*DELC(I)*ST(J+1,I,K)*(TH2**2))
C-------MULTIPLY BY SENSITIVITIES FROM LAST ITERATION
          CO=0.
          CO=D1CR*AO+D2CR*AP
C-------MULTIPLY BY HEAD VECTOR AND ADD TO RHS
          RHS(J,I,K)=RHS(J,I,K)-CO*(HP-HO)
          RHS(J+1,I,K)=RHS(J+1,I,K)-CO*(HO-HP)
300     CONTINUE
       ENDIF
C
C      CC
C
       IF(NR.GT.1) THEN
         DO 330 J=1,NC
         DO 330 I=1,NR1
          IF(IBOUND(J,I,K).EQ.0.OR.IBOUND(J,I+1,K).EQ.0) GO TO 330
          IND=J+NC*(I-1)+NRC*(K-1)
          HO=H(IND)
          HP=H(IND+NC)
          AO=A(IND)
          AP=A(IND+NC)
          BO=BOT(J,I,KB)
          BP=BOT(J,I+1,KB)
          TH1=HO-BO
          TH2=HP-BP
C-------MATRIX DERIVATIVES
          D1CC=(CC(J,I,K)**2)*DELC(I)/
     1                       (2.*DELR(J)*ST(J,I,K)*(TH1**2))
          D2CC=(CC(J,I,K)**2)*DELC(I+1)/
     1                       (2.*DELR(J)*ST(J,I+1,K)*(TH2**2))
C-------MULTIPLY BY DERIVATIVES FROM LAST ITERATION
          CO=0.
          CO=D1CC*AO+D2CC*AP
C-------MULTIPLY BY HEAD VECTOR AND ADD TO RHS
          RHS(J,I,K)=RHS(J,I,K)-CO*(HP-HO)
          RHS(J,I+1,K)=RHS(J,I+1,K)-CO*(HO-HP)
  330   CONTINUE
       ENDIF
      RETURN
      END


C======================================================================
      SUBROUTINE SSEN1O(
     1       NP,B,B1,IWPG,PID,WP,IOUT,NH,H,HOBS,WT,DID,NDER,ROFF,COFF,
     2       TOFF,NQT,IQOB,ND,IPAR,NPR,MPR,PRM,RSQ,RSQP,IO,LN,RSQH,NPO)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     CALCULATE AND PRINT WEIGHTED RESIDUALS FOR DEPENDENT-VARIABLE
C     OBSERVATIONS, PRIOR PARAMETER ESTIMATES BY PARAMETER GROUPS,
C     AND PRIOR ESTIMATES OF PARAMETER SUMS.
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 PID(NP),DID(ND)
      DIMENSION B(NPO),IWPG(NP),B1(NPO),WP(NPO+MPR),H(ND),HOBS(ND),
     1          WT(ND),NDER(5,ND),ROFF(NH),COFF(NH),TOFF(NH),IQOB(NQT),
     2          PRM(NPO+1,MPR),LN(NP)
C     ------------------------------------------------------------------
C
  580 FORMAT(/,' PARAMETERS WITH PRIOR INFORMATION, BY GROUP',//,
     1       '  PARAMETER        VALUES                ',9X,'WEIGHT  ',
     1' WEIGHTED'
     1,/,    '   #   PID     MEAS.    CALC.     RESIDUAL      **.5  ',
     1'  RESIDUAL')
  585 FORMAT(I4,3X,A4,E10.3,1X,E10.3,1X,E10.3,1X,G10.3,1X,G10.3)
  620 FORMAT(/,' DATA FOR FLOWS',//,7X,
     1 '       TIME     MEAS.      CALC.',28X,'WEIGHTED',/,
     1 '    OBS#  ID   STEP      FLOW        ',
```

```
     1  'FLOW',6X,'RESIDUAL',3X,'WEIGHT**.5',2X,'RESIDUAL',/)
  625 FORMAT(1X,I5,3X,A4,I4,4X,5G11.3)
  630 FORMAT(1X,I5,3X,A4,I4,4X,G11.3,'   DISCONNECTED')
  635 FORMAT(/,' DATA AT HEAD LOCATIONS',/)
  637 FORMAT(28X,'TIME    ROW/COL/TIME        MEAS.',6X,
     1  'CALC.',24X,'WEIGHTED',/,
     1  ' OBS#  ID    LAYER,ROW,COL  STEP      OFFSETS',10X,'HEAD',7X,
     1  'HEAD',2X,'RESIDUAL',2X,'WEIGHT**.5',3X,'RESIDUAL',/)
  640 FORMAT(I5,1X,A4,3X,3I4,I6,3X,3F5.2,F11.3,F11.3,3G11.3)
  645 FORMAT(I5,1X,A4,3X,3I4,I6,3X,3F5.2,F11.3,'      DRY')
  650 FORMAT(/,' PARAMETER SUMS WITH PRIOR INFORMATION',//,
     1    19X, 'VALUES',24X,'WEIGHT  WEIGHTED',/,
     1    4X,'EQ.#',7X,'MEAS.',4X,'CALC.',5X,'RESIDUAL',8X,'**.5',3X,
     1    'RESIDUAL')
  660 FORMAT(I7,4X,E10.3,1X,E10.3,1X,E10.3,1X,G10.3,1X,G10.3)
  665 FORMAT(/,' SUM OF SQUARED WEIGHTED RESIDUALS (HEADS)  ',
     1    G11.5)
  670 FORMAT(/,' SUM OF SQUARED WEIGHTED RESIDUALS (ALL DEP. VAR.)  ',
     1    G11.5)
  675 FORMAT(' SUM OF SQUARED WEIGHTED RESIDUALS (WITH PARAMETERS) ',
     1    G11.5)
  680 FORMAT(' STATISTICS FOR THESE RESIDUALS :',/,
     2         ' MAXIMUM WEIGHTED RESIDUAL  :',E10.3,' OBS#',I7,/,
     3         ' MINIMUM WEIGHTED RESIDUAL  :',E10.3,' OBS#',I7,/,
     4         ' AVERAGE WEIGHTED RESIDUAL  :',E10.3,/,
     X         ' # RESIDUALS >= 0. :',I7,/,' # RESIDUALS < 0.  :',I7,/,
     5         ' NUMBER OF RUNS  :',I5,'  IN',I5,' OBSERVATIONS')
  685 FORMAT(' STATISTICS FOR ALL RESIDUALS :',/,
     1         ' AVERAGE WEIGHTED RESIDUAL  :',E10.3,/,
     2         ' # RESIDUALS >= 0. :',I7,/,' # RESIDUALS < 0.  :',I7,/,
     3         ' NUMBER OF RUNS  :',I5,'  IN',I5,' OBSERVATIONS')
  690 FORMAT(' RUNS STATISTIC (TOO FEW RUNS):',G13.3,/,
     1         '  (IF #NEG>10 AND #POS>10, P(STAT < -1.28) = 0.10,',/,
     2         '                           P(STAT < -1.645) = 0.05,',/,
     3         '                           P(STAT < -1.96) = 0.025)',/,
     4         ' RUNS STATISTIC (TOO MANY RUNS):',G13.3,/,
     5         '  (IF #NEG>10 AND #POS>10, P(STAT > 1.28) = 0.10,',/,
     6         '                           P(STAT > 1.645) = 0.05,',/,
     7         '                           P(STAT > 1.96) = 0.025)')
  695 FORMAT(' STATISTICS FOR THESE RESIDUALS :',/,
     1         ' MAXIMUM WEIGHTED RESIDUAL  :',E10.3,/,
     2         ' MINIMUM WEIGHTED RESIDUAL  :',E10.3,/,
     3         ' AVERAGE WEIGHTED RESIDUAL  :',E10.3,/,
     4         ' # RESIDUALS >= 0. :',I7,/,' # RESIDUALS < 0.  :',I7,/,
     5         ' NUMBER OF RUNS  :',I5,'  IN',I5,' OBSERVATIONS')
C
        RSQ=0.
        RSQP=0.
        NNEGT=0
        NPOST=0
        AVET=0.
        JDRY=0
        IDIS=0
        NRUNS=1
C--------------HEADS
      IF(NH.GT.0) THEN
        IF(IO.EQ.1) WRITE(IOUT,635)
        NNEG=0
        NPOS=0
        VMAX=-1.E20
        VMIN=1.E20
        AVE=0.
        IF(IO.EQ.1) WRITE(IOUT,637)
        DO 160 N=1,NH
          RES=HOBS(N)-H(N)
          W=WT(N)
          IF(W.LT.0.) THEN
            WRITE(IOUT,645) N,DID(N),(NDER(J,N),J=1,4),
     1              ROFF(N),COFF(N),TOFF(N)
            JDRY=JDRY+1
            GO TO 160
          ENDIF
          WT2=SQRT(W)
          WTR=RES*WT2
          IF(IO.EQ.1) WRITE(IOUT,640) N,DID(N),(NDER(J,N),J=1,4),
     1              ROFF(N),COFF(N),TOFF(N),HOBS(N),H(N),RES,WT2,WTR
          RSQ=RSQ+(WTR**2)
          IF(WTR.GT.VMAX) THEN
            VMAX=WTR
            NMAX=N
          ENDIF
          IF(WTR.LT.VMIN) THEN
            VMIN=WTR
```

```
            NMIN=N
          ENDIF
          IF(WTR.GE.0.) NPOS=NPOS+1
          IF(WTR.LT.0.) NNEG=NNEG+1
          IF(N.GT.1.AND.(WTRL*WTR).LT.0.) NRUNS=NRUNS+1
          WTRL=WTR
          AVE=AVE+WTR
  160   CONTINUE
        IF(JDRY.NE.NH) THEN
          AVET=AVET+AVE
          NPOST=NPOST+NPOS
          NNEGT=NNEGT+NNEG
          AVE=AVE/REAL(NH-JDRY)
          IF(IO.EQ.1) WRITE(IOUT,680) VMAX,NMAX,VMIN,NMIN,AVE,
     1                                NPOS,NNEG,NRUNS,NH
          IF(IO.EQ.1) WRITE(IOUT,665) RSQ
        ENDIF
      ENDIF
      RSQH=RSQ
C-----HEAD-DEPENDENT FLOWS
      IF(NQT.GT.0) THEN
        IF(IO.EQ.1) WRITE(IOUT,620)
        NNEG=0
        NPOS=0
        VMAX=-1.E20
        VMIN=1.E20
        AVE=0.
        NH1=NH+1
        NH2=NH+NQT
        DO 170 N=NH1,NH2
          RES=HOBS(N)-H(N)
          W=WT(N)
          IF(W.LT.0.) THEN
            WRITE(IOUT,630) N,DID(N),IQOB(N-NH),HOBS(N)
            IDIS=IDIS+1
            GO TO 170
          ENDIF
          WT2=SQRT(W)
          WTR=RES*WT2
          IF(IO.EQ.1) WRITE(IOUT,625) N,DID(N),IQOB(N-NH),HOBS(N),
     1                H(N),RES,WT2,WTR
          RSQ=RSQ+(WTR**2)
          IF(WTR.GT.VMAX) THEN
            VMAX=WTR
            NMAX=N
          ENDIF
          IF(WTR.LT.VMIN) THEN
            VMIN=WTR
            NMIN=N
          ENDIF
          IF(WTR.GE.0.) NPOS=NPOS+1
          IF(WTR.LT.0.) NNEG=NNEG+1
          IF(N.GT.1.AND.(WTRL*WTR).LT.0.) NRUNS=NRUNS+1
          WTRL=WTR
          AVE=AVE+WTR
  170   CONTINUE
        N=N-1
        IF(NQT.NE.IDIS) THEN
          AVET=AVET+AVE
          NPOST=NPOST+NPOS
          NNEGT=NNEGT+NNEG
          AVE=AVE/REAL(NQT-IDIS)
          IF(IO.EQ.1) WRITE(IOUT,680) VMAX,NMAX,VMIN,NMIN,AVE,
     1      NPOS,NNEG,NRUNS,N
          IF(IO.EQ.1) WRITE(IOUT,670) RSQ
        ENDIF
      ENDIF
C------PRINT WEIGHTED RESIDUALS FOR PRIOR INFORMATION ON INDIVIDUAL
C------PARAMETERS BY PARAMETER GROUP
      RSQP=RSQ
      IF(IPAR.GT.0.AND.NPR.GT.0) THEN
        IF(IO.EQ.1) WRITE(IOUT,580)
        NNEG=0
        NPOS=0
        VMAX=-1.E20
        VMIN=1.E20
        AVE=0.
        DO 327 IG=1,NP
          II=0
          DO 326 IP=1,NP
            IF(IWPG(IP).EQ.IG.AND.WP(IP).GT.0.) THEN
              II=1
              WPSR=WP(IP)**.5
```

```
            BB=B(IP)
            BB1=B1(IP)
            IF(LN(IP).NE.0) THEN
              BB=ALOG(B(IP))
              BB1=EXP(B1(IP))
            ENDIF
            BDIF=B1(IP)-BB
            BWP=BDIF*WPSR
            IF(IO.EQ.1) WRITE(IOUT,585) IP,PID(IP),BB1,B(IP),BDIF,
     1                                  WPSR,BWP
            RSQP=RSQP+(BWP**2)
            IF(BWP.GT.VMAX) VMAX=BWP
            IF(BWP.LT.VMIN) VMIN=BWP
            IF(BWP.GE.0.) NPOS=NPOS+1
            IF(BWP.LT.0.) NNEG=NNEG+1
            N=N+1
            IF(N.GT.1.AND.(WTRL*BWP).LT.0.) NRUNS=NRUNS+1
            WTRL=BWP
            AVE=AVE+BWP
          ENDIF
  326     CONTINUE
  327   CONTINUE
        AVET=AVET+AVE
        NPOST=NPOST+NPOS
        NNEGT=NNEGT+NNEG
        AVE=AVE/REAL(NPR)
        IF(IO.EQ.1) WRITE(IOUT,695) VMAX,VMIN,AVE,NPOS,NNEG,NRUNS,N
      ENDIF
C-----PRINT WEIGHTED RESIDUALS FOR PRIOR INFORMATION ON PARAMETER SUMS
      IF(IPAR.GT.0.AND.MPR.GT.0) THEN
        IF(IO.EQ.1) WRITE(IOUT,650)
        NNEG=0
        NPOS=0
        VMAX=-1.E20
        VMIN=1.E20
        AVE=0.
        DO 350 IMP=1,MPR
          TEMP=0.
          LFLAG=0
          DO 340 IIP=1,NPO
            IF(PRM(IIP,IMP).NE.0..AND.LN(IIP).GT.0) LFLAG=1
            TEMP=TEMP+PRM(IIP,IMP)*B(IIP)
  340     CONTINUE
          WPSR=WP(NPO+IMP)**.5
          TEMP1=PRM(NPO+1,IMP)
          TEMP1E=TEMP1
          TEMPL=TEMP
          IF(LFLAG.NE.0) THEN
            TEMPL=ALOG(TEMP)
            TEMP1E=EXP(TEMP1)
          ENDIF
          BDIF=TEMP1-TEMPL
          BWP=BDIF*WPSR
          IF(IO.EQ.1) WRITE(IOUT,660) IMP,TEMP1E,TEMP,BDIF,WPSR,BWP
          RSQP=RSQP+(BWP**2)
          IF(BWP.GT.VMAX) VMAX=BWP
          IF(BWP.LT.VMIN) VMIN=BWP
          IF(BWP.GE.0.) NPOS=NPOS+1
          IF(BWP.LT.0.) NNEG=NNEG+1
          N=N+1
          IF(N.GT.1.AND.(WTRL*BWP).LT.0.) NRUNS=NRUNS+1
          WTRL=BWP
          AVE=AVE+BWP
  350   CONTINUE
        AVET=AVET+AVE
        NPOST=NPOST+NPOS
        NNEGT=NNEGT+NNEG
        AVE=AVE/REAL(MPR)
        IF(IO.EQ.1) WRITE(IOUT,695) VMAX,VMIN,AVE,NPOS,NNEG,NRUNS,N
      ENDIF
C-------FINAL PRINTOUT
      WRITE(IOUT,670) RSQ
      IF(IPAR.GT.0.AND.NPR+MPR.GT.0) WRITE(IOUT,675) RSQP
      IF(IO.EQ.1) THEN
        AVET=AVET/REAL(NH+NQT+NPR+MPR-JDRY-IDIS)
        RP=REAL(NPOST)
        RN=REAL(NNEGT)
        RNP=2.*RP*RN
        RNS=RP+RN
        RNR=REAL(NRUNS)
        IF(RNP.GT.0.) THEN
          ERUNS=(RNP/RNS)+1.0
          SDRUNS=((RNP*(RNP-RNS))/((RNS**2.)*(RNS-1.)))**.5
```

```
          STRUNS=(RNR-ERUNS+.5)/SDRUNS                                  SSJ0627
          ST2RUNS=(RNR-ERUNS-.5)/SDRUNS                                 SSJ0628
          WRITE(IOUT,685) AVET,NPOST,NNEGT,NRUNS,N                      SSJ0629
          WRITE(IOUT,690) STRUNS,ST2RUNS                                SSJ0630
        ELSE                                                            SSJ0631
          WRITE(IOUT,685) AVET,NPOST,NNEGT,NRUNS,N                      SSJ0632
        ENDIF                                                           SSJ0633
      ENDIF                                                             SSJ0634
      RETURN                                                            SSJ0635
      END                                                               SSJ0636


C=======================================================================SSJ0637
      SUBROUTINE SSEN1P(                                                SSJ0638
     1       SMAT,NLL,NSM,NM,H,HOLD,A,NC,NR,NL,KPER,NPER,PID,ST,DELR,   SSJ0639
     2       DELC,IBOUND,TRPY,DELT,XX,ISN,RHS,SHOLD,BOT,DD,IHESS,B,LN,  SSJ0640
     3       SCL,NMM,NZM,LZI1,SFAC,LZ,LM,MATZ,NLLI1,CV)                 SSJ0641
C-----VERSION 1000 01FEB1992                                            SSJ0642
C     ******************************************************************SSJ0643
C      CALCULATE MATRIX DERIVATIVES AND MULTIPLY BY HEADS AND ADJOINTS  SSJ0644
C      STATES, AS NEEDED.  ADD RESULTING CONTRIBUTION TO THE ADJOINT-   SSJ0645
C      STATE SENSITIVITIES OR, FOR THE SENSITIVITY-EQUATION METHOD, TO  SSJ0646
C      RHS.                                                             SSJ0647
C     ******************************************************************SSJ0648
C        SPECIFICATIONS:                                                SSJ0649
C     ------------------------------------------------------------------SSJ0650
       CHARACTER*4 PID,PID1                                             SSJ0651
       DOUBLE PRECISION A(NC*NR*NL),DD,XX                               SSJ0652
       DIMENSION SMAT(NC,NR,NMM),NLL(NLLI1),H(NC*NR*NL),ST(NC,NR,NL),   SSJ0653
     1       DELR(NC),DELC(NR),RHS(NC,NR,NL),IBOUND(NC,NR,NL),TRPY(NL), SSJ0654
     2       HOLD(NC*NR*NL),SHOLD(NC*NR*NL),BOT(NC,NR,NL),CV(NC,NR,NL), SSJ0655
     3       MATZ(NC,NR,NZM),LZ(LZI1,NSM),SFAC(NSM),LM(NSM)             SSJ0656
       COMMON /FLWCOM/LAYCON(80)                                        SSJ0657
C     ------------------------------------------------------------------SSJ0658
C                                                                       SSJ0659
       NRC=NR*NC                                                        SSJ0660
       NR1=NR-1                                                         SSJ0661
       NC1=NC-1                                                         SSJ0662
C                                                                       SSJ0663
       DO 390 LL=1,NLLI1                                                SSJ0664
         IL=NLL(LL)                                                     SSJ0665
         IF(IL.EQ.0) THEN                                               SSJ0666
           IF(PID.EQ.'TKV ') GO TO 390                                  SSJ0667
           GO TO 400                                                    SSJ0668
         ENDIF                                                          SSJ0669
         IF(IL.GT.0.AND.PID.NE.'ANI '.AND.PID.NE.'ANIV') THEN           SSJ0670
           NM=NM+1                                                      SSJ0671
           SF=SFAC(NM)                                                  SSJ0672
           LZ1=LZ(1,NM)                                                 SSJ0673
           M=LM(NM)                                                     SSJ0674
         ENDIF                                                          SSJ0675
         K=IABS(IL)                                                     SSJ0676
         IF(PID.EQ.'TKV '.AND.MOD(LL+2,3).EQ.0) K=K-1                   SSJ0677
         LT=LAYCON(K)                                                   SSJ0678
         IF(LT.EQ.1) THEN                                               SSJ0679
           KB=0                                                         SSJ0680
           DO 210 KK=1,K                                                SSJ0681
             IF(LAYCON(KK).EQ.1) KB=KB+1                                SSJ0682
  210      CONTINUE                                                     SSJ0683
         ENDIF                                                          SSJ0684
         PID1=PID                                                       SSJ0685
         IF(PID.EQ.'TKV ') THEN                                         SSJ0686
           IF(MOD(LL+1,3).EQ.0) THEN                                    SSJ0687
             PID='T   '                                                 SSJ0688
           ELSE                                                         SSJ0689
             PID='KV                                                    SSJ0690
           ENDIF                                                        SSJ0691
         ENDIF                                                          SSJ0692
C-----HORIZONTAL CONDUCTANCES                                           SSJ0693
         IF(PID.EQ.'T   '.OR.PID.EQ.'ANI ') THEN                        SSJ0694
           YX=TRPY(K)*2.                                                SSJ0695
           DO 330 I=1,NR                                                SSJ0696
           DO 330 J=1,NC                                                SSJ0697
             IF(IBOUND(J,I,K).EQ.0) GO TO 330                           SSJ0698
             IND=J+NC*(I-1)+NRC*(K-1)                                   SSJ0699
             HO=H(IND)                                                  SSJ0700
             DR=DELR(J)                                                 SSJ0701
             DC=DELC(I)                                                 SSJ0702
             R0=0.0                                                     SSJ0703
             SMAT0=1.                                                   SSJ0704
             IF(IL.GT.0.AND.PID.NE.'ANI ') THEN                         SSJ0705
               IF(M.EQ.0) SMAT0=SF                                      SSJ0706
               IF(M.GT.0) SMAT0=SF*SMAT(J,I,M)                          SSJ0707
```

