A Lake Package for MODFLOW (LAK2)

Documentation and User's Manual
Version 2.2

Gregory W. Council
June, 1999

HSI GEOTRANS
A TETRA TECH COMPANY

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1 Overview

A new package for MODFLOW (McDonald and Harbaugh, 1988, Harbaugh and McDonald, 1996) has been developed for simulating lake-groundwater interaction. This new Lake Package (LAK2) provides features and capabilities that can be useful for modelers familiar with MODFLOW who desire to simulate one or more lakes with variable, and potentially unknown, water levels.

The Lake Package is easily incorporated into MODFLOW's modular program structure. Like many of the standard MODFLOW packages (e.g. River, General-Head Boundary, Drain, Well), the Lake Package provides boundary conditions for the mathematical formulation of the groundwater flow system. However, unlike the standard boundary-condition packages, the Lake Package contains routines to calculate water budgets for a lake that overlies many groundwater cells, and the package can subsequently update the lake water level (stage), volume, and area, as a result of the computed water budget.

The Lake Package can be particularly useful for predicting the effect of certain types of underground development (such as well pumping and mining) on nearby water bodies. In these situations the modeler often desires to know how a lake will respond to the development. Because the Lake Package treats lake stage as a dependent variable, it can calculate the long-term (steady) state of an affected lake, and it can be used to estimate the how quickly lake stage would change during development.

Traditional methods of simulating lakes in MODFLOW include: 1) using one of the original MODFLOW boundary conditions (i.e. constant head, river, drain, well, recharge, or general-head), 2) using high-conductivity, high-storage aquifer cells, and 3) using other lake or reservoir packages developed for MODFLOW. The original boundary conditions allowed by MODFLOW require prior knowledge of the stage of a lake or its seepage rate to the groundwater. These methods also do not provide a way to automatically update stage as a result of changing water fluxes to and from a lake. The method of high-conductivity, high-storage cells can be difficult to implement in certain situations and may lead to instability in the solution of groundwater head. Other packages for MODFLOW, notably the original Lake Package (LAK1) (Cheng and Anderson, 1993) and the Reservoir Package (RES1) (Fenske et al., 1996), were designed to offer greater flexibility in simulating lake-groundwater interaction. The newer Lake Package (LAK2) discussed here expands the capabilities of the earlier packages for even greater functionality.

1.2 Conceptualization

Figure 1 is a cross-sectional view of a typical lake showing the various components of its water budget. Water flows into the lake via precipitation, overland runoff, and feeding streams. Water outflows include evaporation and flow to outlet streams. Groundwater can be an inflow or outflow component for the lake. On some lakes, groundwater may flow into the lake in certain regions and flow out of the lake in others. The state of the lake is described by its stage. The lake water volume and surface area are functions of the lake stage.

The Lake Package is designed to have two primary functions: 1) serve as a boundary condition for the mathematical description of groundwater flow in a fashion similar to that of the River Package, and 2) calculate the water budget of the lake during simulation, allowing for the stage of the lake to vary in response to budget changes.

The Lake Package communicates with MODFLOW's Streamflow Routing Package (STR2) (Prudic, 1989) for lake-stream interactions. The flow from a stream to a lake is found by determining the outflow of the last reach on the appropriate stream segment, which is calculated by the Streamflow Routing Package. The Lake Package can calculate stream outflow to one or more streams, based on user-defined stage-discharge relationships. The calculated lake-to-stream flows then automatically become the inflows to the first reach of the outflowing streams.

Figure 2 illustrates how two lakes may overlie portions of a MODFLOW domain. In this example, several streams segments would also be modeled using the Streamflow Routing Package, and the connections between the stream segments and the lakes would be specified as part of the Lake Package input. The Lake Package input would also specify the hydraulic conductance between the lakes and each underlying groundwater cell, the maximum area of the lakes over each cell, the lakebed top and bottom elevations at each cell, and the parameters to describe inflows and outflows to each lake in each modeled stress period.

The Lake Package provides four modes for lake stage simulation: 1) a temporally constant specified stage, 2) a temporally interpolated specified stage, 3) steady-state stage calculation, and 4) transient stage calculation. Like the River Package, the Lake Package allows the user to input a known or anticipated lake stage for each simulated stress period, which is held constant for each time step in that period. The Lake Package can also use a beginning and ending stage for each stress period to determine, through interpolation, the stage at each time step, similar to the Reservoir Package. In steady-state mode, the lake stage is computed for each MODFLOW estimation of head such that the total lake inflows are equal to the total lake outflows, providing mass balance. In a transient simulation, the Lake Package can calculate the change in lake stage after the completion of each MODFLOW time step in a manner similar to that of the original (LAK1) Lake Package.

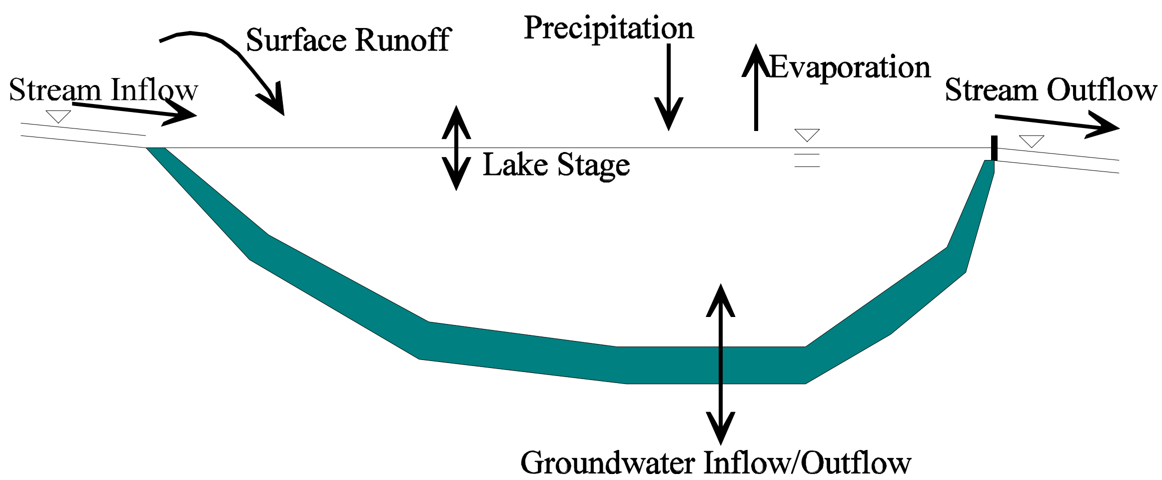


Figure 1. Cross Sectional View of a Lake Showing its Volumetric Budget Components