```
            IF(SMAT0.NE.0..AND.LZ1.GT.0) THEN
              IFLAG0=0
              DO 250 IZ=2,LZI1
                NZ=LZ(IZ,NM)
                IF(NZ.EQ.0.OR.IFLAG0.EQ.1) GO TO 255
                IF(MATZ(J,I,LZ1).EQ.NZ) IFLAG0=1
  250         CONTINUE
  255         IF(IFLAG0.EQ.0) SMAT0=0.
            ENDIF
          ENDIF
          IF(LT.EQ.1) TH1=HO-BOT(J,I,KB)
C-------CR
          IF(IBOUND(J+1,I,K).EQ.0.OR.PID.EQ.'ANI '.OR.J.EQ.NC)
     1       GO TO 300
          SMAT1=1.
          IF(IL.GT.0) THEN
            IF(M.EQ.0) SMAT1=SF
            IF(M.GT.0) SMAT1=SF*SMAT(J+1,I,M)
            IF(SMAT1.NE.0..AND.LZ1.GT.0) THEN
              IFLAG1=0
              DO 270 IZ=2,LZI1
                NZ=LZ(IZ,NM)
                IF(NZ.EQ.0.OR.IFLAG1.EQ.1) GO TO 275
                IF(MATZ(J+1,I,LZ1).EQ.NZ) IFLAG1=1
  270         CONTINUE
  275         IF(IFLAG1.EQ.0) SMAT1=0.
            ENDIF
          ENDIF
          IF((IL.GT.0).AND.(SMAT0.EQ.0..AND.SMAT1.EQ.0.))
     1       GO TO 300
          HP=H(IND+1)
          DR1=DELR(J+1)
          IF(LT.EQ.1) TH2=HP-BOT(J+1,I,KB)
          IF(IL.LT.0.OR.(SMAT0.GT.0..AND.SMAT1.GT.0.))
     1     THEN
            IF(LT.EQ.0) THEN
              IF(IL.GT.0) THEN
                T1=SMAT0
                T2=SMAT1
              ELSE
                T1=1.
                T2=1.
              ENDIF
            ELSE
              T1=TH1
              T2=TH2
            ENDIF
            CO=2.*T2*T1*DC/(T1*DR1+T2*DR)
          ELSE
            T1=ST(J,I,K)
            T2=ST(J+1,I,K)
            SM1=SMAT0
            SM2=SMAT1
            IF(LT.EQ.1) THEN
              T1=T1*TH1
              T2=T2*TH2
              IF(SM1.NE.0.) SM1=TH1*SM1
              IF(SM2.NE.0.) SM2=TH2*SM2
            ENDIF
            IF(SM1.EQ.0.)
     1         CO=2.*DC*(T1**2)*DR1*SM2/((T1*DR1+T2*DR)**2)
            IF(SM2.EQ.0.)
     1         CO=2.*DC*(T2**2)*DR*SM1/((T1*DR1+T2*DR)**2)
          ENDIF
          HH=HO-HP
          IF(ISN.GT.0) THEN
            IF(IHESS.EQ.0) XX=XX - CO*(HO-HP)*A(IND)
     1                           + CO*(HO-HP)*A(IND+1)
            IF(IHESS.GT.0) THEN
              BB=1.
              IF(LN.GT.0) BB=B/SCL
              RHS(J,I,K)=RHS(J,I,K)+CO*(HO-HP)*DD*BB*SCL
              RHS(J+1,I,K)=RHS(J+1,I,K)-CO*(HO-HP)*DD*BB*SCL
            ENDIF
          ELSE
            R0=R0+CO*HH
            RHS(J+1,I,K)=RHS(J+1,I,K)-CO*HH
          ENDIF
300     CONTINUE
C-------CC
        IF(IBOUND(J,I+1,K).EQ.0.OR.I.EQ.NR) GO TO 325
        SMAT1=1.
        IF(IL.GT.0.AND.PID.NE.'ANI ') THEN
```

```
            IF(M.EQ.0) SMAT1=SF
            IF(M.GT.0) SMAT1=SF*SMAT(J,I+1,M)
            IF(SMAT1.NE.0..AND.LZ1.GT.0) THEN
              IFLAG1=0
              DO 310 IZ=2,LZI1
                NZ=LZ(IZ,NM)
                IF(NZ.EQ.0.OR.IFLAG1.EQ.1) GO TO 315
                IF(MATZ(J,I+1,LZ1).EQ.NZ) IFLAG1=1
310           CONTINUE
315           IF(IFLAG1.EQ.0) SMAT1=0.
            ENDIF
          ENDIF
          IF(PID.EQ.'T   '.AND.IL.GT.0.AND.
     1          (SMAT0.EQ.0..AND.SMAT1.EQ.0.)) GO TO 325
          HP=H(IND+NC)
          DC1=DELC(I+1)
          IF(LT.EQ.1) TH2=HP-BOT(J,I+1,KB)
          IF(PID.EQ.'T   ') THEN
            IF(IL.LT.0.OR.
     1          (SMAT0.GT.0..AND.SMAT1.GT.0.))THEN
              IF(LT.EQ.0) THEN
                IF(IL.GT.0) THEN
                  T1=SMAT0
                  T2=SMAT1
                ELSE
                  T1=1.
                  T2=1.
                ENDIF
              ELSE
                T1=TH1
                T2=TH2
              ENDIF
              CO=YX*T2*T1*DR/(T1*DC1+T2*DC)
            ELSE
              T1=ST(J,I,K)
              T2=ST(J,I+1,K)
              SM1=SMAT0
              SM2=SMAT1
              IF(LT.EQ.1) THEN
                T1=T1*TH1
                T2=T2*TH2
                IF(SM1.NE.0.) SM1=TH1*SM1
                IF(SM2.NE.0.) SM2=TH2*SM2
              ENDIF
              IF(SM1.EQ.0.)
     1              CO=YX*DR*(T1**2)*DC1*SM2/((T1*DC1+T2*DC)**2)
              IF(SM2.EQ.0.)
     1              CO=YX*DR*(T2**2)*DC*SM1/((T1*DC1+T2*DC)**2)
            ENDIF
          ENDIF
          IF(PID.EQ.'ANI ') THEN
            T1=ST(J,I,K)
            T2=ST(J,I+1,K)
            IF(LT.EQ.1) THEN
              T1=T1*TH1
              T2=T2*TH2
            ENDIF
            IF(T2.EQ.0.0.OR.T1.EQ.0.0) GO TO 330
            CO=2.*T2*T1*DR/(T1*DC1+T2*DC)
          ENDIF
          HH=HO-HP
          IF(ISN.GT.0) THEN
            IF(IHESS.EQ.0) XX=XX - CO*(HO-HP)*A(IND)
     1                           + CO*(HO-HP)*A(IND+NC)
            IF(IHESS.GT.0) THEN
              BB=1.
              IF(LN.GT.0) BB=B/SCL
              RHS(J,I,K)=RHS(J,I,K)+CO*(HO-HP)*DD*BB*SCL
              RHS(J,I+1,K)=RHS(J,I+1,K)-CO*(HO-HP)*DD*BB*SCL
            ENDIF
          ELSE
            R0=R0+CO*HH
            RHS(J,I+1,K)=RHS(J,I+1,K)-CO*HH
          ENDIF
325       RHS(J,I,K)=RHS(J,I,K)+R0
330     CONTINUE
      ENDIF
C-------CV
      IF(PID.EQ.'KV  '.OR.PID.EQ.'ANIV') THEN
        DO 360 I=1,NR
        DO 360 J=1,NC
          IF(IBOUND(J,I,K).EQ.0.OR.IBOUND(J,I,K+1).EQ.0) GO TO 360
          IF(IL.LT.0) CO=DELR(J)*DELC(I)
```

```
          IF(IL.GT.0.AND.PID.EQ.'KV  ') THEN
            SM=SF
            IF(M.NE.0) THEN
              SM=0.
              IF(SMAT(J,I,M).NE.0.) SM=SF*SMAT(J,I,M)
            ENDIF
            IF(SM.NE.0..AND.LZ1.GT.0) THEN
              IFLAG1=0
              DO 340 IZ=2,LZI1
                NZ=LZ(IZ,NM)
                IF(NZ.EQ.0.OR.IFLAG1.EQ.1) GO TO 345
                IF(MATZ(J,I,LZ1).EQ.NZ) IFLAG1=1
  340         CONTINUE
  345         IF(IFLAG1.EQ.0) SM=0.
            ENDIF
            IF(SM.EQ.0.) GO TO 360
            CO=(CV(J,I,K)**2)/(SM*B*B*DELR(J)*DELC(I))
          ENDIF
          IF(PID.EQ.'ANIV') CO=CV(J,I,K)/B
          IND=J+NC*(I-1)+NRC*(K-1)
          HH=H(IND)-H(IND+NRC)
          IF(ISN.GT.0) THEN
            IF(IHESS.EQ.0) XX=XX - CO*HH*A(IND) + CO*HH*A(IND+NRC)
            IF(IHESS.GT.0) THEN
              BB=1.
              IF(LN.GT.0) BB=B/SCL
              RHS(J,I,K)=RHS(J,I,K)+CO*HH*DD*BB*SCL
              RHS(J,I,K+1)=RHS(J,I,K+1)-CO*HH*DD*BB*SCL
            ENDIF
          ELSE
            RHS(J,I,K)=RHS(J,I,K)+CO*HH
            RHS(J,I,K+1)=RHS(J,I,K+1)-CO*HH
          ENDIF
 360    CONTINUE
      ENDIF
C-----S
      IF(PID.EQ.'S1  ') THEN
        IF((ISN.GT.0.AND.IHESS.EQ.0.AND.KPER.EQ.NPER).OR.
     1      ((ISN.LT.0.OR.IHESS.GT.0).AND.KPER.EQ.1)) GO TO 390
        DO 370 I=1,NR
        DO 370 J=1,NC
          IF(IBOUND(J,I,K).LE.0) GO TO 370
          IND=J+NC*(I-1)+NRC*(K-1)
          IF(IL.LT.0) CO=DELR(J)*DELC(I)/DELT
          IF(IL.GT.0) THEN
            SM=SF
            IF(M.NE.0) SM=SF*SMAT(J,I,NM)
            IF(SM.NE.0..AND.LZ1.GT.0) THEN
              IFLAG1=0
              DO 365 IZ=2,LZI1
                NZ=LZ(IZ,NM)
                IF(NZ.EQ.0.OR.IFLAG1.EQ.1) GO TO 367
                IF(MATZ(J,I,LZ1).EQ.NZ) IFLAG1=1
  365         CONTINUE
  367         IF(IFLAG1.EQ.0) SM=0.
            ENDIF
            IF(SM.EQ.0.) GO TO 370
            IF(LT.EQ.1) SM=1.
            CO=SM*DELR(J)*DELC(I)/DELT
          ENDIF
C       DAH AND DBH
          AO=A(IND)
          HO=H(IND)
          IF(ISN.GT.0) THEN
            IF(IHESS.EQ.0) XX = XX - CO*(HO-HOLD(IND))*AO
            IF(IHESS.GT.0) THEN
              BB=1.
              IF(LN.GT.0) BB=B/SCL
              RHS(J,I,K)=RHS(J,I,K)-CO*(SHOLD(IND)-HO)*DD*BB*SCL
            ENDIF
          ELSE
            RHS(J,I,K)=RHS(J,I,K)-CO*(SHOLD(IND)-HO)
          ENDIF
370     CONTINUE
      ENDIF
      PID=PID1
390   CONTINUE
400   RETURN
      END
```

```
C=======================================================================
      SUBROUTINE SSEN1R(
     1       NLL,KPER,NPER,NCOL,NROW,NLAY,DELR,DELC,SMAT,NSM,NM,IOUT,XX,
     2       IFLAG,ILAY,A,IBOUND,ISN,RHS,SURF,EXDP,PID,SHNW,DD,IHESS,
     3       SCL,NMM,NZM,LZI1,SFAC,LZ,LM,MATZ,NLLI1,NLLIT)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     CALCULATE FORCING FUNCTION DERIVATIVE FOR RCH OR ETM. MULTIPLY BY
C     THE ADJOINT STATE AND ADD TO THE DERIVATIVE, OR ADD TO RHS, AS
C     NEEDED.
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 PID
      DOUBLE PRECISION A(NCOL,NROW,NLAY),RO,DD,XX
      DIMENSION DELR(NCOL),DELC(NROW),NLL(NLLI1),ILAY(NCOL,NROW),
     1      SMAT(NCOL,NROW,NMM),IBOUND(NCOL,NROW,NLAY),SFAC(NSM),
     I      RHS(NCOL,NROW,NLAY),SURF(NCOL,NROW),EXDP(NCOL,NROW),LM(NSM),
     3      SHNW(NCOL,NROW,NLAY),LZ(LZI1,NSM),MATZ(NCOL,NROW,NZM)
C     ------------------------------------------------------------------
      NL1=NLL(1)
      ITS=NPER-KPER
      IF(ISN.LT.0.OR.IHESS.GT.0) ITS=KPER-1
C-------LOOP THROUGH TIMES
      DO 450 K=1,NLLIT
        IF(NLL(2*K).GE.0.AND.(ITS.GE.NLL(2*K).AND.ITS.LE.NLL(2*K+1)))
     1       THEN
          SM=1.
          IF(NL1.GT.0) THEN
            NM=NM+1
            M=LM(NM)
            SF=SFAC(NM)
            LZ1=LZ(1,NM)
          ENDIF
C-------LOOP THROUGH CELLS
          DO 400 IR=1,NROW
          DO 300 IC=1,NCOL
            SM=1.
            IF(NL1.GT.0) THEN
              SM=SF
              IF(M.GT.0) SM=SF*SMAT(IC,IR,M)
              IF(LZ1.GT.0) THEN
                IFL=0
                DO 230 IZ=2,LZI1
                  NZ=LZ(IZ,NM)
                  IF(NZ.EQ.0.OR.IFL.EQ.1) GO TO 235
  230           IF(NZ.EQ.MATZ(IC,IR,LZ1)) IFL=1
  235           IF(IFL.EQ.0) SM=0.
              ENDIF
            ENDIF
            IF(SM.EQ.0.) GO TO 300
C----------IFLAG=1
            IF(IFLAG.EQ.1.AND.IBOUND(IC,IR,1).GT.0) THEN
              RO=SM*DELR(IC)*DELC(IR)
              KK=1
            ENDIF
C----------IFLAG=2
            IF(IFLAG.EQ.2.AND.IBOUND(IC,IR,(ILAY(IC,IR))).GT.0)THEN
              RO=SM*DELR(IC)*DELC(IR)
              KK=ILAY(IC,IR)
            ENDIF
C----------IFLAG=3
            IF(IFLAG.EQ.3) THEN
              DO 250 KK=1,NLAY
                IF(IBOUND(IC,IR,KK).GT.0) GO TO 255
  250         CONTINUE
              GO TO 300
  255         RO=SM*DELR(IC)*DELC(IR)
            ENDIF
C----------ADJUST FOR ETM
            IF(PID.EQ.'ETM ') THEN
              RO=-RO
              S=SURF(IC,IR)
              H=SHNW(IC,IR,KK)
              IF(H.GE.S) GO TO 270
              DDD=S-H
              XXX=EXDP(IC,IR)
              IF(DDD.GE.XXX) GO TO 300
              RO=RO*(1.-(DDD/XXX))
  270         CONTINUE
            ENDIF
C----------CONTRIBUTIONS TO SENSITIVITIES OR RHS.
            IF(ISN.GT.0.AND.IHESS.EQ.0) XX=XX+A(IC,IR,KK)*RO
```

```
            IF(ISN.GT.0.AND.IHESS.GT.0)
     1          RHS(IC,IR,KK)=RHS(IC,IR,KK)-RO*DD*SCL
            IF(ISN.LT.0) RHS(IC,IR,KK)=RHS(IC,IR,KK)-RO
  300     CONTINUE
  400     CONTINUE
        ENDIF
  450 CONTINUE
      IF(NLL(1).GT.0.AND.IR.EQ.0) NM=NM+1
C
      RETURN
      END


C=======================================================================
      SUBROUTINE SSEN1U(
     1        NH,NDER,IOFF,JOFF,KPER,MLAY,IBOUND,RINT,DID,COFF,ROFF,DELR,
     2     DELC,NCOL,NROW,NLAY,PR,H,HOBS,WT,HNEW,X,IP,IDRY,NP,MOBS,B,
     3        LN,ISN,NQT,TOFF,NPER,RHS,PV,MAXM,ITERP,JDRY,IPAR,IOUT)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     INTERPOLATE HEADS; ACCOUNT FOR DRY CELLS, IF NEEDED
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 DID(NH+NQT)
      DOUBLE PRECISION HNEW(NCOL,NROW,NLAY),V,PV
      DIMENSION IBOUND(NCOL,NROW,NLAY),X(NP,NH+NQT),DELR(NCOL),
     1          DELC(NROW),COFF(NH),ROFF(NH),H(NH+NQT),WT(NH+NQT),
     1          HOBS(NH+NQT),RINT(4,NH),JOFF(NH),IOFF(NH),
     1          MLAY(MAXM,MOBS),PR(MAXM,MOBS),NDER(5,NH),B(NP),LN(NP),
     1          TOFF(NH),RHS(NCOL,NROW,NLAY),PV(NH)
      COMMON /FLWCOM/LAYCON(80)
C     ------------------------------------------------------------------
555    FORMAT(/,' HEAD OBS#',I5,', ID ',A4,' IS DRY -- OMIT (SSEN1U)')
560    FORMAT(/,' INTERPOLATION FOR HEAD OBS#',I5,', ID ',A4,' CHANGED',
     1 ' BECAUSE AT LEAST ONE NEIGHBORING CELL IS DRY (SSEN1U)')
565    FORMAT(/,' ', I5,' OF',I5,' OBSERVATIONS OMITTED.  EXECUTION ',
     1  'STOPS IF # OF REMAINING OBS <= # OF PARAMETERS')
C
C-------CALCULATE TIME STEP
      ITS=KPER-1
      IF(ISN.GT.0.AND.IP.EQ.1) ITS=NPER-KPER
C-------CHECK FOR NODES USED TO INTERPOLATE HEADS THAT HAVE GONE DRY.
C-------ELIMINATE OBSERVATIONS OR RECALC. INTERPOLATION COEFFICIENTS.
      IF(IP.EQ.0.AND.ISN.EQ.-1) THEN
C---------SKIP IF ALL LAYERS ARE CONFINED
        KK=0
        DO 20 K=1,NLAY
   20   KK=KK+LAYCON(K)
        IF(KK.EQ.0) GO TO 75
C--------CHECK FOR OBSERVATIONS THAT NEED TO BE OMITTED OR NEED TO HAVE
C--------THE INTERPOLATION RECALCULATED
        IF(KPER.EQ.1) THEN
          IDRY=0
          JDRY=0
        ENDIF
        ML=0
        DO 70 N=1,NH
          K=NDER(1,N)
          II=NDER(2,N)
          JJ=NDER(3,N)
          IO=IOFF(N)
          JO=JOFF(N)
          MM=1
          IF(K.LT.0) THEN
            ML=ML+1
            MM=MAXM
          ENDIF
          IF(NDER(4,N).EQ.ITS.OR.(TOFF(N).GT.0..AND.NDER(4,N).EQ.ITS-1))
     1        THEN
            DO 65 M=1,MM
              KK=K
              IF(K.LT.0) KK=MLAY(M,ML)
              IF(KK.EQ.0) GO TO 70
              IF(LAYCON(KK).EQ.1) THEN
                IF(IBOUND(JJ,II,KK).EQ.0) THEN
                  IDRY=IDRY+1
                  IF(IPAR.GE.1) WT(N)=-ABS(WT(N))
                  WRITE(IOUT,555) N,DID(N)
                  GO TO 70
                ELSE
                  IF((RINT(2,N).NE.0..AND.IBOUND(JJ+JO,II,KK).EQ.0).OR.
     1              (RINT(3,N).NE.0..AND.IBOUND(JJ,II+IO,KK).EQ.0).OR.
```

```
     1            (RINT(4,N).NE.0..AND.IBOUND(JJ+JO,II+IO,KK).EQ.0))THEN
                    IF(MM.GT.1) THEN
                      IDRY=IDRY+1
                      IF(IPAR.GE.1) WT(N)=-ABS(WT(N))
                      WRITE(IOUT,555) N,DID(N)
                      GO TO 70
                    ENDIF
                    WRITE(IOUT,560) N,DID(N)
                    CALL SSEN1I (NDER(1,N),COFF(N),ROFF(N),DELR,DELC,
     1                IBOUND,NCOL,NROW,NLAY,RINT(1,N),JOFF(N),IOFF(N),
     2                MLAY(1,ML))
                    COFF(N)=COFF(N)+5.
                    JDRY=JDRY+1
                  ENDIF
                ENDIF
              ENDIF
   65       CONTINUE
          ENDIF
   70   CONTINUE
        IF(IDRY.GT.0) WRITE(IOUT,565) IDRY,NH
        IF((NH-IDRY).LE.NP) STOP
      ENDIF
C-----INTERPOLATION
   75 ML=0
      DO 80 N=1,NH
        IF(IPAR.GE.1.AND.WT(N).LT.0.) GO TO 80
        IF(NDER(4,N).EQ.ITS-1.AND.TOFF(N).GT.0.) GO TO 77
        IF(NDER(4,N).NE.ITS) GO TO 80
   77   CONTINUE
        K=NDER(1,N)
        II=NDER(2,N)
        JJ=NDER(3,N)
        IO=IOFF(N)
        JO=JOFF(N)
        V=0.
        MM=1
        IF(K.LT.0) THEN
          ML=ML+1
          MM=-K
        ENDIF
        DO 78 M=1,MM
          KK=K
          PROP=1.
          IF(K.LT.0) THEN
            KK=MLAY(M,ML)
            PROP=PR(M,ML)
          ENDIF
          IF(KK.EQ.0) GO TO 79
C--------ADD CONTRIBUTION FROM THIS LAYER TO ....
C-----------SENSITIVITY-EQUATION SENSITIVITIES
          IF(ISN.LT.0.OR.(ISN.GT.0.AND.IP.NE.1))
     1      V = V + PROP * ( RINT(1,N)*HNEW(JJ,II,KK) +
     1      RINT(2,N)*HNEW(JJ+JO,II,KK) +  RINT(3,N)*HNEW(JJ,II+IO,KK)
     1      + RINT(4,N)*HNEW(JJ+JO,II+IO,KK) )
C-----------RHS FOR ADJOINT STATES
          IF(IP.EQ.1.AND.ISN.GT.0) THEN
            FACT=1.0
            IF(TOFF(N).GT.0.) THEN
              FACT=TOFF(N)
              IF(NDER(4,N).EQ.ITS-1) FACT=1.-FACT
            ENDIF
            HD=HOBS(N)-H(N)
            W=WT(N)
            RHS(JJ,II,KK)=RHS(JJ,II,KK)+2.*PROP*FACT*RINT(1,N)*W*HD
            RHS(JJ+JO,II,KK)=RHS(JJ+JO,II,KK)+
     1                     2.*PROP*FACT*RINT(2,N)*W*HD
            RHS(JJ,II+IO,KK)=RHS(JJ,II+IO,KK)+
     1                     2.*PROP*FACT*RINT(3,N)*W*HD
            RHS(JJ+JO,II+IO,KK)=RHS(JJ+JO,II+IO,KK)+
     1                     2.*PROP*FACT*RINT(4,N)*W*HD
            GO TO 80
          ENDIF
   78   CONTINUE
C-------INDICE WHICH, IF NOT ZERO, IDENTIFIES THE HEAD USED TO
C-------CALCULATED DRAWDOWN
   79   N1=NDER(5,N)
C---------HEADS OR DRAWDOWNS
        IF(IP.EQ.0) THEN
          IF(NDER(4,N).EQ.ITS) H(N)=V
          IF(NDER(4,N).EQ.ITS-1.AND.TOFF(N).GT.0.)
     1            H(N)=H(N)+TOFF(N)*(V-H(N))
          IF(N1.GT.0) H(N)=H(N)-H(N1)
        ENDIF
```

```
C---------ITERATION PARAMETER VECTOR FOR CONJUGATE-DIRECTION REGRESSION
        IF(ISN.GT.0.AND.IP.EQ.2) THEN
          IF(NDER(4,N).EQ.ITS) PV(N)=V
          IF(NDER(4,N).EQ.ITS-1.AND.TOFF(N).GT.0.)
     1                 PV(N)=PV(N)+TOFF(N)*(V-PV(N))
          IF(N1.GT.0) PV(N)=PV(N)-PV(N1)
        ENDIF
C---------SENSITIVITY-EQUATION SENSITIVITIES
        IF(ISN.LT.0.AND.IP.GT.0) THEN
          IF(NDER(4,N).EQ.ITS) THEN
            IF(LN(IP).NE.0) X(IP,N)=B(IP)*V
            IF(LN(IP).EQ.0) X(IP,N)=V
          ENDIF
          IF(NDER(4,N).EQ.ITS-1.AND.TOFF(N).GT.0.) THEN
            IF(LN(IP).NE.0) X(IP,N)=X(IP,N)+TOFF(N)*((B(IP)*V)-X(IP,N))
            IF(LN(IP).EQ.0) X(IP,N)=X(IP,N)+TOFF(N)*(V-X(IP,N))
          ENDIF
          IF(N1.GT.0) X(IP,N)=X(IP,N)-X(IP,N1)
        ENDIF
   80 CONTINUE
      RETURN
      END


C=======================================================================
      SUBROUTINE SSEN1V(
     1       NQ,NQC,NQT,NQOB,NQCL,IQOB,QCLS,IBT,MXBND,NBOUND,BNDS,
     2       MXRIVR,NRIVER,RIVR,SHNW,IP,HNEW,NCOL,NROW,NLAY,IOUT,
     3       IBOUND,NPER,KPER,NH,X,DID,NP,H,B,LN,TOFF,MXDRN,NDRAIN,DRAI,
     4       MXSTRM,NSTREM,STRM,ISTRM,ISN,RHS,WT,HOBS)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     SAVE SIMULATED FLOWS AND CALCULATE SENSITIVITIES
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 DID(NH+NQT)
      DOUBLE PRECISION HNEW(NCOL,NROW,NLAY)
      DIMENSION SHNW(NCOL,NROW,NLAY),RIVR(6,MXRIVR),BNDS(5,MXBND),
     1   IBOUND(NCOL,NROW,NLAY),X(NP,NH+NQT),IBT(2,NQ),B(NP),LN(NP),
     2   NQOB(NQ),NQCL(NQ),IQOB(NQT),QCLS(5,NQC),H(NH+NQT),TOFF(NH+NQT),
     3   STRM(11,MXSTRM),ISTRM(5,MXSTRM),RHS(NCOL,NROW,NLAY),WT(NH+NQT),
     4   HOBS(NH+NQT),DRAI(5,MXDRN)
C     ------------------------------------------------------------------
  505 FORMAT(/,' HEADS AT STREAM, RIVER, OR DRAIN CELLS ARE BELOW THE ',
     1 'BOTTOM OF THE STREAMBED, RIVER BED, OR DRAIN AT THE REACHES OR',
     2 /,' CELLS LISTED BELOW,',
     3 ' OR, FOR STREAMS, THE LOSS EXCEEDS THE FLOW IN THE STREAM (*).',
     4 /,' THESE CONDITIONS DIMINISH THE IMPACT',
     2 ' OF THE OBSERVATION ON ALL PARAMETERS EXCEPT, IN SOME CASES, ',
     3 /,' THE HYDRAULIC CONDUCTIVITY OF THE STREAMBED, RIVER BED, OR ',
     7 'DRAIN (SEE TEXT FOR MORE INFORMATION).')
  510 FORMAT(/,' OBS#',I5,', ID ',A4,', TIME STEP ',I5)
  515 FORMAT('    LAYER   ROW  COLUMN')
  520 FORMAT('   SEGMENT REACH')
  525 FORMAT(3I7)
  527 FORMAT(' *',I5,I7)
  530 FORMAT(I7,' OF THE',I7,' REACHES OR CELLS USED TO SIMULATE THE',
     1 ' GAIN OR LOSS ARE AFFECTED.')
  540 FORMAT(' THIS OBSERVATION NO LONGER IMPACTS PARAMETER ',
     1 'ESTIMATION AND WILL BE ELIMINATED FROM THIS PARAMETER ',/,
     2 'ESTIMATION ITERATION')
  550 FORMAT(' CELL OR REACH #',I5,' OF HEAD-DEP. BOUNDARY GAIN OR',
     1 ' LOSS OBS#',I5,' ID=',A4,' NOT FOUND IN CELLS LISTED FOR',/,
     2 ' PACKAGE',I5,' (1 RIVER; 2 GHB; 3 STREAM; 4 DRAIN) --',
     3 ' STOP EXECUTION')
C
C-------INITIALIZE VARIABLES
      ITS=KPER-1
      IF(ISN.GT.0.AND.IP.EQ.1) ITS=NPER-KPER
      NC=0
      NT1=1
      JRBOT=0
C-------LOOP THROUGH BOUNDARY FLOWS
      DO 200 IQ=1,NQ
        IBT1=IBT(1,IQ)
        NT2=NT1+NQOB(IQ)-1
C----------WAS THERE A MEASUREMENT AT THIS BOUNDARY THIS TIME STEP?
        DO 100 NT=NT1,NT2
  100   IF(IQOB(NT).EQ.ITS.OR.(IQOB(NT).EQ.ITS-1.AND.
     1     TOFF(NH+NT).GT.0.)) GO TO 110
        GO TO 190
C----------ASSIGN VARIABLES ACCORDING TO BOUNDARY TYPE
```

```
  110   CONTINUE
        IRBOT=0
        KRBOT=0
        IF(IBT1.EQ.1) NBN=NRIVER
        IF(IBT1.EQ.2) NBN=NBOUND
        IF(IBT1.EQ.3) NBN=NSTREM
        IF(IBT1.EQ.4) NBN=NDRAIN
C----------LOOP THROUGH CELLS.
        NC1=NC+1
        NC2=NC+NQCL(IQ)
        DO 130 N=NC1,NC2
          K=QCLS(1,N)
          I=QCLS(2,N)
          J=QCLS(3,N)
          IF(IBT1.NE.3.AND.IBOUND(J,I,K).LE.0) GO TO 130
C----------LOOP THROUGH DATA FILE TO FIND A MATCH.
          IFLAG=0
          DO 120 NB=1,NBN
            IF(IBT1.EQ.1) THEN
              KK=RIVR(1,NB)
              II=RIVR(2,NB)
              JJ=RIVR(3,NB)
            ENDIF
            IF(IBT1.EQ.2) THEN
              KK=BNDS(1,NB)
              II=BNDS(2,NB)
              JJ=BNDS(3,NB)
            ENDIF
            IF(IBT1.EQ.3) THEN
              IS=ISTRM(4,NB)
              IR=ISTRM(5,NB)
            ENDIF
            IF(IBT1.EQ.4) THEN
              KK=DRAI(1,NB)
              II=DRAI(2,NB)
              JJ=DRAI(3,NB)
            ENDIF
C----------DO CALCULATIONS IF THIS IS A MATCH
            IF((IBT1.NE.3.AND.I.EQ.II.AND.J.EQ.JJ.AND.K.EQ.KK).OR.
     1         (IBT1.EQ.3.AND.K.EQ.IS.AND.I.EQ.IR)) THEN
              IFLAG=1
C-------------ASSIGN VARIABLE VALUES
              IF(IBT1.EQ.3) THEN
                K=ISTRM(1,NB)
                I=ISTRM(2,NB)
                J=ISTRM(3,NB)
              ENDIF
              IF(IP.EQ.0) HHNEW=HNEW(J,I,K)
              IF(IP.GT.0) HHNEW=SHNW(J,I,K)
              IF(IBT1.EQ.1) THEN
                HB=RIVR(4,NB)
                C=RIVR(5,NB)
                RBOT=RIVR(6,NB)
              ENDIF
              IF(IBT1.EQ.2) THEN
                HB=BNDS(4,NB)
                C=BNDS(5,NB)
              ENDIF
              IF(IBT1.EQ.3) THEN
                HB=STRM(2,NB)
                IF(STRM(10,NB).LE.0.) HB=STRM(5,NB)
                C=STRM(3,NB)
                RBOT=STRM(4,NB)
              ENDIF
              IF(IBT1.EQ.4) THEN
                HB=DRAI(4,NB)
                C=DRAI(5,NB)
                RBOT=HB
              ENDIF
C-------------CALCULATE FLOWS
              IF(IP.EQ.0) THEN
                HH=C*(HB-HHNEW)
                ISTRF=0
                IF(IBT1.EQ.3.AND.STRM(10,NB)-STRM(11,NB).LE.0.) ISTRF=1
                IF(((IBT1.NE.2).AND.HHNEW.LE.RBOT).OR.ISTRF.EQ.1) THEN
                  HH=C*(HB-RBOT)
                  IF(ISTRF.EQ.1) HH=STRM(10,NB)
                  IF(JRBOT.EQ.0) WRITE(IOUT,505)
                  JRBOT=1
                  IF(IRBOT.EQ.0) THEN
                    WRITE(IOUT,510)NH+NT,DID(NH+NT),ITS
                    IF(IBT1.NE.3) WRITE(IOUT,515)
                    IF(IBT1.EQ.3) WRITE(IOUT,520)
```

```
                        ENDIF
                        IRBOT=IRBOT+1
                        IF(IBT(2,IQ).EQ.0.OR.ISTRF.EQ.1) KRBOT=KRBOT+1
                        IF(IBT1.NE.3) WRITE(IOUT,525) K,I,J
                        IF(IBT1.EQ.3) THEN
                          IF(ISTRF.EQ.1) THEN
                            WRITE(IOUT,527) IS,IR
                          ELSE
                            WRITE(IOUT,525) IS,IR
                          ENDIF
                        ENDIF
                      ENDIF
                    ENDIF
C-------------CALCULATE SENSITIVITIES
                    IF(IP.GT.0) THEN
                      XX=0.
                      IF((IBT1.EQ.3.AND.STRM(10,NB).LE.0..AND.
     1                   STRM(11,NB).GE.0.).OR.
     1                   (IBT1.EQ.4.AND.HHNEW.LE.HB)) GO TO 120
                      IF(IBT1.EQ.2.OR.HHNEW.GT.RBOT) XX = -C*HNEW(J,I,K)
                      IF(IP.EQ.IBT(2,IQ).AND.(IBT1.EQ.2.OR.HHNEW.GT.RBOT))
     1                   XX = XX + QCLS(5,N)*(HB-HHNEW)
                      IF(IP.EQ.IBT(2,IQ).AND.(IBT1.EQ.1.OR.IBT1.EQ.3)
     1                   .AND.HHNEW.LE.RBOT)
     1                   XX = QCLS(5,N)*(HB-RBOT)
                    ENDIF
                    GO TO 125
                  ENDIF
  120         CONTINUE
              IF(IFLAG.EQ.0) THEN
                WRITE(IOUT,550) N,NH+NT,DID(NH+NT),IBT1
                STOP
              ENDIF
C-------------SUM VALUES FROM INDIVIDUAL CELLS.
  125         CONTINUE
C----------------CALCULATE FACTOR FOR TEMPORAL INTERPOLATION
              FACT=1.0
              IF(TOFF(NH+NT).GT.0.) THEN
                IF(IQOB(NT).EQ.ITS) FACT=1.-TOFF(NH+NT)
                IF(IQOB(NT).EQ.ITS-1) FACT=TOFF(NH+NT)
              ENDIF
C---------------FLOWS
              IF(IP.EQ.0) H(NH+NT)=H(NH+NT)+HH*FACT*QCLS(4,N)
C---------------SENSITIVITY-EQUATION SENSITIVITIES
              IF(IP.GT.0.AND.(ISN.LT.0.OR.IP.EQ.2)) THEN
                IF(LN(IP).GT.0) XX=XX*B(IP)
                X(IP,NH+NT)=X(IP,NH+NT)+XX*FACT*QCLS(4,N)
              ENDIF
  130     CONTINUE
C-------PRINT NUMBER OF CELLS AT WHICH HEAD IS BELOW THE BOTTOM OF THE
C-------STREAM OR RIVER BED; CHECK FOR DISCONNECTED OBSERVATIONS.
          IF(IP.EQ.0.AND.IRBOT.GT.0) WRITE(IOUT,530)IRBOT,NQCL(IQ)
          IF(IP.EQ.0.AND.KRBOT.EQ.NQCL(IQ)) THEN
            WT(NH+NT)=-WT(NH+NT)
            WRITE(IOUT,540)
          ENDIF
C-------UPDATE COUNTERS
  190     NC=NC+NQCL(IQ)
          NT1=NT2+1
C
  200 CONTINUE
      RETURN
      END
```

New PAR Primary Modules

Listed in order of first appearance in MAIN

```
C=======================================================================
      SUBROUTINE PAR1AL (
     1                  ISUM,LENX,IOUT,NP,NH,LCC,LCSCLE,LCG,LCDD,LCWP,
     2                  LCB1,NQT,MPR,LCPRM,ISN,LCSCL,LCPV,ND,
     3                  LCR,LCU,LCGD,LCS)
C-----VERSION 1000 01FEB1992 PAR1AL
C     ******************************************************************
C     ALLOCATE ARRAY STORAGE FOR PARAMETER ESTIMATION
C     ******************************************************************
C     SPECIFICATIONS:
C     ------------------------------------------------------------------
C     ------------------------------------------------------------------
C
C1------IDENTIFY PACKAGE
      WRITE(IOUT,1)
    1 FORMAT(1H0,'PAR1 -- PARAMETER ESTIMATION PACKAGE, ',
     1 'VERSION 1, 12/01/91')
C-------STORE, IN ISOLD, LOCATION OF FIRST UNALLOCATED SPACE IN X.
      ISOLD=ISUM
C----------ARRAYS USED ONLY FOR PARAMETER ESTIMATION
      LCSCL=ISUM
      IF(ISN.GT.0) ISUM=ISUM+NP
      LCWP=ISUM
      ISUM=ISUM+NP+MPR
      LCB1=ISUM
      ISUM=ISUM+NP
      LCPRM=ISUM
      ISUM=ISUM+(NP+1)*MPR
      IF(MOD(ISUM,2).EQ.0) ISUM=ISUM+1
      LCG=ISUM
      IF(ISN.LT.0) ISUM=ISUM+NP*2
      IF(ISN.GT.0) THEN
        IF(MOD(ISUM,2).EQ.0) ISUM=ISUM+1
        ISUM=ISUM+NP*2
      ENDIF
      LCC=ISUM
      IF(ISN.LT.0.OR.ISN.GE.4) ISUM=ISUM+2*NP*NP
      IF(ISN.EQ.1) ISUM=ISUM+2*(NP*NP/2+NP/2+1)
      LCSCLE=ISUM
      IF(ISN.LT.0.OR.ISN.GE.4) ISUM=ISUM+2*NP
      LCPV=ISUM
      IF(ISN.GT.0)  ISUM=ISUM+ND*2
      LCDD=ISUM
      ISUM=ISUM+2*NP
C-------FOR QUASI-NEWTON ADDITION TO THE GAUSS-NEWTON MATRIX
      IF(ISN.LT.0) THEN
        LCR=ISUM
        ISUM=ISUM+2*(NP*NP/2+NP)
        LCU=ISUM
        ISUM=ISUM+NP*2
        LCS=ISUM
        LCS=ISUM+NP*2
        LCGD=ISUM
        ISUM=ISUM+NP*2
      ENDIF
C
C8------PRINT AMOUNT OF SPACE USED BY PARAMETER-ESTIMATION PACKAGE.
      ISP=ISUM-ISOLD
      WRITE (IOUT,4)ISP
    4 FORMAT(1X,I6,' ELEMENTS IN X ARRAY ARE USED FOR PARAMETER',
     1      ' ESTIMATION')
      ISUM1=ISUM-1
      WRITE(IOUT,5)ISUM1,LENX
    5 FORMAT(1X,I6,' ELEMENTS OF X ARRAY USED OUT OF',I7)
      IF(ISUM1.GT.LENX) WRITE(IOUT,6)
    6 FORMAT(1X,'   ***X ARRAY MUST BE DIMENSIONED LARGER***')
C
C9------RETURN
      RETURN
      END


C=======================================================================
      SUBROUTINE PAR1RP (
     1              IU,IOUT,NP,PID,WP,B,CONV,LN,LASTX,DMAX,CSA,TOL,ITMXP,
     2              DD,IOUR,IPRC,IPRINT,FCONV,NPR,EV,NH,WT,NQT,KPRINT,
     3              MPR,PRM,IWPG,SCL,ISN,NFIT,LPRINT,NPO,SOSC,NPER,SOSR)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
```

```
C     READ, CHECK AND STORE DATA FOR PARAMETER ESTIMATION, AND
C        INITIALIZE SOME VARIABLES.
C     ***************************************************************
C        SPECIFICATIONS:
C     ---------------------------------------------------------------
      CHARACTER*4  PID(NPO)
      CHARACTER*10 EST,CVAR
      DOUBLE PRECISION DD(NPO)
      DIMENSION WP(NPO+MPR),B(NPO),CONV(2*NPO+2),LN(NPO),WT(NH+NQT),
     1          PRM(NPO+1,MPR),IWPG(NPO),SCL(NPO)
      DATA EST/'ESTIMATE  '/,CVAR/'WEIGHT**.5'/
C     ---------------------------------------------------------------
503   FORMAT(8F10.0)
505   FORMAT(8I10)
507   FORMAT(3F5.0,I5,F5.0,8I5,2F5.0)
550   FORMAT(/,'PRIOR ESTIMATES OF PARAMETER SUMS EQUATION',I3,
     1 ' INCLUDES MULTIPLE PARAMETERS AND PARAMETER(S) WITH LN > 0 --',
     1 /,' STOP EXECUTION (PAR1RP)',/)
560   FORMAT(/,' PRIOR ESTIMATES OF PARAMETER SUMS',/,' EQ.# : ',4X,
     18I10,/,5(12X,8I10,/))
565   FORMAT(' PAR#    PID')
570   FORMAT(I5,4X,A4,4X,8G10.3,/,5(17X,8G10.3,/))
575   FORMAT(2X,A10,5X,8G10.3,/,5(17X,8G10.3,/))
580   FORMAT(/,' PARAMETER INFORMATION:',/,
     1 '  (CONVERGENCE CRITERIA LISTED HERE ARE USED TO SOLVE FOR ',
     1 'SENSITIVITY-EQUATION SENSITIVITIES)',/,'       #  ID   ',
     1 'INITIAL VALUE  LN     CONVERGENCE CRITERIA   GROUP# ',
     1 ' WEIGHT OF PRIOR EST. ')
582   FORMAT(/,' PARAMETER INFORMATION',/,
     1 '  (CONVERGENCE CRITERIA LISTED HERE ARE USED TO SOLVE FOR ',
     1 'SENSITIVITY-EQUATION SENSITIVITIES)',/,'       #  ID   ',
     1 'INITIAL VALUE  LN     CONVERGENCE CRITERIA   GROUP# ',
     1 '     SCALING FACTOR     ',' WT OF PRIOR EST. ',/)
584   FORMAT(/,' PARAMETER INFORMATION',/,'      #  ID   ',
     1 'INITIAL VALUE  LN   GROUP# ',
     1 '     SCALING FACTOR     ',' WT OF PRIOR EST. ',/)
585   FORMAT(' ',I5,2X,A4,G14.4,I3,1X,2G13.3,1X,I4,3X,2G14.4)
590   FORMAT(' ',I5,2X,A4,G14.4,I3,I4,3X,2G14.4)
595   FORMAT(/,' REGRESSION DATA:',
     $/1H ,36HMAX. PARAMETER CORRECTION (DMAX)  = ,G11.5
     $/1H ,36HSEARCH DIR. ADJUSTMENT PAR. (CSA) = ,G11.5
     $/1H ,36HCLOSURE CRITERION (TOL) --------- = ,G11.5
     $/1H ,36HMAXIMUM NO. OF ITERATIONS (ITMXP) = ,I7,
     $/1H ,36HMODIFY CONV. CRITERIA (>0) (FCONV)= ,G11.5,
     $/1H ,36HUSE PRIOR PAR EST (>0) (IPRIOR)   = ,I7,
     $/1H ,36HOUTPUT UNIT FOR RESAN (IOUR) ---- = ,I7,
     $/1H ,36HFORMAT CODE FOR COV AND COR (IPRC)= ,I7,
     $/1H ,36HPRINT SENS., ETC. (>0)    (IPRINT)= ,I7,
     $/1H ,36HPRINT LAST HEADS (1) (KPRINT) --- = ,I7,
     $/1H ,36HPRINT EIGEN V&V OF COV(>0)(LPRINT)= ,I7,
     $/1H ,36HCALC. SEN. W/ LAST PAR (>0)(LASTX)= ,I7,
     $/1H ,36HNO. FLETCHER-REEVES ITERS   (NFIT)= ,I7,
     $/1H ,36HSUM OF SQUARES CLOSURE CRIT.(SOSC)= ,G11.5,
     $/1H ,36HCRITERIA FOR ADDING MATRIX R(SOSR)= ,G11.5)
600   FORMAT(/,'ESTIMATED ERROR VARIANCE <= 0 -- '
     1'STOP EXECUTION (PAR1RP)',/)
610   FORMAT(/,' WARNING -- NP > ND/3 : YOU MAY BE TRYING TO ESTIMATE'
     1     ,' TOO MANY PARAMETERS FOR THE DATA (PAR1RP)',/)
620   FORMAT(/,' ND MUST BE GREATER THAN NP -- STOP EXECUTION (PAR1RP)')
630   FORMAT(/,' ERROR IN SCL FOR PARAMETER SUMS EQ#',I5,
     1 ' : MULTIPLE PARAMETERS REQUIRE SCL=1.0 -- STOP EXECUTION',
     2   ' (PAR1RP)')
635   FORMAT(/,' ISN=2, BUT NFIT<ITMXP -- STOP EXECUTION (PAR1RP)')
640   FORMAT(/,' FOR ISN>0, LASTX MUST EQUAL 0, 1 OR 2, AND IF LASTX>0,'
     1 ' ISN MUST EQUAL 4 -- STOP EXECUTION (PAR1RP)')
C-------SET ERROR FLAG
      IERR=0
C-------STATS FOR PRIOR USING INITIAL PARAMETER VALUES (DATA SET 10)
      READ(IU,503)(WP(IP),IP=1,NPO)
C-------SPECIFY STATISTIC TYPE (DATA SET 11)
      READ(IU,505)IWP
C-------READ PRIOR INFORMATION ON SUMS OF PARAMETERS (DATA SET 12)
      IF(MPR.GT.0) THEN
        DO 60 IPM=1,MPR
          READ(IU,503)(PRM(IP,IPM),IP=1,NPO+1),WP(NPO+IPM)
   60   CONTINUE
      ENDIF
C-------DATA SET 13
      READ(IU,507) DMAX,CSA,TOL,ITMXP,FCONV,IPRIOR,IOUR,IPRC,IPRINT,
     1             KPRINT,LPRINT,LASTX,NFIT,SOSC,SOSR
      WRITE(IOUT,595) DMAX,CSA,TOL,ITMXP,FCONV,IPRIOR,IOUR,IPRC,
     1                IPRINT,KPRINT,LPRINT,LASTX,NFIT,SOSC,SOSR
C-------SCALING FACTORS FOR GRADIENT-SEARCH NONLINEAR REGRESSION
```

```
      IF(ISN.GT.0.AND.ISN.NE.10) READ(IU,503) (SCL(IP),IP=1,NPO)
C-------ERROR CHECKING
      IF(ISN.EQ.2.AND.NFIT.LT.ITMXP) THEN
        WRITE(IOUT,635)
        IERR=1
      ENDIF
      IF(ISN.GT.0.AND.(LASTX.GT.2.OR.(LASTX.NE.0.AND.ISN.NE.4))) THEN
        WRITE(IOUT,640)
        IERR=1
      ENDIF
      IF(EV.LE.0.) THEN
        WRITE(IOUT,600)
        IERR=1
      ENDIF
C-------OPTION TO IGNORE PRIOR INFORMATION
      IF(IPRIOR.EQ.0) THEN
        DO 150 I=1,NPO
  150   WP(I)=0.
        MPR=0
      ENDIF
C-------CONVERT SCALING FACTORS FOR LOG-TRANSFORMED PARAMETERS
      IF(ISN.GT.0) THEN
        DO 155 IP=1,NPO
  155   IF(LN(IP).GT.0) SCL(IP)=ALOG(SCL(IP))
      ENDIF
C-------WRITE HEADER
      IF(ISN.LT.0) WRITE(IOUT,580)
      IF(ISN.EQ.4) WRITE(IOUT,582)
      IF(ISN.GT.0.AND.ISN.LT.4) WRITE(IOUT,584)
C-------LOOP THROUGH PARAMETERS
      NPR=0
      DO 160 IP=1,NPO
C---------CONVERT STATS OF PRIOR SINGLE-PARAMETER ESTIMATES TO WEIGHTS
        IF(WP(IP).NE.0..AND.IP.LE.NP) THEN
          BB=B(IP)
          IF(LN(IP).GT.0) BB=ALOG(BB)
          WPP=WP(IP)
          IF(IWP.EQ.2) WPP=WPP*BB
          IF(IWP.GT.0) WPP=WPP*WPP
          WP(IP)=EV/WPP
          NPR=NPR+1
        ENDIF
C---------IF GROUP# < 0, SET LN, WP, AND SCL TO ZERO
        IF(IP.GT.NP) THEN
          LN(IP)=0
          WP(IP)=0.
          IF(ISN.GT.0) SCL(IP)=0.
        ENDIF
C---------PRINT INITIAL PARAMETER VALUES, CONV. CRIT., PARAMETER,
C---------GROUP #, WEIGHT OF PRIOR PARAMETER EST., AND SCALING FACTORS
        IF(ISN.LT.0.OR.ISN.EQ.10) THEN
          IF(WP(IP).LE.0) WRITE(IOUT,585) IP,PID(IP),B(IP),LN(IP),
     1                CONV(1+IP),CONV(NPO+2+IP),IWPG(IP)
          IF(WP(IP).GT.0) WRITE(IOUT,585) IP,PID(IP),B(IP),LN(IP),
     1                CONV(1+IP),CONV(NPO+2+IP),IWPG(IP),WP(IP)
        ELSEIF (ISN.EQ.4) THEN
          IF(WP(IP).LE.0) WRITE(IOUT,585) IP,PID(IP),B(IP),LN(IP),
     1                CONV(1+IP),CONV(NPO+2+IP),IWPG(IP),SCL(IP)
          IF(WP(IP).GT.0) WRITE(IOUT,585) IP,PID(IP),B(IP),LN(IP),
     1                CONV(1+IP),CONV(NPO+2+IP),IWPG(IP),SCL(IP),WP(IP)
        ELSEIF (ISN.GT.0.AND.ISN.LT.4) THEN
          IF(WP(IP).LE.0) WRITE(IOUT,590) IP,PID(IP),B(IP),LN(IP),
     1                IWPG(IP),SCL(IP)
          IF(WP(IP).GT.0) WRITE(IOUT,590) IP,PID(IP),B(IP),LN(IP),
     1                IWPG(IP),SCL(IP),WP(IP)
        ENDIF
  160 CONTINUE
C-------PRIOR ESTIMATES OF PARAMETER SUMS
      IF(MPR.NE.0) THEN
C----------CONVERT STATISICS TO WEIGHTS; CALCULATE WP**.5 FOR PRINTING
        DO 165 IPM=1,MPR
          WPP=WP(NPO+IPM)
          BB=PRM(NPO+1,IPM)
          DO 163 IP=1,NPO
  163     IF(PRM(IP,IPM).NE.0..AND.LN(IP).NE.0) BB=ALOG(BB)
          PRM(NPO+1,IPM)=BB
          IF(IWP.EQ.2) WPP=WPP*BB
          IF(IWP.GT.0) WPP=WPP**2.
          WP(NPO+IPM)=(EV/WPP)**.5
  165   CONTINUE
C----------PRINT
        WRITE(IOUT,560) (IPM,IPM=1,MPR)
        WRITE(IOUT,565)
```

```
      DO 170 IP=1,NPO
  170   WRITE(IOUT,570) IP,PID(IP),(PRM(IP,IPM),IPM=1,MPR)
      WRITE(IOUT,575) EST,(PRM(NPO+1,IPM),IPM=1,MPR)
      WRITE(IOUT,575) CVAR,(WP(NPO+IPM),IPM=1,MPR)
C----------CONVERT FROM WP**.5 TO WP FOR USE IN REGRESSION
      DO 171 IPM=1,MPR
  171   WP(NPO+IPM)=WP(NPO+IPM)**2.
C----------ERROR CHECKING
      DO 174 IPM=1,MPR
        ICOUNT=0
        IFL=0
        IFL2=0
        DO 173 IP=1,NPO
          IF(PRM(IP,IPM).NE.0) THEN
            ICOUNT=ICOUNT+1
            IF(ISN.GT.0.AND.SCL(IP).NE.1.) IFL=1
            IF(LN(IP).NE.0) IFL2=1
          ENDIF
  173   CONTINUE
        IF(ICOUNT.GT.1.AND.IFL.EQ.1) THEN
          WRITE(IOUT,630) IPM
          IERR=1
        ENDIF
        IF(ICOUNT.GT.1.AND.IFL2.EQ.1) THEN
          WRITE(IOUT,550) IPM
          IERR=1
        ENDIF
  174   CONTINUE
      ENDIF
C-------CHECK NP VERSUS ND
      ND=NH+NQT
      IF(NP.GT.(ND/3).AND.NP.LT.ND) WRITE(IOUT,610)
      IF(NP.GE.ND) THEN
        WRITE(IOUT,620)
        IERR=1
      ENDIF
C-------STOP IF THERE ARE ERRORS IN THE DATA
      IF(IERR.NE.0) STOP
C-------INITIALIZE DD
      DO 190 IP=1,NPO
  190 DD(IP)=0.D00
      RETURN
      END


C======================================================================
      SUBROUTINE PAR1AS(
     1         RSQP,KNT,AP,B,DD,ITERP,G,NP,IP,H,HOBS,SCL,WP,WT,ND,MPR,
     2         IOUT,SSX,NFIT,ITG,PID,BUFF,B1,C,XX,LN,IFO,LASTX,ADMX,PRM,
     3         IPAR,TOL,ITMXP,NPR,RSQO,AGMX,DID,NDER,ROFF,COFF,TOFF,NH,
     4         NQT,IQOB,IDRY,IWPG,IOUB,EV,ISN,ISN1,SCLE,CC,NPO)
C-----VERSION 1000 01FEB1992 PAR1AS
C     ****************************************************************
C     FOR GRADIENT-SEARCH NONLINEAR OPTIMIZATION
C        IP=0: CALCULATE THE SUM OF SQUARED ERRORS
C        IP=1: CALCULATE SEARCH DIRECTION
C     ****************************************************************
C     SPECIFICATIONS:
C     ----------------------------------------------------------------
      CHARACTER*4 PID(NP),DID(ND)
      DOUBLE PRECISION C((NP*NP/2)+NP),DD(NP),SSG,BDP,SCLDP,XX(NP),
     1  G(NP),TEMP,XD,XHX,SUM,GAM,BUFF(NP),CC(NP,NP),SCLE(NP)
      DIMENSION B(NPO),SCL(NP),H(ND),HOBS(ND),WP(NP+MPR),WT(ND),
     1     B1(NPO),LN(NP),PRM(NPO+1,MPR),NDER(4,ND),ROFF(ND),COFF(ND),
     1     TOFF(ND),IQOB(NQT),IWPG(NP)
C     ----------------------------------------------------------------
C
  514 FORMAT (/,' ITERATION NO. = ',I5)
  521 FORMAT (//,'  RSQ(W/PARAMETERS) = ',G11.5,/
     1          ,'  ERROR VARIANCE--- = ',G11.5,/,
     1           '  CORRELATION COEF. = ',G11.5,/
     1          ,'        W/PARAMETERS   = ',G11.5,/
     1          ,'  ITERATIONS------- = ',I5)
  525 FORMAT(/,' CONVERGENCE ACHIEVED',/,'   CONVERGENCE CRITERIA :',/,
     $ 2X,' MAX. FRACTIONAL PAR. CHANGE (ADMX) = ',G11.5,/,
     1 2X,' MAX. NORM OF GRADIENT        (AGMX) = ',G11.5)
  530 FORMAT(/,' RSQ NOT REDUCED, BACK UP 50 PERCENT ALONG DIRECTION',
     1 ' VECTOR')
  540 FORMAT(/,' RSQ NOT REDUCED IN THIS DIRECTION -- STOP EXECUTION',
     1        ' (PAR1AS)')
  545 FORMAT(/,' QUASI-NEWTON METHOD',/)
  550 FORMAT(/,' FLETCHER-REEVES METHOD',/)
  555 FORMAT(/,' FLETCHER-REEVES ITERATION PARAMETER :',/
```

```
     $,2X,' SUM OF SQUARED COMPONENTS OF GRADIENT = ',G11.5,/
     $,2X,' PREVIOUS SUM OF SQUARED COMP. OF GRAD = ',G11.5,/
     $,2X,' ITERATION PARAMETER = RATIO OF SUMS   = ',G11.5,/)
  560 FORMAT(2X,' DIRECTION = -GRADIENT')
  562 FORMAT(13(5X,A4,9X,A4,9X,A4,9X,A4,9X,A4,9X,A4,9X,A4,
     1     9X,A4,9X,A4,9X,A4,9X,A4,/))
  570 FORMAT(/' SCALED SEARCH DIRECTION :',/)
  575 FORMAT (1H0,10(1H ,G11.5,1X))
  585 FORMAT(/,' FINAL APPROXIMATE QUASI-NEWTON INVERSE HESSIAN MATRIX')
  587 FORMAT(12(/,1X,10G13.6))
  590 FORMAT (/,'CHANGE IN GRADIENT VECTOR * DIRECTION VECTOR',
     1' LESS THAN OR EQUAL TO -0.0001 ON ITERATION NO. ',I5,
     1'; VALUE = ',G11.5,' (PAR1AS)')
  595 FORMAT(//,' PARAMETER ESTIMATION DID NOT CONVERGE -- STOP ',
     1  'EXECUTION (PAR1AS)')
  600 FORMAT(F10.0,I5,/,F10.0)
  605 FORMAT(/,' SCALING FROM MODIFIED GAUSS-NEWTON:')
  610 FORMAT(/,' UNSCALED GRADIENT OF THE OBJECTIVE FUNCTION WITHOUT',
     1  ' CONTRIBUTION FROM PRIOR:')
  615 FORMAT(/,' SCALED GRADIENT OF THE OBJECTIVE FUNCTION WITH PRIOR:')
  620 FORMAT(' INITIAL QUASI-NEWTON MATRIX FROM MODIFIED ',
     1    'GAUSS-NEWTON')
C
      IF(ISN1.EQ.10) ITERP=ITERP-1
C-----DO THE FOLLOWING WHEN IP=0
      IF(IP.EQ.0) THEN
C-------INITIALIZE ITG (COUNTER FOR QN)
        IF(ITERP.EQ.1) ITG=1
C-------CHECK FOR REDUCTION IN SUM OF SQUARED ERRORS
        IF(ITERP.GT.1) THEN
          IF(RSQO-RSQP.GT.-0.0001) GO TO 70
          KNT=KNT+1
          IF(ITERP.EQ.ITMXP) THEN
            WRITE(IOUT,595)
            CALL SSEN1O(NP,B,B1,IWPG,PID,WP,IOUT,NH,H,HOBS,WT,DID,NDER,
     2                  ROFF,COFF,TOFF,NQT,IQOB,ND,IPAR,NPR,MPR,PRM,
     3                  RSQ,RSQP,1,LN,RSQH,NPO)
            IF(IOUB.GT.0) CALL SPAR1P(PID,BUFF,NP,IOUB,IOUT,ISN)
            STOP
          ENDIF
          IF(KNT.GT.5) THEN
            WRITE(IOUT,540)
            CALL SSEN1O(NP,B,B1,IWPG,PID,WP,IOUT,NH,H,HOBS,WT,DID,NDER,
     2                  ROFF,COFF,TOFF,NQT,IQOB,ND,IPAR,NPR,MPR,PRM,
     3                  RSQ,RSQP,1,LN,RSQH,NPO)
            IF(IOUB.GT.0) CALL SPAR1P(PID,BUFF,NP,IOUB,IOUT,ISN)
            STOP
          ENDIF
          WRITE(IOUT,530)
        DO 10 IIP=1,NP
   10   IF(LN(IIP).NE.0) B(IIP)=ALOG(B(IIP))
          AP=.5*AP
          DO 60 IIP=1,NP
   60       B(IIP)=B(IIP)-AP*DD(IIP)
        DO 90 IIP=1,NP
   90     IF(LN(IIP).GT.0) B(IIP)=EXP(B(IIP))
          IF(IOUB.GT.0) THEN
            ZERO=0.
            IZERO=0
            WRITE(IOUB,600) ZERO,IZERO,ZERO
          ENDIF
          GO TO 80
   70     KNT=0
        ENDIF
C-------SAVE SUM OF SQUARED ERRORS
        RSQO=RSQP
   80   IF(ISN1.EQ.10) ITERP=ITERP+1
        RETURN
      ENDIF
C
C-----DO THE FOLLOWING WHEN THE GRADIENT IS AVAILABLE (IP=1)
C-------DEFINE FLAG WHEN ISN=10
      IITERP=ITERP
      IF(ISN1.EQ.10.AND.(ITERP.EQ.1.OR.MOD(ITERP-1,NFIT+1).EQ.0))
     1    IITERP=1
C-------ASSIGN SCALE IF ISN=10
      IF(ISN1.EQ.10.AND.IITERP.EQ.1) THEN
        DO 91 IIP=1,NP
   91   SCL(IIP)=1./SCLE(IIP)
        WRITE(IOUT,605)
        WRITE(IOUT,562) (PID(IIP),IIP=1,NP)
        WRITE(IOUT,575) (SCL(IIP),IIP=1,NP)
      ENDIF
```

```
C-------NATURAL LOG CONVERSIONS
      DO 92 IIP=1,NP
        IF(LN(IIP).NE.0) THEN
          XX(IIP)=B(IIP)*XX(IIP)
          B(IIP)=ALOG(B(IIP))
        ENDIF
   92 CONTINUE
C--------PRINT UNSCALED GRADIENT WITHOUT PRIOR INFORMATION
      IF(ITERP.EQ.1) THEN
        WRITE(IOUT,610)
        WRITE(IOUT,562) (PID(IIP),IIP=1,NP)
        WRITE(IOUT,575) (XX(IIP),IIP=1,NP)
      ENDIF
C--------SCALING
      DO 95 IIP=1,NP
        XX(IIP)=XX(IIP)*SCL(IIP)
   95 CONTINUE
C-------ADD PRIOR COMPONENTS TO GRADIENT
C---------PRIOR ESTIMATES OF SINGLE PARAMETERS
      IF(IPAR.GT.0) THEN
        IF(NPR.GT.0) THEN
          DO 100 IIP=1,NP
  100     IF(WP(IIP).GT.0) XX(IIP)=XX(IIP)-2.*(WP(IIP))*
     1                           (B1(IIP)-B(IIP))*SCL(IIP)
        ENDIF
C---------PRIOR ESTIMATES OF PARAMETER SUMS
        IF(MPR.GT.0) THEN
          DO 130 IMP=1,MPR
            TEMP=0.
            DO 110 IIP=1,NPO
  110       TEMP=TEMP+PRM(IIP,IMP)*B(IIP)
            DO 120 IIP=1,NP
  120       XX(IIP)=XX(IIP)-2.*PRM(IIP,IMP)*WP(NPO+IMP)*
     1                      (PRM(NPO+1,IMP)-TEMP)*SCL(IIP)
  130     CONTINUE
        ENDIF
      ENDIF
C--------PRINT GRADIENT
      WRITE(IOUT,615)
      WRITE(IOUT,562) (PID(IIP),IIP=1,NP)
      WRITE(IOUT,575) (XX(IIP),IIP=1,NP)
C--------IF IPAR=0, RETURN
      IF(IPAR.EQ.0) GO TO 340
C--------SAVE GRADIENT FOR THE FIRST ITERATION
      IF(IITERP.EQ.1) THEN
        DO 140 IIP=1,NP
  140   G(IIP)=XX(IIP)
      ENDIF
C-----INITIALIZE
      AGMX=0.
C-----THE FOLLOWING IS DONE ONLY IF IITERP>1.
      IF(IITERP.GT.1) THEN
        XD=0.
C--------COMPUTE MAXIMUM NORM OF GRADIENT (AGMX), GRADIENT DIFFERENCE
C--------(STORE IN XX), AND XX*DD (XD). STORE MOST RECENT GRADIENT IN G.
        DO 150 IIP=1,NP
          TEMP=XX(IIP)
          TMP=ABS(TEMP)
          IF(TMP.GT.AGMX) AGMX=TMP
          XX(IIP)=TEMP-G(IIP)
          XD=XD+XX(IIP)*DD(IIP)
  150   G(IIP)=TEMP
C--------CHECK FOR CONVERGENCE
        IF(ADMX.LE.TOL.AND.AGMX.LE.TOL) THEN
          IFO=1
          WRITE(IOUT,525) ADMX,AGMX
          IF(ISN1.EQ.10) LASTX=1
C-----------IF GAUSS-NEWTON STATISTICS ARE NOT CALCULATED . . . .
C-------------CALCULATE AND PRINT STATISTICS AND RESIDUALS
          IF(LASTX.EQ.0) THEN
            CALL SPAR1A(ND,WT,HOBS,H,R,R1,NPR,MPR,B1,NP,WP,B,PRM,NPO)
            DO 160 IIP=1,NP
  160       IF(LN(IIP).NE.0) B(IIP)=EXP(B(IIP))
            CALL SSEN1O(NP,B,B1,IWPG,PID,WP,IOUT,NH,H,HOBS,WT,DID,NDER,
     2                  ROFF,COFF,TOFF,NQT,IQOB,ND,IPAR,NPR,MPR,PRM,
     3                  RSQ,RSQP,1,LN,RSQH,NPO)
            IF(IOUB.GT.0) CALL SPAR1P(PID,BUFF,NP,IOUB,IOUT,ISN)
            VAR=RSQP/(ND-IDRY+NPR+MPR-NP)
            WRITE(IOUT,521) RSQP,VAR,R,R1,ITERP
            CALL SPAR1M(RSQP,WT,NP,ND,WP,MPR,IOUT,EV,NPR)
C-------------PRINT QUASI-NEWTON MATRIX, IF USED ON LAST ITERATION
            IF(ITERP.GT.NFIT) THEN
              WRITE(IOUT,585)
```

```
            IP1=1
            IP2=0
            DO 180 IPP=1,NP
              IF(IPP.GT.1) IP1=IP1+(NP-(IPP-1)+1)
              IP2=IP2+(NP-IPP+1)
  180         WRITE(IOUT,587)(C(IIP),IIP=IP1,IP2)
          ENDIF
          RETURN
        ENDIF
        GO TO 340
      ENDIF
C--------EXCEEDED MAXIMUM PARAMETER-ESTIMATION ITERATIONS?
      IF(ITERP.EQ.ITMXP.OR.(ISN1.EQ.10.AND.ITERP.EQ.ITMXP-1)) THEN
        WRITE(IOUT,595)
        DO 185 IIP=1,NP
  185   IF(LN(IIP).NE.0) B(IIP)=EXP(B(IIP))
        CALL SSEN1O(NP,B,B1,IWPG,PID,WP,IOUT,NH,H,HOBS,WT,DID,NDER,
     2              ROFF,COFF,TOFF,NQT,IQOB,ND,IPAR,NPR,MPR,PRM,
     3              RSQ,RSQP,1,LN,RSQH,NPO)
        IF(IOUB.GT.0) CALL SPAR1P(PID,BUFF,NP,IOUB,IOUT,ISN)
        STOP
      ENDIF
C--------IF XD IS ZERO OR NEGATIVE, DON'T CALCULATE A NEW HESSIAN MATRIX
C--------USE SAME HESSIAN AND THE NEW GRADIENT TO CALCULATE DISPLACEMENT
      IF(XD.LE.0.) WRITE(IOUT,590) ITERP+1,XD
    ENDIF
C
C-----COMPUTE DISPLACEMENT VECTOR
C-------FLETCHER-REEVES METHOD
    IF(ITERP.LE.NFIT.AND.ISN1.NE.10) THEN
      IF(ITERP.EQ.1) WRITE(IOUT,550)
      SSG=0.
      DO 190 IIP=1,NP
  190 SSG=SSG+G(IIP)*G(IIP)
      IF(ITERP.EQ.1.OR.MOD(ITERP-1,NP).EQ.0) THEN
        DO 200 IIP=1,NP
  200   DD(IIP)=-G(IIP)
        WRITE(IOUT,560)
      ELSE
        BETA=SSG/SSX
        DO 210 IIP=1,NP
  210   DD(IIP)=-G(IIP)+BETA*DD(IIP)
        WRITE(IOUT,555) SSX,SSG,BETA
      ENDIF
      SSX=SSG
      GO TO 330
    ENDIF
C--------QUASI-NEWTON METHOD
C----------INITIAL APPROXIMATE INVERSE HESSIAN MATRIX
    IF(IITERP.EQ.NFIT+1.OR.(ISN1.EQ.10.AND.IITERP.EQ.1)) THEN
      WRITE(IOUT,545)
      K=0
      IF(ISN1.EQ.10) THEN
        WRITE(IOUT,620)
        CALL SPAR1I(NP,CC)
        DO 215 IIP=1,NP
        DO 215 I=IIP,NP
          K=K+1
  215     C(K)=-2.*CC(I,IIP)
        GO TO 295
      ENDIF
      DO 230 IIP=1,NP
        KT=K+1
        DO 220 I=IIP,NP
          K=K+1
  220   C(K)=0.
        BDP=B(IIP)
        SCLDP=SCL(IIP)
        LNN=LN(IIP)
        IF(LNN.EQ.0) BDP=BDP/SCLDP
        IF(LNN.NE.0) BDP=BDP-ALOG(SCL(IIP))
  230 C(KT)=(BDP)**2
    ENDIF
C---------SUBSEQUENT APPROXIMATE INVERSE HESSIAN MATRIX
    IF(IITERP.GT.NFIT+1.OR.(ISN1.EQ.10.AND.IITERP.NE.1)) THEN
      K=0
      XHX=0.
      DO 270 IIP=1,NP
        SUM=0.
        M=NP-IIP+1
        L=K+1
        DO 240 I=1,M
          K=K+1
```

```
  240      SUM=SUM+C(K)*XX(I+IIP-1)                                          PAR0574
           N=IIP-1                                                           PAR0575
           IF(N.LT.1) GO TO 260                                              PAR0576
           DO 250 I=1,N                                                      PAR0577
             L=L-M                                                           PAR0578
             SUM=SUM+C(L)*XX(N-I+1)                                          PAR0579
  250      M=M+1                                                             PAR0580
  260      BUFF(IIP)=SUM                                                     PAR0581
  270    XHX=XHX+XX(IIP)*SUM                                                 PAR0582
         GAM=1.                                                              PAR0583
         IF(ITG.LT.2) GAM=AP*XD/XHX                                          PAR0584
         TEMP=(AP+GAM*XHX/XD)/XD                                             PAR0585
         K=0                                                                 PAR0586
         DO 290 IIP=1,NP                                                     PAR0587
           DO 280 I=IIP,NP                                                   PAR0588
             K=K+1                                                           PAR0589
  280        C(K)=(C(K)-(BUFF(I)*DD(IIP)+DD(I)*BUFF(IIP))/XD)*GAM+           PAR0590
     1                              TEMP*DD(I)*DD(IIP)                       PAR0591
  290    CONTINUE                                                            PAR0592
         ITG=2                                                               PAR0593
       ENDIF                                                                 PAR0594
C----------DISPLACEMENT VECTOR                                                PAR0595
  295 K=0                                                                    PAR0596
      DO 320 IIP=1,NP                                                        PAR0597
        SUM=0.                                                               PAR0598
        M=NP-IIP+1                                                           PAR0599
        L=K+1                                                                PAR0600
        DO 300 I=1,M                                                         PAR0601
          K=K+1                                                              PAR0602
  300     SUM=SUM+C(K)*G(I+IIP-1)                                            PAR0603
        N=IIP-1                                                              PAR0604
        IF(N.LT.1) GO TO 320                                                 PAR0605
        DO 310 I=1,N                                                         PAR0606
          L=L-M                                                              PAR0607
          SUM=SUM+C(L)*G(N-I+1)                                              PAR0608
  310     M=M+1                                                              PAR0609
  320 DD(IIP)=-SUM                                                           PAR0610
  330 CONTINUE                                                               PAR0611
C-------PRINT RESULTS                                                        PAR0612
      WRITE(IOUT,570)                                                        PAR0613
      WRITE(IOUT,575) (DD(IIP),IIP=1,NP)                                     PAR0614
  340 IF(IFO.GT.0.AND.IOUB.GT.0) THEN                                        PAR0615
        ZERO=0.                                                              PAR0616
        IZERO=0                                                              PAR0617
        WRITE(IOUB,600) ZERO,IZERO,ZERO                                      PAR0618
      ENDIF                                                                  PAR0619
C-------CONVERT TO PHYSICAL PARAMETERS FROM NATURAL LOGS                     PAR0620
      DO 350 IIP=1,NP                                                        PAR0621
  350 IF(LN(IIP).NE.0) B(IIP)=EXP(B(IIP))                                    PAR0622
      IF(ISN1.EQ.10) ITERP=ITERP+1                                           PAR0623
      RETURN                                                                 PAR0624
      END                                                                    PAR0625


C=======================================================================PAR0626
      SUBROUTINE PAR1AQ(                                                     PAR0627
     1  ND,HOBS,H,B1,B,DD,NP,WT,WP,PID,HNEW,NCOL,NROW,NLAY,ISN,IP,SCL,      PAR0628
     2  PV,NH,NDER,IOFF,JOFF,KPER,MLAY,IBOUND,RINT,DID,COFF,ROFF,DELR,      PAR0629
     3  DELC,PR,MOBS,LN,NQT,TOFF,NQ,NQC,NQOB,NQCL,IQOB,QCLS,IBT,             PAR0630
     4  MXBND,NBOUND,BNDS,MXRIVR,NRIVER,RIVR,SHNW,NPER,MXDRN,NDRAIN,        PAR0631
     5  DRAI,MXSTRM,NSTREM,STRM,ISTRM,ADMX,PRM,MPR,IOUT,DMX,DMAX,AP,        PAR0632
     6  NPR,ITERP,AGMX,MAXM,JDRY,IPAR,IOUB,NPO)                              PAR0633
C-----VERSION 1000 01FEB1992 PAR1AQ                                          PAR0634
C     ******************************************************************PAR0635
C     NONLINEAR REGRESSION BY CONJUGATE-DIRECTION METHODS:                   PAR0636
C     CALCULATE FINAL ITERATION PARAMETER AND UPDATE PARAMETERS              PAR0637
C     ******************************************************************PAR0638
C     SPECIFICATIONS:                                                        PAR0639
C     ------------------------------------------------------------------PAR0640
      CHARACTER*4 PID(NP),DID(NH+NQT)                                        PAR0641
      DOUBLE PRECISION HNEW(NCOL,NROW,NLAY),DD(NP),SWS,SWY,DDSCL,            PAR0642
     1                 TMP,F,PV,SCLDP                                        PAR0643
      DIMENSION B(NPO),B1(NPO),HOBS(ND),H(ND),WT(ND),WP(NPO+MPR),            PAR0644
     1      SCL(NPO),NDER(5,NH),DELR(NCOL),DELC(NROW),                       PAR0645
     1      IBOUND(NCOL,NROW,NLAY),COFF(NH),ROFF(NH),RINT(4,NH),JOFF(NH),PAR0646
     1      IOFF(NH),MLAY(MAXM,MOBS),PR(MAXM,MOBS),LN(NP),RIVR(6,MXRIVR),PAR0647
     1      BNDS(5,MXBND),IBT(2,NQ),NQOB(NQ),IQOB(NQT),QCLS(4,NQC),          PAR0648
     1      TOFF(NH+NQT),DRAI(5,MXDRN),PV(ND),STRM(11,MXSTRM),               PAR0649
     1      ISTRM(5,MXSTRM),PRM(NPO+1,MPR),NQCL(NQ)                          PAR0650
C     ------------------------------------------------------------------PAR0651
C                                                                            PAR0652
  514 FORMAT (/,                                                             PAR0653
     $' VALUES FROM CONJUGATE-DIRECTION REGRESSION PROCEDURE, ',             PAR0654
```

```
     $'ITERATION',I3,' :',/
     $ 2X,' ITERATION PARAMETER (ALPH)          = ',G11.5,/,
     $ 2X,' FACTOR FOR SCALING PAR. CHANGE (AP)= ',G11.5,/,
     $ 2X,' MAX. FRACTIONAL PAR. CHANGE (DMX)  = ',G11.5,/,
     $ 2X,' MAX. FRAC. CHANGE OCCURRED FOR PAR.# ',I6,/,
     1 2X,' MAX. NORM OF GRADIENT       (AGMX) = ',G11.6,/,
     1 /' UPDATED ESTIMATES OF REGRESSION PARAMETERS :',/
     1,13(5X,A4,9X,A4,9X,A4,9X,A4,9X,A4,9X,A4,
     1    9X,A4,9X,A4,9X,A4,9X,A4,/))
  530 FORMAT(E10.3,I5,/,E10.3)
  540 FORMAT(8G10.3)
  570 FORMAT(/
     $,2X,' ITERATION PARAMETER                    = ',G11.5,/
     $,2X,' NUMERATOR OF ITERATION PARAMETER       = ',G11.5,/
     $,2X,' DENOMINATOR OF ITERATION PARAMETER  = ',G11.5,/)
  575 FORMAT (1H0,10(1H ,G11.5,1X))
C-------INTERPOLATE ITERATION PARAMETER VECTOR TO HEAD LOCATIONS
      IF(NH.GT.0) THEN
        IF(KPER.EQ.1) THEN
          DO 10 N=1,NH
   10     PV(N)=0.D0
        ENDIF
        CALL SSEN1U(NH,NDER,IOFF,JOFF,KPER,MLAY,IBOUND,RINT,
     1    DID,COFF,ROFF,DELR,DELC,NCOL,NROW,NLAY,PR,H,HOBS,WT,HNEW,
     2    X,IP,IDRY,NP,MOBS,B,LN,ISN,NQT,TOFF,NPER,RHS,PV,
     3    MAXM,ITERP,JDRY,IPAR,IOUT)
      ENDIF
C-------CALCULATE ITERATION PARAMETER VECTOR VALUES FOR HEAD-DEPENDENT
C-------FLOW BOUNDARY DATA
      IF(NQ.GT.0) THEN
        ITS=KPER-1
C----------INITIALIZE COMPONENTS OF PV
        IF(ITS.EQ.0) THEN
          DO 30 NT=1,NQT
   30     PV(NH+NT)=0.
        ENDIF
        CALL SSEN1V(NQ,NQC,NQT,NQOB,NQCL,IQOB,QCLS,IBT,MXBND,NBOUND,
     1    BNDS,MXRIVR,NRIVER,RIVR,SHNW,IP,HNEW,NCOL,NROW,NLAY,
     1    IOUT,IBOUND,NPER,KPER,NH,PV,DID,NP,FLAGD,H,B,LN,TOFF,
     1    MXDRN,NDRAIN,DRAI,MXSTRM,NSTREM,STRM,ISTRM,ISN,RHS,WT,HOBS)
      ENDIF
      IF(KPER.NE.NPER) RETURN
C-----PARAMETER CONVERSIONS FOR NATURAL LOG OPTIONS
      DO 50 IIP=1,NP
   50 IF(LN(IIP).NE.0) B(IIP)=ALOG(B(IIP))
C-------CALCULATE ITERATION PARAMETER
C----------INITIALIZE VARIABLES
      SWS=0.D0
      SWY=0.D0
C----------DEPENDENT VARIABLES
      DO 220 N=1,ND
        F=PV(N)
        W=WT(N)
        IF(W.LT.0.) W=1.E-20
        TMP=F*W
        TMPA=HOBS(N)-H(N)
        SWS=SWS+TMP*F
  220 SWY=SWY+TMP*TMPA
C----------PRIOR ESTIMATES OF SINGLE PARAMETERS
      IF(NPR.GT.0) THEN
        DO 230 IIP=1,NP
          IF(WP(IIP).GT.0.) THEN
            LNN=LN(IIP)
            SCLL=SCL(IIP)
            DDSCL=DD(IIP)*SCLL
            TMPA=B1(IIP)-B(IIP)
            TMP=DDSCL*WP(IIP)
            SWS=SWS+TMP*DDSCL
            SWY=SWY+TMP*TMPA
          ENDIF
  230   CONTINUE
      ENDIF
C----------PRIOR ESTIMATES OF PARAMETER SUMS
      IF(MPR.GT.0) THEN
        TMPA=0.
        SCLDP=1.D0
        DO 300 IPM=1,MPR
          DO 298 IIP=1,NPO
            IF(PRM(IIP,IPM).EQ.0.) GO TO 298
            IF(SCL(IIP).NE.1.) SCLDP=SCL(IIP)
            TMPA=TMPA+PRM(IIP,IPM)*B(IIP)
  298     CONTINUE
          TMPA=PRM(NPO+1,IPM)-TMPA
```

```
         DO 299 IIP=1,NPO
           IF(PRM(IIP,IPM).EQ.0.) GO TO 299
           SCLDP=SCL(IIP)
           DDSCL=DD(IIP)*PRM(IIP,IPM)*SCLDP
           IF(LN(IIP).NE.0) DDSCL=DD(IIP)+ALOG(SCL(IIP))
           TMP=DDSCL*WP(NPO+IPM)
           SWS=SWS+TMP*DDSCL
           SWY=SWY+TMP*TMPA
  299    CONTINUE
  300  CONTINUE
      ENDIF
C-------ITERATION PARAMETER
      ALPH=SWY/SWS
C-------UPDATE PARAMETERS
      ADMX=0.
      DO 255 IIP=1,NP
        LNN=LN(IIP)
        SCLL=SCL(IIP)
        BB=B(IIP)
        TMPA=1.
        IF(BB.NE.0.) TMPA=BB
        IF(LNN.NE.0.AND.ABS(TMPA).LT..00001) TMPA=1.
        TMPA=TMPA/SCLL
        TMPA=(ALPH*DD(IIP)/TMPA)
        IF(LNN.GT.0) TMPA=EXP(ALPH*DD(IIP))-1.
        TMP=ABS(TMPA)
        IF((LNN.EQ.0.AND.TMP.LE.ADMX).OR.
     1     (LNN.NE.0.AND.TMPA.LE.ADMX)) GO TO 255
          ADMX=TMP
          JJ=IIP
          DMX=TMPA
          LNMX=0
          IF(LNN.NE.0) LNMX=1
  255 CONTINUE
      AP=1.
        IF(LNMX.EQ.0.AND.ADMX.GT.DMAX) AP=DMAX/ADMX
        IF(LNMX.NE.0.AND.ALPHA*DD(JJ).GT.ALOG(DMAX+1.))
     1          AP=ALOG(DMAX+1.)/(ALPHA*DD(JJ))
C-------COMPUTE NEW PARAMETER VECTOR
      DO 260 IIP=1,NP
        LNN=LN(IIP)
        SCLL=SCL(IIP)
        IF(LNN.EQ.0) DDSCL=DD(IIP)*SCLL
        IF(LNN.NE.0) DDSCL=DD(IIP)
  260 B(IIP)=B(IIP)+AP*ALPH*DDSCL
C-----PARAMETER CONVERSIONS FOR NATURAL LOG OPTIONS
      DO 270 IIP=1,NP
  270 IF(LN(IIP).NE.0) B(IIP)=EXP(B(IIP))
C-----PRINT RESULTS
      WRITE(IOUT,514) ITERP,ALPH,AP,DMX,JJ,AGMX,(PID(IIP),IIP=1,NP)
      WRITE(IOUT,575) (B(IIP),IIP=1,NP)
      IF(IOUB.GT.0) THEN
        WRITE(IOUB,530) DMX,JJ,AGMX
        WRITE(IOUB,540) (B(IIP),IIP=1,NP)
      ENDIF
      AP=ALPH*AP
      RETURN
      END


C=======================================================================
      SUBROUTINE PAR1AP(
     1     X,B,ND,NP,HOBS,WT,WP,C,SCLE,G,H,DD,DMAX,CSA,TOL,PID,IND,IFO,
     2     AMP,AP,DMX,IOUT,B1,ITERP,IPRINT,ADMX,LN,LASTX,MPR,PRM,NPR,
     3     IOUB,RSQP,JMAX,ITMXP,ISN1,NFIT,NPO,R,GD,U,NOPT,XD,NPER,
     4     S,SOSR)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     REGRESSION BY THE MODIFIED GAUSS-NEWTON METHOD
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4   PID(NP)
      DOUBLE PRECISION C(NP,NP),SCLE(NP),DD(NP),DTMPA,R(NP*NP/2+NP),
     1       S(NP),U(NP),G,GD,DTMP
      DIMENSION X(NP,ND),B(NPO),HOBS(ND),WT(ND),WP(NPO+MPR),H(ND),
     1           B1(NPO),G(NP),LN(NP),PRM(NPO+1,MPR),GD(NP),XD(NP,ND)
C     ------------------------------------------------------------------
  500 FORMAT(/,' SCALED LEAST-SQUARES MATRIX :')
  508 FORMAT(/,' UNSCALED RIGHT-HAND SIDE VECTOR (= -0.5*GRADIENT)')
  510 FORMAT(/,' SCALED GRADIENT VECTOR :')
  514 FORMAT (/,' ITERATION NO. = ',I5,/,
     $' VALUES FROM LEAST-SQUARES REGRESSION PROCEDURE :',/
```

```
     $,2X,' DET OF SCALED LEAST-SQUARES MATRIX = ',G11.5,/,
     $ 2X,' MARQUARDT PARAMETER (AMP)--------- = ',G11.5,/,
     $ 2X,' FACTOR FOR SCALING PAR. CHANGE (AP)= ',G11.5,/,
     $ 2X,' MAX. FRACTIONAL PAR. CHANGE (DMX)  = ',G11.5,/,
     $ 2X,' MAX. FRAC. CHANGE OCCURRED FOR PAR.# ',I6,/
     $/' UPDATED ESTIMATES OF REGRESSION PARAMETERS :',/
     1,13(5X,A4,9X,A4,9X,A4,9X,A4,9X,A4,9X,A4,
     1    9X,A4,9X,A4,9X,A4,9X,A4,/))
  516 FORMAT (1H0,10(1H ,G11.5,1X))
  530 FORMAT(E10.3,I5)
  600 FORMAT(/,' PARAMETER ESTIMATION CONVERGED BY SATISFYING TOL',
     1   ' CRITERIA',/)
C
C-------INITIALIZE SOME VARIABLES.
      NP1=NP-1
      AMP=0.
      IND=0
C-------ASSEMBLE LEAST-SQUARES MATRIX (C) AND GRADIENT VECTOR (G)
C----------INITIALIZE C AND CONVERT LN PARAMETERS
      DO 60 IP=1,NP
        DO 50 I=1,NP
   50   C(I,IP)=0.D00
        IF(LN(IP).NE.0) B(IP)=ALOG(B(IP))
   60   G(IP)=0.
C----------CALCULATE SENSITIVITY CONTRIBUTIONS TO C AND G.
C----------CALCULATE SUM OF SQUARED RESIDUALS.
      DO 90 N=1,ND
        TMPA=HOBS(N)-H(N)
        W=WT(N)
        IF(W.LT.0.) THEN
          W=1.E-20
          WT(N)=-WT(N)
        ENDIF
        DO 80 IP=1,NP
          DTMPA=DBLE(W)*DBLE(X(IP,N))
          DO 70 I=IP,NP
   70       C(I,IP)=DBLE(X(I,N))*DTMPA+C(I,IP)
   80     G(IP)=DTMPA*TMPA+G(IP)
   90 CONTINUE
C----------FOR FIRST ITERATION, PRINT UNSCALED GRADIENT WITHOUT PRIOR
      IF(ITERP.EQ.1) THEN
        WRITE(IOUT,508)
        WRITE(IOUT,516) (G(IP),IP=1,NP)
      ENDIF
C----------ACCOUNT FOR PRIOR INFORMATION
C-------------SINGLE PARAMETER ESTIMATES
      IF(NPR.GT.0) THEN
        DO 93 IP=1,NP
          IF(WP(IP).GT.0.) THEN
            C(IP,IP)=C(IP,IP)+WP(IP)
            G(IP)=G(IP)+WP(IP)*(B1(IP)-B(IP))
          ENDIF
   93   CONTINUE
      ENDIF
C-------------ESTIMATES OF PARAMETER SUMS
      IF(MPR.GT.0) THEN
        DO 97 IPM=1,MPR
          TMPA=0.
          DO 94 IP=1,NPO
   94     IF(PRM(IP,IPM).GT.0.) TMPA=TMPA+PRM(IP,IPM)*B(IP)
          TMPA=PRM(NPO+1,IPM)-TMPA
          DO 96 IP=1,NPO
            IF(PRM(IP,IPM).EQ.0.) GO TO 96
            DTMPA=DBLE(WP(NPO+IPM))*DBLE(PRM(IP,IPM))
            DO 95 I=IP,NPO
   95         C(I,IP)=DBLE(PRM(I,IPM))*DTMPA+C(I,IP)
            G(IP)=DTMPA*TMPA+G(IP)
   96     CONTINUE
   97   CONTINUE
      ENDIF
C-------QUASI-NEWTON ADDITION TO COEFFICIENT MATRIX
      IF((NFIT.GT.0.OR.SOSR.GT.0.).AND.NOPT.EQ.1.AND.IFO.EQ.0)
     1  CALL SPAR1Q(C,DD,G,NP,R,GD,U,ITERP,1,X,ND,HOBS,H,WT,
     2           S,NFIT,B1,B,WP,NPR,MPR,PRM,NPO,XD,NPER,SOSR)
C-------FOR ONE PARAMETER CASE
      IF(NP.LT.2) GO TO 140
C-------SCALE COEFFICIENT MATRIX AND GRADIENT VECTOR
      DO 100 IP=1,NP
        SCLE(IP)=1.D00
  100   IF(C(IP,IP).GT.1.D-30) SCLE(IP)=DSQRT(C(IP,IP))
      DO 120 IP=1,NP1
        DTMPA=SCLE(IP)
        IP1=IP+1
```

```
        DO 110 I=IP1,NP
          C(I,IP)=C(I,IP)/(SCLE(I)*DTMPA)
  110     C(IP,I)=C(I,IP)
        G(IP)=G(IP)/DTMPA
  120   C(IP,IP)=1.D00+AMP
      G(NP)=G(NP)/SCLE(NP)
      C(NP,NP)=1.D00+AMP
C-------PRINT AS INDICATED BY IPRINT
      IF(IPRINT.NE.0) THEN
        WRITE(IOUT,500)
        DO 125 J=1,NP
  125   WRITE(IOUT,516)(C(I,J),I=1,NP)
        WRITE(IOUT,510)
        WRITE(IOUT,516)(G(I),I=1,NP)
      ENDIF
C-------COMPUTE PARAMETER STEP LENGTHS
      CALL SPAR1R(C,DD,G,NP,CSA,IND,IFO,AMP,DET)
C-------IF MATRIX EQUATION IS SINGULAR, OR CONVERGENCE HAS BEEN REACHED:
      IF(IND.GT.0.OR.IFO.GT.0) THEN
C---------CONVERT NATURAL LOGS OF PARAMETER VALUES AND RETURN
  126   DO 127 IP=1,NP
  127   IF(LN(IP).NE.0) B(IP)=EXP(B(IP))
        RETURN
      ENDIF
C-------UNSCALE PARAMETER CHANGE VECTOR
      DO 130 IP=1,NP
  130   DD(IP)=DD(IP)/SCLE(IP)
      GO TO 150
C-------CALCULATE STEP LENGTH FOR SINGLE-PARAMETER CASE
  140 DET=C(1,1)
      SCLE(1)=1.D00
      IF(IFO.GT.0) THEN
        IF(LN(1).GT.0) B(1)=EXP(B(1))
        RETURN
      ENDIF
      DD(1)=G(1)/DET
C-------COMPUTE DAMPING PARAMETER AND NEW ESTIMATES OF REGRESSION
C-------PARAMETERS
  150 ADMX=0.
      ADMX1=0.
      DMXO=DMX
      JJ=0
      DO 160 IP=1,NP
        TMPA=0.
        TMPB=0.
        LNN=LN(IP)
        IF(LNN.EQ.0) THEN
          TMPA=B(IP)
          IF(ABS(B(IP)).LT.ABS(B1(IP))*1.E-3) TMPA=B1(IP)
          TMPA=DD(IP)/ABS(TMPA)
          TMPB=ABS(TMPA)
        ENDIF
        IF(LNN.GT.0) THEN
          IF(DD(IP).GE.0.) THEN
            TMPA=1.E+10
            IF(DD(IP).LT.23.) TMPA=DEXP(DD(IP))-1.
          ELSEIF(DD(IP).LT.0.) THEN
            TMPA=-1.
            IF(DD(IP).GT.-23.) TMPA=DEXP(DD(IP))-1.
          ENDIF
          TMPB=ABS(TMPA)
        ENDIF
        IF(NP.GT.1.AND.TMPB.LE.ADMX) GO TO 160
          IF(LN(IP).GT.0.AND.DD(IP).LT.0..AND.DMAX.GE.1..AND.
     1      TMPB.GT.ADMX1) THEN
            JJ1=IP
            DMX1=TMPA
            ADMX1=TMPB
            GO TO 160
          ENDIF
          JJ=IP
          DMX=TMPA
          ADMX=TMPB
          LNMX=0
          IF(LNN.NE.0) LNMX=1
  160 CONTINUE
      IF(ADMX1.GT.ADMX) THEN
        DMX=DMX1
        JJ=JJ1
        ADMX=ADMX1
        LNMX=1
      ENDIF
      IF(ITERP.LE.1.OR.JJ.NE.JMAX.OR.
```

```
     1  (ISN1.EQ.10.AND.ITERP.EQ.NFIT+2)) GO TO 166
          SPR=DMX/(AP*DMXO)
          IF(SPR.LT.-1.) GO TO 164
            AP=(3.+SPR)/(3.+ABS(SPR))
            GO TO 166
  164     AP=.5/ABS(SPR)
  166     CONTINUE
        JMAX=JJ
        IF(ITERP.EQ.1.OR.JJ.EQ.0.OR.JJ.NE.JMAX.OR.
     1      (ISN1.EQ.10.AND.ITERP.EQ.NFIT+2)) AP=1.
        IF(LNMX.EQ.0.AND.AP*ADMX.GT.DMAX) AP=DMAX/ADMX
        IF(LNMX.NE.0) THEN
          DTMP=DD(JJ)
          IF(AP*DTMP.GT.ALOG(DMAX+1.)) AP=ALOG(DMAX+1.)/DTMP
          IF(DMAX.LT.1..AND.AP*DTMP.LT.ALOG(1.-DMAX))AP=ALOG(1.-DMAX)/DTMP
        ENDIF
        DO 170 IP=1,NP
          DDC=AP*DD(IP)
          B(IP)=B(IP)+DDC
          DD(IP)=DDC
          IF(LN(IP).NE.0) B(IP)=EXP(B(IP))
  170 CONTINUE
C-------PRINT DATA FROM CURRENT ITERATION
      WRITE(IOUT,514) ITERP,DET,AMP,AP,DMX,JJ,(PID(IP),IP=1,NP)
      WRITE(IOUT,516) (B(IP),IP=1,NP)
      IF(IOUB.GT.0) WRITE(IOUB,530) DMX,JJ,AMP
      AMP=0.
C-------CHECK FOR CONVERGENCE
      IF(ADMX.LE.TOL) THEN
        IFO=1
        WRITE(IOUT,600)
      ENDIF
      RETURN
      END


C=======================================================================
      SUBROUTINE PAR1OT(
     1                  C,WT,NP,IOUR,RSQ,IOUT,BUFF,ND,IPRC,IFO,IND,PID,
     2                  SCLE,B,HOBS,H,NPR,B1,WP,ITERPF,X,HNEW,LN,MPR,
     3                  PRM,IWPG,NODES,LPRINT,IDRY,EV,RSQP,ISN,NPO)
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     FINAL GAUSS-NEWTON OUTPUT FOR PARAMETER ESTIMATION
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 PID(NP),ANAME
      DOUBLE PRECISION C(NP,NP),SCLE(NP),HNEW(NODES),DSUM,TMP
      DIMENSION BUFF(NP,NP+2),ANAME(4,2),WT(ND),B(NPO),HOBS(ND),H(ND),
     1   WP(NPO+MPR),B1(NPO),X(NP,ND),LN(NP),PRM(NPO+1,MPR),IWPG(NP)
      DATA ANAME(1,1),ANAME(2,1),ANAME(3,1),ANAME(4,1)
     1 /'COVA','RIAN','CE M','AT. '/
      DATA ANAME(1,2),ANAME(2,2),ANAME(3,2),ANAME(4,2)
     1 /'CORR','ELAT','ION ','MAT.'/
C     ------------------------------------------------------------------
C
  520 FORMAT(//,' LEAST SQUARES COEFFICIENT MATRIX SINGULAR -- '
     1      'STOP EXECUTION (PAR1OT)')
  521 FORMAT (//,'  RSQ (DEP.VAR. ONLY)------ = ',G11.5,/
     1          ,'  RSQ (W/PARAMETERS)------- = ',G11.5,/
     1          ,'  CALCULATED ERROR VARIANCE = ',G11.5,/,
     1           '  CORRELATION COEFFICIENT-- = ',G11.5,/
     1          ,'          W/PARAMETERS-------- = ',G11.5,/
     1          ,'  ITERATIONS--------------- = ',I7)
  525 FORMAT('  ',I5,' CELLS WERE DRY IN FINAL PARAMETER-ESTIMATION ',
     1      'ITERATION')
  530 FORMAT(//,' MAXIMUM NUMBER OF PARAMETER ESTIMATION ITERATIONS',
     1 ' EXCEEDED -- STOP EXECUTION (PAR1OT)')
  535 FORMAT(/' THE CORRELATION OF THE FOLLOWING PARAMETER PAIRS >=',
     1    ' .95',/,5X,'PARAMETER  #   ID     #   ID   CORRELATION')
  540 FORMAT(13X,I5,2X,A4,I5,2X,A4,F8.2)
  545 FORMAT(/' THE CORRELATION OF THE FOLLOWING PARAMETER PAIRS IS ',
     1    'BETWEEN .90 AND .95',/,5X,'PARAMETER  #   ID     #   ID',
     1    '   CORRELATION')
  550 FORMAT(/' THE CORRELATION OF THE FOLLOWING PARAMETER PAIRS IS ',
     1    'BETWEEN .85 AND .90',/,5X,'PARAMETER  #   ID     #   ID',
     1    '   CORRELATION')
555   FORMAT(//,' PARAMETER SUMMARY'/,
     1       '  ________________',//,' PARAMETER # :',3X,I6,9I10,/
     1    ,12(14X,3X,I6,9I10,/))
557   FORMAT(' PARAMETER ID :',7X,A4,6X,A4,6X,A4,6X,A4,6X,A4,6X,A4,
     1    6X,A4,6X,A4,6X,A4,6X,A4,/,12(15X,7X,A4,6X,A4,6X,A4,6X,A4,
```

```
     1    6X,A4,6X,A4,6X,A4,6X,A4,6X,A4,6X,A4,/))
  558 FORMAT(' FINAL VALUES')
  559 FORMAT('    EXPONENTIAL OF LN PARAMETERS',/,
     1        '    (0.0 FOR UNTRANSFORMED PARAMETERS)')
  560 FORMAT(' STD. DEV.')
  562 FORMAT(' COEF. OF VAR.')
  565 FORMAT(17X,10E10.3,/,12(17X,10E10.3,/))
  570 FORMAT(/,10X,4A4/10X,16('-'))
  580 FORMAT(16G13.6)
  585 FORMAT(16D13.6)
  590 FORMAT(' ')
  600 FORMAT(/,' + 2 STD. DEV.')
  610 FORMAT(/,' - 2 STD. DEV.')
C
C-----SINGULAR COEFFICIENT MATRIX
      IF(IND.GT.0) THEN
        WRITE(IOUT,520)
        STOP
      ENDIF
C-----EXCEEDED MAXIMUM NUMBER OF PARAMETER-ESTIMATION ITERATIONS
      IF(IFO.EQ.0) THEN
        WRITE(IOUT,530)
        STOP
      ENDIF
C-------COMPLETE COMPUTATION OF INVERSE OF THE SCALED COEFFICIENT
C-------MATRIX C STARTING FROM DECOMPOSED MATRIX FROM SPAR1R.
      IF(NP.LT.2) THEN
        C(1,1)=1./C(1,1)
        GO TO 290
      ENDIF
      CALL SPAR1I(NP,C)
C-------COMPUTE FINAL CALCULATED ERROR VARIANCE
  290 VAR=RSQP/(ND-IDRY+NPR+MPR-NP)
C-------COMPUTE VARIANCE-COVARIANCE MATRIX FOR PARAMETERS BY UNSCALING
C-------C-INVERSE AND MULTIPLYING BY THE FINAL ERROR VARIANCE.
      DO 320 IP=1,NP
        DTMPA=SCLE(IP)
        DO 310 I=IP,NP
          C(I,IP)=VAR*C(I,IP)/(SCLE(I)*DTMPA)
          C(IP,I)=C(I,IP)
  310   CONTINUE
        SCLE(IP)=DSQRT(C(IP,IP))
  320 CONTINUE
C-------PRINT VARIANCE-COVARIANCE MATRIX.
      DO 325 IP=1,NP
        DO 325 I=1,NP
  325   BUFF(I,IP)=C(I,IP)
      IDUM=1
      WRITE(IOUT,570) (ANAME(I,1),I=1,4)
      CALL ULAPRW(BUFF,ANAME(1,1),IDUM,IDUM,NP,NP,-IDUM,IPRC,IOUT)
C-------EIGENVALUES AND EIGENVECTORS OF COV MATRIX SCALED BY PARAMETERS
      IF(LPRINT.NE.0) CALL SPAR1E(C,HNEW(1),HNEW(NP*NP+1),
     1       NP,HNEW(2*NP*NP+1),IOUT,B,B1,ITERPF,BUFF,PID,IPRC,LPRINT)
C-------WRITE INPUT FOR RESIDUAL ANALYSIS PROGRAM
      IF(IOUR.GT.0) THEN
        DO 330 IP=1,NP
  330   WRITE(IOUR,585) (C(I,IP),I=IP,NP)
        WRITE(IOUR,580) (WT(N),N=1,ND)
        DO 335 N=1,ND
  335   WRITE(IOUR,580) (X(IP,N),IP=1,NP)
      ENDIF
C-------PRINT FINAL PARAMETER VALUES, STD. DEV., COEFFICIENTS OF
C-------VARIATION, AND PARAMETER VALUES +- 2 STANDARD DEVIATIONS
C---------LOG TRANSFORM
      LNFLAG=0
      DO 200 IP=1,NP
        IF(LN(IP).NE.0) THEN
          LNFLAG=1
          B(IP)=ALOG(B(IP))
        ENDIF
  200 CONTINUE
  205 CONTINUE
C---------PRINT HEADER
      WRITE(IOUT,555)(IP,IP=1,NP)
      WRITE(IOUT,557)(PID(IP),IP=1,NP)
      WRITE(IOUT,558)
C---------PRINT ESTIMATED PARAMETERS
      WRITE(IOUT,565)(B(IP),IP=1,NP)
C---------PRINT EXPONENTIALS OF LOG-TRANSFORMED PARAMETERS
      IF(LNFLAG.NE.0) THEN
        WRITE(IOUT,559)
        DO 337 IP=1,NP
        BUFF(IP,1)=0.
```

```
  337   IF(LN(IP).GT.0) BUFF(IP,1)=EXP(B(IP))
        WRITE(IOUT,565)(BUFF(IP,1),IP=1,NP)
      ENDIF
C---------PUT STANDARD DEVIATIONS IN SINGLE-PRECISION ARRAY AND PRINT
      DO 340 IP=1,NP
  340 BUFF(IP,1)=SNGL(SCLE(IP))
      WRITE(IOUT,560)
      WRITE(IOUT,565)(BUFF(IP,1),IP=1,NP)
C---------COMPUTE COEFFICIENTS OF VARIATION AND PRINT
      DO 345 IP=1,NP
  345 BUFF(IP,1)=BUFF(IP,1)/ABS(B(IP))
      WRITE(IOUT,562)
      WRITE(IOUT,565)(BUFF(IP,1),IP=1,NP)
C---------COMPUTE +- 2 STANDARD DEVIATIONS AND PRINT
      DO 346 IP=1,NP
  346 BUFF(IP,1)=B(IP)+2.*SNGL(SCLE(IP))
      WRITE(IOUT,600)
      WRITE(IOUT,565) (BUFF(IP,1),IP=1,NP)
      IF(LNFLAG.NE.0) THEN
        WRITE(IOUT,559)
        DO 347 IP=1,NP
          IF(LN(IP).EQ.0) THEN
            BUFF(IP,1)=0.
          ELSE
            BT=BUFF(IP,1)
            IF(BT.LT.53.0) BUFF(IP,1)=EXP(BT)
            IF(BUFF(IP,1).LT.0.) BUFF(IP,1)=1.E-23
            IF(BT.GE.53.0) BUFF(IP,1)=1.E+23
          ENDIF
  347   CONTINUE
        WRITE(IOUT,565)(BUFF(IP,1),IP=1,NP)
      ENDIF
      DO 348 IP=1,NP
  348 BUFF(IP,1)=B(IP)-2.*SNGL(SCLE(IP))
      WRITE(IOUT,610)
      WRITE(IOUT,565) (BUFF(IP,1),IP=1,NP)
      IF(LNFLAG.NE.0) THEN
        WRITE(IOUT,559)
        DO 349 IP=1,NP
        IF(LN(IP).EQ.0) BUFF(IP,1)=0.
  349   IF(LN(IP).GT.0) BUFF(IP,1)=EXP(BUFF(IP,1))
        WRITE(IOUT,565)(BUFF(IP,1),IP=1,NP)
      ENDIF
      WRITE(IOUT,590)
C--------COMPUTE AND PRINT CORRELATION MATRIX FOR PARAMETERS
      IF(NP.GT.1) THEN
      DO 360 IP=1,NP
      DTMPA=SCLE(IP)
      DO 350 I=IP,NP
      C(I,IP)=C(I,IP)/(SCLE(I)*DTMPA)
  350 C(IP,I)=C(I,IP)
  360 CONTINUE
      DO 370 IP=1,NP
        DO 370 I=1,NP
  370   BUFF(I,IP)=C(I,IP)
      WRITE(IOUT,570) (ANAME(I,2),I=1,4)
      CALL ULAPRW(BUFF,ANAME(1,2),IDUM,IDUM,NP,NP,-IDUM,IPRC,IOUT)
C--------CHECK FOR HIGHLY CORRELATED PARAMETER PAIRS
      WRITE(IOUT,535)
      DO 380 IP=1,NP
        I1=IP+1
        DO 380 I=I1,NP
  380     IF(DABS(C(IP,I)).GE..95) WRITE(IOUT,540) IP,PID(IP),I,PID(I),
     1      SNGL(C(IP,I))
      WRITE(IOUT,545)
      DO 390 IP=1,NP
        I1=IP+1
        DO 390 I=I1,NP
          TMP=DABS(C(IP,I))
  390     IF(TMP.GE..90.AND.TMP.LT..95)
     1         WRITE(IOUT,540) IP,PID(IP),I,PID(I),SNGL(C(IP,I))
      WRITE(IOUT,550)
      DO 400 IP=1,NP
        I1=IP+1
        DO 400 I=I1,NP
          TMP=DABS(C(IP,I))
  400     IF(TMP.GE..85.AND.TMP.LT..90)
     1         WRITE(IOUT,540) IP,PID(IP),I,PID(I),SNGL(C(IP,I))
      ENDIF
C-----CALCULATE CORRELATION COEFFICIENT
      CALL SPAR1A(ND,WT,HOBS,H,R,R1,NPR,MPR,B1,NP,WP,B,PRM,NPO)
C-----PRINT FINAL RSQ'S, ERROR VARIANCE, CORRELATION, ITERATIONS
      WRITE(IOUT,521) RSQ,RSQP,VAR,R,R1,ITERPF
```

```
C-----CALC AND PRINT STATISTICS BASED ON MAX LIKELIHOOD OBJ FUNCTION
      CALL SPAR1M(RSQP,WT,NP,ND,WP,MPR,IOUT,EV,NPR)
      RETURN
      END


C=======================================================================
      SUBROUTINE PAR1RE(
     1                   NP,B,B1,IWPG,WP,IOUT,NH,H,HOBS,WT,DID,
     2                   NQT,ND,IPAR,NPR,MPR,PRM,NPO
     1                   )
C-----VERSION 1000 01FEB1992
C     ******************************************************************
C     ANALYSIS OF RESIDUALS
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION B(NPO),IWPG(NP),B1(NPO),WP(NPO+MPR),H(ND+NPR+MPR),
     1          HOBS(ND),WT(ND),PRM(NPO+1,NP)
C     ------------------------------------------------------------------
C
       AVET=0.
       NND=0
C-------------HEADS
      IF(NH.GT.0) THEN
        DO 160 N=1,NH
          RES=H(N)-HOBS(N)
          W=WT(N)
          IF(W.LT.0.) GO TO 160
          WT2=SQRT(W)
          WTR=RES*WT2
          NND=NND+1
          H(NND)=WTR
          AVET=AVET+WTR
  160   CONTINUE
      ENDIF
C-----HEAD-DEPENDENT FLOWS
      IF(NQT.GT.0) THEN
        NH1=NH+1
        NH2=NH+NQT
        DO 170 N=NH1,NH2
          RES=H(N)-HOBS(N)
          W=WT(N)
          IF(W.LT.0.) GO TO 170
          WT2=SQRT(W)
          WTR=RES*WT2
          NND=NND+1
          H(NND)=WTR
          AVET=AVET+WTR
  170   CONTINUE
      ENDIF
C------TEST FOR NORMALITY OF THE DEPENDENT-VARIABLE WEIGHTED RESIDUALS
C-------CALCULATE THE MEAN
      AVE=AVET/REAL(NND)
C-------ORDER THE RESIDUALS
      DO 240 NN=1,NND-1
        NMIN=NN
        RMIN=H(NN)
        DO 200 N=NN+1,NND
          IF(H(N).LE.RMIN) THEN
            RMIN=H(N)
            NMIN=N
          ENDIF
  200   CONTINUE
        IF(NMIN.NE.NN) THEN
          H(NMIN)=H(NN)
          H(NN)=RMIN
        ENDIF
  240 CONTINUE
      WRITE(IOUT,515) NND
  515 FORMAT(/,' ORDERED DEPENDENT-VARIABLE WEIGHTED RESIDUALS',
     1/,' NUMBER OF RESIDUALS INCLUDED:',I10)
      WRITE(IOUT,530) (H(N),N=1,NND)
C------CALCULATE THE STATISTIC
      RNUM=0.
      DEN1=0.
      DEN2=0.
      DO 250 N=1,NND
        RNORM=(REAL(N)-.5)/REAL(NND)
        CALL UNORM(U,RNORM,-1)
        DIF=H(N)-AVE
        RNUM=RNUM+DIF*U
        DEN1=DEN1+DIF**2
```

```
        DEN2=DEN2+U**2
  250 CONTINUE
      STAT=RNUM**2/(DEN1*DEN2)
      WRITE(IOUT,520) STAT
      IF((NPR+MPR).EQ.0) RETURN
C------PRIOR INFORMATION ON INDIVIDUAL PARAMETERS BY PARAMETER GROUP
      IF(IPAR.GT.0.AND.NPR.GT.0) THEN
        DO 327 IG=1,NP
          DO 326 IP=1,NP
            IF(IWPG(IP).EQ.IG.AND.WP(IP).GT.0.) THEN
              WPSR=WP(IP)**.5
              BDIF=B1(IP)-B(IP)
              BWP=BDIF*WPSR
              NND=NND+1
              H(NND)=BWP
              AVET=AVET+BWP
            ENDIF
  326     CONTINUE
  327   CONTINUE
      ENDIF
C-----PRIOR INFORMATION ON PARAMETER SUMS
      IF(IPAR.GT.0.AND.MPR.GT.0) THEN
        DO 350 IMP=1,MPR
          TEMP=0.
          DO 340 IIP=1,NPO
  340     TEMP=TEMP+PRM(IIP,IMP)*B(IIP)
          WPSR=WP(NPO+IMP)**.5
          TEMP1=PRM(NPO+1,IMP)
          BDIF=TEMP1-TEMP
          BWP=BDIF*WPSR
          NND=NND+1
          H(NND)=BWP
          AVET=AVET+BWP
  350   CONTINUE
      ENDIF
C-------CALCULATE THE MEAN
      AVET=AVET/REAL(NND)
C-------ORDER THE RESIDUALS
      DO 390 NN=1,NND-1
        RMIN=H(NN)
        NMIN=NN
        DO 370 N=NN,NND
          IF(H(N).LE.RMIN) THEN
            RMIN=H(N)
            NMIN=N
          ENDIF
  370   CONTINUE
        IF(NMIN.NE.NN) THEN
          H(NMIN)=H(NN)
          H(NN)=RMIN
        ENDIF
  390 CONTINUE
      WRITE(IOUT,510) NND
  510 FORMAT(/,' ORDERED WEIGHTED RESIDUALS',
     1/,' NUMBER OF RESIDUALS INCLUDED:',I10)
      WRITE(IOUT,530) (H(N),N=1,NND)
  530 FORMAT (8G10.3)
C------CALCULATE THE STATISTIC
      RNUM=0.
      DEN1=0.
      DEN2=0.
      DO 420 N=1,NND
        RNORM=(REAL(N)-.5)/REAL(NND)
        CALL UNORM(U,RNORM,-1)
        DIF=H(N)-AVET
        RNUM=RNUM+DIF*U
        DEN1=DEN1+DIF**2
        DEN2=DEN2+U**2
  420 CONTINUE
      STAT=RNUM**2/(DEN1*DEN2)
      WRITE(IOUT,520) STAT
  520 FORMAT(' CORRELATION BETWEEN ORDERED WEIGHTED RESIDUALS AND',/,
     1 ' INDEPENDENT NORMAL DEVIATES =',G13.3)
      RETURN
      END
```

New PAR Submodules

Listed in alphabetical order

```
C=========================================================================
      SUBROUTINE SPAR1A(
     1                 ND,WT,HOBS,H,R,R1,NPR,MPR,B1,NP,WP,B,PRM,NPO
     1                )
C-----VERSION 1000 01FEB1992
C
C     ********************************************************************
C     CALCULATE CORRELATION COEFFICIENT
C     ********************************************************************
C        SPECIFICATIONS:
C     --------------------------------------------------------------------
      DOUBLE PRECISION SUMA,SUMB,SUMC,SUMD,SUM,TMP,TEMP,TMPP
      DIMENSION WT(ND),HOBS(ND),H(ND),B1(NPO),B(NPO),PRM(NPO+1,MPR),
     1          WP(NPO+MPR)
C     --------------------------------------------------------------------
C
      OBS=ND
      SUMA=0.
      SUMB=0.
      SUMC=0.
      SUMD=0.
      SUM=0.
      DO 460 N=1,ND
        IF(WT(N).LT.0.) THEN
          OBS=OBS-1.
          GO TO 460
        ENDIF
        TMP=WT(N)**.5
        TEMP=TMP*HOBS(N)
        TMP=TMP*H(N)
        SUMA=SUMA+TEMP
        SUMB=SUMB+TMP
        SUMC=SUMC+TEMP*TEMP
        SUMD=SUMD+TMP*TMP
        SUM=SUM+TEMP*TMP
  460 CONTINUE
      R=(OBS*SUM-SUMA*SUMB)/((OBS*SUMC-SUMA*SUMA)*(OBS*SUMD-SUMB*SUMB))
     1**.5
      R1=R
C------PRIOR PARAMETER ESTIMATES
      IF(NPR+MPR.GT.0) THEN
        OBS=OBS+NPR+MPR
        DO 540 I=1,NPO+MPR
          IF(WP(I).GT.0..AND.(I.LE.NP.OR.I.GT.NPO)) THEN
            TMP=WP(I)**.5
            IF(I.LE.NP) THEN
              TEMP=TMP*B1(I)
              TMP=TMP*B(I)
            ELSE
              TEMP=TMP*PRM(NPO+1,I-NP)
              TMPP=0.
              DO 520 II=1,NPO
  520         TMPP=TMPP+PRM(II,I-NPO)*B(II)
              TMP=TMPP*TMP
            ENDIF
            SUMA=SUMA+TEMP
            SUMB=SUMB+TMP
            SUMC=SUMC+TEMP*TEMP
            SUMD=SUMD+TMP*TMP
            SUM=SUM+TEMP*TMP
          ENDIF
  540   CONTINUE
        R1=(OBS*SUM-SUMA*SUMB)/
     1    ((OBS*SUMC-SUMA*SUMA)*(OBS*SUMD-SUMB*SUMB))**.5
      ENDIF
      RETURN
      END


C=========================================================================
      SUBROUTINE SPAR1E(
     1                 C,BUFF1,BUFF2,NP,BUFF3,IOUT,B,B1,ITERPF,BUFF,
     1                 PID,IPRC,IE
     1                )
C-----VERSION 1000 01FEB1992
C
C     ********************************************************************
C     CALCULATE EIGENVALUES AND EIGENVECTORS OF COVARIANCE MATRIX ON THE
C     PARAMETERS.
```

```
C     *****************************************************************
C        SPECIFICATIONS:
C     -----------------------------------------------------------------
      CHARACTER*4 PID(NP),ANAME
      DOUBLE PRECISION C(NP,NP),BUFF3(NP),BUFF1(NP,NP),BUFF2(NP)
      DIMENSION B(NP),B1(NP),ANAME(4),BUFF(NP,NP)
      DATA ANAME(1),ANAME(2),ANAME(3),ANAME(4)
     # /'SCAL','ED W',' PAR','AM. '/
C     -----------------------------------------------------------------
  500 FORMAT(/,A4,8D13.4,/,10(4X,8D13.4))
  510 FORMAT(/,4X,8D13.4,/,10(4X,8D13.4))
  520 FORMAT(/,' EIGENVALUES')
  530 FORMAT(/,' EIGENVECTORS')
  540 FORMAT(/,' EIGENVALUE AND EIGENVECTOR CALCULATION PERFORMANCE',
     1' INDEX = ',E13.4,' (<1 EXCELLENT, 1 TO 100 GOOD, >100 POOR)')
  550 FORMAT(/,' NUMBER OF JACOBI ROTATIONS:',I7)
  570 FORMAT(/,10X,4A4/10X,16('-'))
C
C---------SCALE VARIANCE-COVARIANCE MATRIX WITH PARAMETER VALUES
      DO 20 IP=1,NP
        DO 10 IIP=1,NP
   10   BUFF1(IP,IIP)=0.
        BUFF2(IP)=0.
   20   BUFF3(IP)=0.
      IF(ITERPF.GT.1) THEN
        DO 30 IP=1,NP
        DO 30 IIP=1,NP
   30   C(IP,IIP)=C(IP,IIP)/(B(IP)*B(IIP))
      ELSE
        DO 40 IP=1,NP
        DO 40 IIP=1,NP
   40   C(IP,IIP)=C(IP,IIP)/(B1(IP)*B1(IIP))
      ENDIF
      DUM=0.
      CALL USAVEF(DUM,BUFF,C,DUM,NP*NP,3)
      DO 50 IP=1,NP
        DO 50 I=1,NP
   50   BUFF(I,IP)=C(I,IP)
      IDUM=1
      WRITE(IOUT,570) (ANAME(I),I=1,4)
      CALL ULAPRW(BUFF,ANAME(1),IDUM,IDUM,NP,NP,-IDUM,IPRC,IOUT)
C-------CALCULATE EIGENVALUES AND EIGENVECTORS
      IF(IE.EQ.1) THEN
C        CALL DE2CSF(NP,C,NP,BUFF3,BUFF1,NP,BUFF2)
C-------CALCULATE PERFORMANCE INDEX
C        PI = DEPISF(NP,NP,C,NP,BUFF3,BUFF1,NP)
      ELSEIF(IE.GT.1) THEN
        DO 60 IP=1,NP
        DO 60 I=1,NP
   60   BUFF1(I,IP)=C(I,IP)
        CALL TRED2(BUFF1,NP,NP,BUFF3,BUFF2)
        CALL TQLI(BUFF3,BUFF2,NP,NP,BUFF1)
        CALL EIGSRT(BUFF3,BUFF1,NP,NP)
      ENDIF
C-------SCALE EIGENVECTORS TO FORM UNIT VECTORS
      DO 80 I=1,NP
        TEMP=0.
        DO 70 J=1,NP
   70   TEMP=TEMP+BUFF1(J,I)**2
        TEMP=TEMP**.5
        DO 75 J=1,NP
   75   BUFF1(J,I)=BUFF1(J,I)/TEMP
   80 CONTINUE
C-------PRINT EIGENVALUES
      WRITE(IOUT,520)
      WRITE(IOUT,510) (BUFF3(J),J=1,NP)
C-------PRINT EIGENVECTORS
      WRITE(IOUT,530)
      DO 90 J=1,NP
   90 WRITE(IOUT,500) PID(J),(BUFF1(J,I),I=1,NP)
      IF(IE.EQ.1) WRITE(IOUT,540) PI
C-------EIGENVECTOR TEST (FROM PRESS AND OTHERS, 1989)
      IF(IE.EQ.3) THEN
        WRITE(IOUT,'(/1X,A)') 'EIGENVECTOR TEST'
        DO 130 J=1,NP
          DO 110 L=1,NP
            BUFF2(L)=0.0
            DO 100 K=1,NP
              IF (K.GT.L) THEN
                KK=L
                LL=K
              ELSE
                KK=K
```

```
                LL=L
              ENDIF
              BUFF2(L)=BUFF2(L)+C(LL,KK)*BUFF1(K,J)
  100         CONTINUE
  110       CONTINUE
            WRITE(IOUT,'(/1X,A,I3)') 'VECTOR NUMBER',J
                    WRITE(IOUT,'(/1X,T7,A,T18,A,T31,A)')
     *                              'VECTOR','MTRX*VEC ','RATIO'
            DO 120 L=1,NP
              RATIO=BUFF2(L)/BUFF1(L,J)
              WRITE(IOUT,'(1X,3G12.5)') BUFF1(L,J),BUFF2(L),RATIO
  120       CONTINUE
  130     CONTINUE
        ENDIF
C---------UNSCALE VARIANCE-COVARIANCE MATRIX
        IF(ITERPF.GT.1) THEN
          DO 140 IP=1,NP
          DO 140 IIP=1,NP
  140     C(IP,IIP)=C(IP,IIP)*(B(IP)*B(IIP))
        ELSE
          DO 150 IP=1,NP
          DO 150 IIP=1,NP
  150     C(IP,IIP)=C(IP,IIP)*(B1(IP)*B1(IIP))
        ENDIF
        RETURN
        END


C=======================================================================
      SUBROUTINE SPAR1I(NP,C)
C-----VERSION 1000 01FEB1992
C
C     ******************************************************************
C         COMPLETE CALCULATION OF THE INVERSE OF SCALED
C         COEFFICIENT MATRIX CSTARTING FROM DECOMPOSED MATRIX
C         FROM SPAR1R, FOR NP>1
C     ******************************************************************
C         SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION C(NP,NP),DSUM
C     ------------------------------------------------------------------
      NP1=NP-1
      C(NP,NP)=1.D00/C(NP,NP)
      DO 260 K=1,NP1
        KP1=K+1
        DO 230 I=KP1,NP
          DSUM=0.
          IM1=I-1
          DO 220 J=K,IM1
  220       DSUM=DSUM+C(I,J)*C(J,K)
          C(K,I)=-DSUM
  230     C(I,K)=-DSUM*C(I,I)
        DO 250 I=1,K
          DSUM=C(K,I)
          DO 240 J=KP1,NP
  240       DSUM=DSUM+C(J,K)*C(I,J)
          C(K,I)=DSUM
  250     C(I,K)=C(K,I)
  260 CONTINUE
      RETURN
      END


C=======================================================================
      SUBROUTINE SPAR1M(
     1                  RSQP,WT,NP,ND,WP,MPR,IOUT,EV,NPR
     1                  )
C-----VERSION 1000 01FEB1992
C
C     ******************************************************************
C     CALCULATE STATISTICS BASED ON THE MAXIMUM LIKELIHOOD OBJECTIVE
C     FUNCTION
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DIMENSION WT(ND),WP(NP+MPR)
C     ------------------------------------------------------------------
  520 FORMAT(/,'  MAX LIKE OBJ FUNC = ',G11.5,/,
     1        '  AIC STATISTIC---- = ',G11.5,/,
     2        '  BIC STATISTIC---- = ',G11.5)
C
      AEV=ALOG(EV)
      RND=REAL(ND)
```

```
      RP=REAL(NPR+MPR)
C-----DEPENDENT-VARIABLE PART OF WEIGHT MATRIX
      PROD=0.
      DO 100 N=1,ND
  100 PROD=PROD+ALOG(WT(N))
      PROD=PROD-RND*AEV
C-----PRIOR PART OF WEIGHT MATRIX
      PRODP=0.
      DO 120 IP=1,NP+MPR
  120 IF(WP(IP).NE.0.) PRODP=PRODP+ALOG(WP(IP))
      PRODP=PRODP-RP*AEV
C-----LOG-LIKELIHOOD FUNCTION
      OF=(RSQP/EV)-PROD-PRODP+(RND+RP)*ALOG(2.*3.14159265)
C-----AIC AND BIC
      AIC=OF+2.*REAL(NP)
      BIC=OF+NP*ALOG(RND+RP)
      WRITE(IOUT,520) OF,AIC,BIC
      RETURN
      END


C==============================================================
      SUBROUTINE SPAR1P(
     1                 PID,B,NP,IOUB,IOUT,ISN
     1                 )
C-----VERSION 1000 01FEB1992
C
C     ******************************************************************
C         READ AND PRINT PARAMETER VALUES AND STATISTICS FOR ALL
C           ITERATIONS
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      CHARACTER*4 PID(NP)
      DIMENSION B(NP)
C     ------------------------------------------------------------------
  510 FORMAT(8F10.0)
  520 FORMAT(6X,6E10.3,/,50(6X,6E10.3,/))
  530 FORMAT(/,' PARAMETER VALUES AND STATISTICS FOR ALL ITERATIONS',
     1       /,10X,' PAR. ID.:',/,
     1            9X,A4,6X,A4,6X,A4,6X,A4,6X,A4,6X,A4,/,
     1         50(9X,A4,6X,A4,6X,A4,6X,A4,6X,A4,6X,A4,/))
  550 FORMAT(70X,'SUM OF SQUARED WEIGHTED RESIDUALS',22X,'AMP OR',/,
     1         ' ITER',66X,'HEADS        W/FLOWS      W/PARAMS',
     2                                 DMX         PAR#    AGMX')
  580 FORMAT(F10.0,I5)
  590 FORMAT(I4,65X,G10.3,2X,G10.3,2X,G10.3,2X,G10.3,I5,2X,G10.3)
C
      REWIND(IOUB)
      WRITE(IOUT,530) (PID(IP),IP=1,NP)
      WRITE(IOUT,550)
      ITER=0
  110 ITER=ITER+1
      READ(IOUB,510,END=140) (B(IP),IP=1,NP)
      WRITE(IOUT,520) (B(IP),IP=1,NP)
      READ(IOUB,510,END=140) RSQH,RSQ,RSQP
C------FOR ISN>0, AMP IS AGMX
      READ(IOUB,580,END=120) DMX,NPAR,AMP
      WRITE(IOUT,590) ITER,RSQH,RSQ,RSQP,DMX,NPAR,AMP
      GO TO 110
  120 WRITE(IOUT,590) ITER,RSQH,RSQ,RSQP
  140 CONTINUE
      RETURN
      END


C=======================================================================
      SUBROUTINE SPAR1Q(C,DD,G,NP,R,GD,U,ITERP,IP,X,ND,HOBS,H,WT,S,NFIT,
     1          B1,B,WP,NPR,MPR,PRM,NPO,XD,NPER,SOSR)
C-----VERSION 1000 01FEB1992
C
C     ******************************************************************
C     COMPUTE QUASI-NEWTON COMPONENT OF C MATRIX AND ADD TO C
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      DOUBLE PRECISION C(NP,NP),DD(NP),SUM,TMPA,DS,DGD,DU,CF,QD,TMP,T,
     1       R(NP*NP/2+NP),DPGD,DPU,DTMPA,U(NP),S(NP),G,GD,DPGI,W1
      DIMENSION G(NP),GD(NP),X(NP,ND),HOBS(ND),H(ND),WT(ND),
     1     WP(NPO+MPR),PRM(NPO+1,MPR),B1(IP),B(IP),XD(NP,ND)
C     ------------------------------------------------------------------
C------FOR FIRST PARAMETER-ESTIMATION ITERATIONS, INITIALIZE UPDATING
C------MATRIX R AND SAVE G AND X
```

```
      IF(ITERP.EQ.1) THEN
        I=0
        DO 50 IIP=1,NP
          DO 55 IPP=IIP,NP
            I=I+1
            R(I)=0.
   55     CONTINUE
   50   GD(IIP)=G(IIP)
        DO 53 N=1,ND
        DO 53 IIP=1,NP
   53   XD(IIP,N)=X(IIP,N)
        RETURN
      ENDIF
C------CALCULATE MATRIX R
      DO 30 IIP=1,NP
   30 S(IIP)=0.
      DO 40 N=1,ND
        IF(WT(N).LT.0.) GO TO 40
        W1=(HOBS(N)-H(N))*WT(N)
        DO 35 IIP=1,NP
   35   S(IIP)=S(IIP)+W1*(XD(IIP,N)-X(IIP,N))
   40 CONTINUE
C
      DO 80 J=1,NP
   80 GD(J)=GD(J)-G(J)
C
      K=0
      QD=0.
      DS=0.
      DO 340 J=1,NP
      M=NP-J+1
      K=K+1
      L=K
      SUM=0.
      IF(J.EQ.NP) GO TO 325
      JP1=J+1
      DO 320 I=JP1,NP
      K=K+1
  320 SUM=SUM+DD(I)*R(K)
  325 TMPA=DD(J)*R(L)
      QD=QD+(SUM+SUM+TMPA)*DD(J)
      SUM=SUM+TMPA
      IF(J.EQ.1) GO TO 335
      JM1=J-1
      DO 330 I=1,JM1
      L=L-M
      SUM=SUM+R(L)*DD(J-I)
  330 M=M+1
  335 U(J)=SUM
  340 DS=DS+DD(J)*S(J)
C
      TMP=1.
      IF(QD.NE.0.) TMP=DABS(DS/QD)
      T=1.
      IF(TMP.LT.1.) T=TMP
      DGD=0.
      DU=0.
      DO 345 J=1,NP
      DGD=DGD+DD(J)*GD(J)
      U(J)=S(J)-T*U(J)
  345 DU=DU+DD(J)*U(J)
C
      CF=DU/(DGD*DGD)
      K=0
      DO 355 J=1,NP
      DO 350 I=J,NP
      K=K+1
      DPGD=GD(J)
      DPU=U(J)
      DPGI=GD(I)
  350 R(K)=T*R(K)+(U(I)*DPGD+DPGI*DPU)/DGD-CF*DPGI*DPGD
  355 CONTINUE
C------SAVE GRADIENT
      DO 360 IIP=1,NP
  360 GD(IIP)=G(IIP)
C-----SAVE SENSITIVITIES
      DO 365 N=1,ND
      DO 365 IIP=1,NP
  365 XD(IIP,N)=X(IIP,N)
C------ADD QN APPROXIMATION TO COEFFICIENT MATRIX
      IF((NFIT.GT.0.AND.ITERP.GT.NFIT).OR.SOSR.LT.0.)THEN
        I=0
        DO 370 IIP=1,NP
```

```
        DO 370 IPP=IIP,NP
          I=I+1
          C(IPP,IIP)=C(IPP,IIP)+R(I)
  370   CONTINUE
      ENDIF
      RETURN
      END

C=====================================================================
      SUBROUTINE SPAR1R(C,DD,G,NP,CSA,IND,IFO,AMP,DET)
C-----VERSION 1000 01FEB1992
C
C     *****************************************************************
C          COMPUTE PARAMETER STEP LENGTHS USING THE MARQUARDT PROCEDURE
C     *****************************************************************
C         SPECIFICATIONS:
C     -----------------------------------------------------------------
      DOUBLE PRECISION C(NP,NP),DD(NP),DPIV,DTMPA,DSUM,G
      DIMENSION G(NP)
C     -----------------------------------------------------------------
C
C          COMPUTE TRIAL PARAMETER STEP LENGTHS USING LDU FACTORIZATION:
C            DECOMPOSE MATRIX
      NM1=NP-1
   20 IND=0
      DET=1.
      DO 60 K=1,NM1
        DPIV=C(K,K)
        DET=DET*DPIV
        IF(DPIV.GT.1.D-13) GO TO 30
        IND=1
        GO TO 115
   30   DPIV=1.D00/DPIV
        KP1=K+1
        DO 50 J=KP1,NP
          DTMPA=C(J,K)*DPIV
          DO 40 I=J,NP
   40       C(I,J)=C(I,J)-DTMPA*C(I,K)
   50   CONTINUE
        C(K,K)=DPIV
   60 CONTINUE
      DET=DET*C(NP,NP)
      IF(C(NP,NP).GT.1.D-13) GO TO 65
      IND=1
      IF(IFO.EQ.0) GO TO 115
   65 IF(IFO.GT.0) RETURN
      DO 67 K=1,NP
   67   DD(K)=G(K)
C            FORWARD SUBSTITUTE
      DO 80 K=1,NM1
        DTMPA=DD(K)*C(K,K)
        KP1=K+1
        DO 70 J=KP1,NP
   70     DD(J)=DD(J)-C(J,K)*DTMPA
   80 CONTINUE
C            BACK SUBSTITUTE
      DD(NP)=DD(NP)/C(NP,NP)
      I=NP
   85 I=I-1
      IF(I.LT.1) GO TO 100
      IP1=I+1
      DSUM=0.
      DO 90 J=IP1,NP
   90   DSUM=DSUM+C(J,I)*DD(J)
      DD(I)=(DD(I)-DSUM)*C(I,I)
      GO TO 85
C          CHECK SOLUTION AND ADD MARQUARDT PARAMETER IF NEEDED
  100 SUM=0.
      SUMA=0.
      SUMB=0.
      DO 110 I=1,NP
        SUM=SUM+DD(I)*G(I)
        SUMA=SUMA+DD(I)*DD(I)
  110   SUMB=SUMB+G(I)*G(I)
      IF(SUM.GT.CSA*SQRT(SUMA*SUMB)) RETURN
  115 AMP=1.5*AMP+.001
      IF(AMP.GT.1.) RETURN
      DO 130 I=1,NP
        C(I,I)=1.D00+AMP
        DO 120 J=I,NP
  120     C(J,I)=C(I,J)
  130 CONTINUE
```

```
      GO TO 20
      END
```

New Utility Modules

```
C=======================================================================
      SUBROUTINE TRED2(A,N,NP,D,E)
C-----VERSION 1000 01FEB1992
C
C     ******************************************************************
C     FROM PRESS AND OTHERS (1989); MODIFIED FOR DOUBLE PRECISION
C     ******************************************************************
C        SPECIFICATIONS:
C     ------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      DIMENSION A(NP,NP),D(NP),E(NP)
C     ------------------------------------------------------------------
      IF(N.GT.1)THEN
        DO 18 I=N,2,-1
          L=I-1
          H=0.
          SCALE=0.
          IF(L.GT.1)THEN
            DO 11 K=1,L
              SCALE=SCALE+DABS(A(I,K))
11          CONTINUE
            IF(SCALE.EQ.0.)THEN
              E(I)=A(I,L)
            ELSE
              DO 12 K=1,L
                A(I,K)=A(I,K)/SCALE
                H=H+A(I,K)**2
12            CONTINUE
              F=A(I,L)
              G=-SIGN(DSQRT(H),F)
              E(I)=SCALE*G
              H=H-F*G
              A(I,L)=F-G
              F=0.
              DO 15 J=1,L
                A(J,I)=A(I,J)/H
                G=0.
                DO 13 K=1,J
                  G=G+A(J,K)*A(I,K)
13              CONTINUE
                IF(L.GT.J)THEN
                  DO 14 K=J+1,L
                    G=G+A(K,J)*A(I,K)
14                CONTINUE
                ENDIF
                E(J)=G/H
                F=F+E(J)*A(I,J)
15            CONTINUE
              HH=F/(H+H)
              DO 17 J=1,L
                F=A(I,J)
                G=E(J)-HH*F
                E(J)=G
                DO 16 K=1,J
                  A(J,K)=A(J,K)-F*E(K)-G*A(I,K)
16              CONTINUE
17            CONTINUE
            ENDIF
          ELSE
            E(I)=A(I,L)
          ENDIF
          D(I)=H
18      CONTINUE
      ENDIF
      D(1)=0.
      E(1)=0.
      DO 23 I=1,N
        L=I-1
        IF(D(I).NE.0.)THEN
          DO 21 J=1,L
            G=0.
            DO 19 K=1,L
              G=G+A(I,K)*A(K,J)
19          CONTINUE
            DO 20 K=1,L
              A(K,J)=A(K,J)-G*A(K,I)
20          CONTINUE
```

```
21          CONTINUE
          ENDIF
          D(I)=A(I,I)
          A(I,I)=1.
          IF(L.GE.1)THEN
            DO 22 J=1,L
              A(I,J)=0.
              A(J,I)=0.
22          CONTINUE
          ENDIF
23      CONTINUE
        RETURN
        END


C=====================================================================
        SUBROUTINE TQLI(D,E,N,NP,Z)
C-----VERSION 1000 01FEB1992
C
C       ****************************************************************
C       FROM PRESS AND OTHERS (1989); MODIFIED FOR DOUBLE PRECISION
C       ****************************************************************
C          SPECIFICATIONS:
C       ----------------------------------------------------------------
        IMPLICIT DOUBLE PRECISION (A-H,P-Z)
        DIMENSION D(NP),E(NP),Z(NP,NP)
C       ----------------------------------------------------------------
        IF (N.GT.1) THEN
          DO 11 I=2,N
            E(I-1)=E(I)
11        CONTINUE
          E(N)=0.
          DO 15 L=1,N
            ITER=0
1           DO 12 M=L,N-1
              DD=DABS(D(M))+DABS(D(M+1))
              IF (DABS(E(M))+DD.EQ.DD) GO TO 2
12          CONTINUE
            M=N
2           IF(M.NE.L)THEN
              IF(ITER.EQ.30)PAUSE 'too many iterations'
              ITER=ITER+1
              G=(D(L+1)-D(L))/(2.*E(L))
              R=DSQRT(G**2+1.)
              G=D(M)-D(L)+E(L)/(G+SIGN(R,G))
              S=1.
              C=1.
              P=0.
              DO 14 I=M-1,L,-1
                F=S*E(I)
                B=C*E(I)
                IF(DABS(F).GE.DABS(G))THEN
                  C=G/F
                  R=DSQRT(C**2+1.)
                  E(I+1)=F*R
                  S=1./R
                  C=C*S
                ELSE
                  S=F/G
                  R=DSQRT(S**2+1.)
                  E(I+1)=G*R
                  C=1./R
                  S=S*C
                ENDIF
                G=D(I+1)-P
                R=(D(I)-G)*S+2.*C*B
                P=S*R
                D(I+1)=G+P
                G=C*R-B
                DO 13 K=1,N
                  F=Z(K,I+1)
                  Z(K,I+1)=S*Z(K,I)+C*F
                  Z(K,I)=C*Z(K,I)-S*F
13              CONTINUE
14            CONTINUE
              D(L)=D(L)-P
              E(L)=G
              E(M)=0.
              GO TO 1
            ENDIF
15        CONTINUE
        ENDIF
        RETURN
```

```
      END

C=========================================================================
      SUBROUTINE EIGSRT(D,V,N,NP)
C-----VERSION 1000 01FEB1992
C
C     *******************************************************************
C     FROM PRESS AND OTHERS (1989); MODIFIED FOR DOUBLE PRECISION AND TO
C     ORDER THE EIGENVALUES FROM SMALLEST TO LARGEST
C     *******************************************************************
C        SPECIFICATIONS:
C     -------------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,P-Z)
      DIMENSION D(NP),V(NP,NP)
C     -------------------------------------------------------------------
      DO 13 I=1,N-1
        K=I
        P=D(I)
        DO 11 J=I+1,N
          IF(D(J).LE.P)THEN
            K=J
            P=D(J)
          ENDIF
11      CONTINUE
        IF(K.NE.I)THEN
          D(K)=D(I)
          D(I)=P
          DO 12 J=1,N
            P=V(J,I)
            V(J,I)=V(J,K)
            V(J,K)=P
12        CONTINUE
        ENDIF
13    CONTINUE
      RETURN
      END

C=========================================================================
      SUBROUTINE U2DINN (IA,II,JJ,IN)
C-----VERSION 1000 01FEB1992 U2DINN
C     *******************************************************************
C     READ INTEGER MATRICES WITHOUT PRINTING ANYTHING
C     *******************************************************************
C     SPECIFICATIONS:
C     -------------------------------------------------------------------
      CHARACTER*20 FMTIN
      DIMENSION IA(JJ,II)
C     -------------------------------------------------------------------
C
      READ(IN,1) LOCAT,ICONST,FMTIN,IPRN
    1 FORMAT(I10,I10,A20,I10)
C
      IF(LOCAT) 200,50,90
C        LOCAT=0
   50 DO 80 I=1,II
      DO 80 J=1,JJ
   80 IA(I,J)=ICONST
      RETURN
C        LOCAT>0
   90 DO 100 I=1,II
  100 READ(LOCAT,FMTIN)(IA(J,I),J=1,JJ)
      GO TO 300
C        LOCAT<0
  200 LOCAT=-LOCAT
      READ(LOCAT)
      READ(LOCAT) IA
C        MULTIPLY BY A CONSTANT
  300 IF(ICONST.EQ.0.) RETURN
      DO 310 I=1,II
      DO 310 J=1,JJ
  310 IA(J,I)=IA(J,I)*ICONST
      RETURN
      END

C=========================================================================
      SUBROUTINE U2DREN (A,II,JJ,IN)
C-----VERSION 1000 01FEB1992 U2DREN
C     *******************************************************************
C     READ REAL MATRICES WITHOUT PRINTING ANYTHING
C     *******************************************************************
```

```
C     SPECIFICATIONS:
C     ----------------------------------------------------------------
      CHARACTER*20 FMTIN
      DIMENSION A(JJ,II)
C     ----------------------------------------------------------------
C
      READ(IN,1) LOCAT,CNSTNT,FMTIN,IPRN
    1 FORMAT(I10,F10.0,A20,I10)
C
      IF(LOCAT) 200,50,90
C       LOCAT=0
   50 DO 80 I=1,II
      DO 80 J=1,JJ
   80 A(I,J)=CNSTNT
      RETURN
C       LOCAT>0
   90 DO 100 I=1,II
  100 READ(LOCAT,FMTIN)(A(J,I),J=1,JJ)
      GO TO 300
C       LOCAT<0
  200 LOCAT=-LOCAT
      READ(LOCAT)
      READ(LOCAT) A
C       MULTIPLY BY A CONSTANT
  300 IF(CNSTNT.EQ.0.) RETURN
      DO 310 I=1,II
      DO 310 J=1,JJ
  310 A(J,I)=A(J,I)*CNSTNT
      RETURN
      END


C=====================================================================
      SUBROUTINE UNORM(U,RNORM,IP)
C-----VERSION 1000 01FEB1992
C     ****************************************************************
C     FIND THE PROBABILITY RELATED TO A U (IP=1), OR A U RELATED TO A
C     PROBABILITY (IP=-1) FOR A STANDARD GAUSSIAN DISTRIBUTION
C     ****************************************************************
C       SPECIFICATIONS:
C     ----------------------------------------------------------------
      DIMENSION PNORM(2,41)
      DATA (PNORM(1,I),I=1,41)/0.0,.15,.20,.25,.30,.35,
     1       .40,.45,.50,.55,.60,.65,.70,.75,.80,
     2       .85,.90,.95,1.00,1.05,1.10,1.15,1.20,1.25,
     3       1.30,1.35,1.40,1.45,1.50,1.55,1.60,1.65,
     4       1.70,1.75,1.80,1.85,1.90,1.95,2.20,
     5       4.00,5.5/
      DATA (PNORM(2,I),I=1,41)/.5,.5596,.5793,.5987,.6179,.6368,
     1       .6554,.6736,.6915,.7088,.7257,.7422,.7580,.7734,.7881,
     2       .8023,.8159,.8289,.8413,.8531,.8643,.8749,.8849,.8944,
     3       .90320,.91149,.91924,.92647,.93319,.93943,.94520,.95053,
     4       .95543,.95994,.96407,.96784,.97128,.97441,.98610,
     5       .9999683,1.0/
C     ----------------------------------------------------------------
C
C-----GIVEN U, GET THE CUMULATIVE PROBABILITY
      IF(IP.EQ.1) THEN
C-------FIND THE VALUES ABOVE AND BELOW U
        AU=ABS(U)
        IF(AU.GE.5.5) THEN
          RNORM=1.0
          IF(U.LT.0.0) RNORM=0.0
          RETURN
        ENDIF
        DO 100 I=1,40
  100     IF(AU.GE.PNORM(1,I).AND.AU.LT.PNORM(1,I+1)) GO TO 150
        STOP 'ERROR IN UNORM -- U NOT FOUND'
C-------INTERPOLATE
  150   FACTOR=(AU-PNORM(1,I))/(PNORM(1,I+1)-PNORM(1,I))
        RNORM=PNORM(2,I)+FACTOR*(PNORM(2,I+1)-PNORM(2,I))
        IF(U.LT.0) RNORM=1.0-RNORM
        RETURN
      ENDIF
C-----GIVEN THE CUMULATIVE PROBABILITY, GET U
      IF(IP.EQ.-1) THEN
C-------FIND THE VALUES ABOVE AND BELOW RNORM
        ARNORM=RNORM
        IF(RNORM.LT..50) ARNORM=1.-RNORM
        IF(ARNORM.EQ.1.0) THEN
          U=5.5
          IF(RNORM.LT..5) U=-5.5
          RETURN
```

```
         ENDIF
         DO 170 I=1,40
 170       IF(ARNORM.GE.PNORM(2,I).AND.ARNORM.LT.PNORM(2,I+1))
    1        GO TO 190
         STOP 'ERROR IN UNORM -- RNORM NOT FOUND'
C-------INTERPOLATE
 190     FACTOR=(ARNORM-PNORM(2,I))/(PNORM(2,I+1)-PNORM(2,I))
         U=PNORM(1,I)+FACTOR*(PNORM(1,I+1)-PNORM(1,I))
         IF(RNORM.LT..50) U=-U
         RETURN
       ENDIF
       END


C======================================================================
       SUBROUTINE UREADF(C1,NCOL,NROW,NLAY,IU)
C-----VERSION 1000 01FEB1992
C      ****************************************************************
C      READ NODAL ARRAY FROM UNFORMATTED FILE
C      ****************************************************************
C         SPECIFICATIONS:
       DIMENSION C1(NCOL,NROW,NLAY)
C      ----------------------------------------------------------------
       DO 110 IL=1,NLAY
       READ(IU)
  110  READ(IU) ((C1(IC,IR,IL),IC=1,NCOL),IR=1,NROW)
       RETURN
       END


C======================================================================
       SUBROUTINE USAVEF(C1,C2,D1,D2,NODES,IFLAG)
C-----VERSION 1000 01FEB1992
C      ****************************************************************
C      PUT ELEMENTS OF ONE ARRAY INTO ANOTHER ARRAY
C      ****************************************************************
C         SPECIFICATIONS:
       DOUBLE PRECISION D1(NODES),D2(NODES)
       DIMENSION C1(NODES),C2(NODES)
C      ----------------------------------------------------------------
       IF(IFLAG.EQ.1) THEN
         DO 100 I=1,NODES
  100    C2(I)=C1(I)
       ENDIF
       IF(IFLAG.EQ.2) THEN
         DO 110 I=1,NODES
  110    D2(I)=C1(I)
       ENDIF
       IF(IFLAG.EQ.3) THEN
         DO 120 I=1,NODES
  120    C2(I)=D1(I)
       ENDIF
       RETURN
       END
```


*U.S. GOVERNMENT PRINTING OFFICE 1992-0-673-207/50020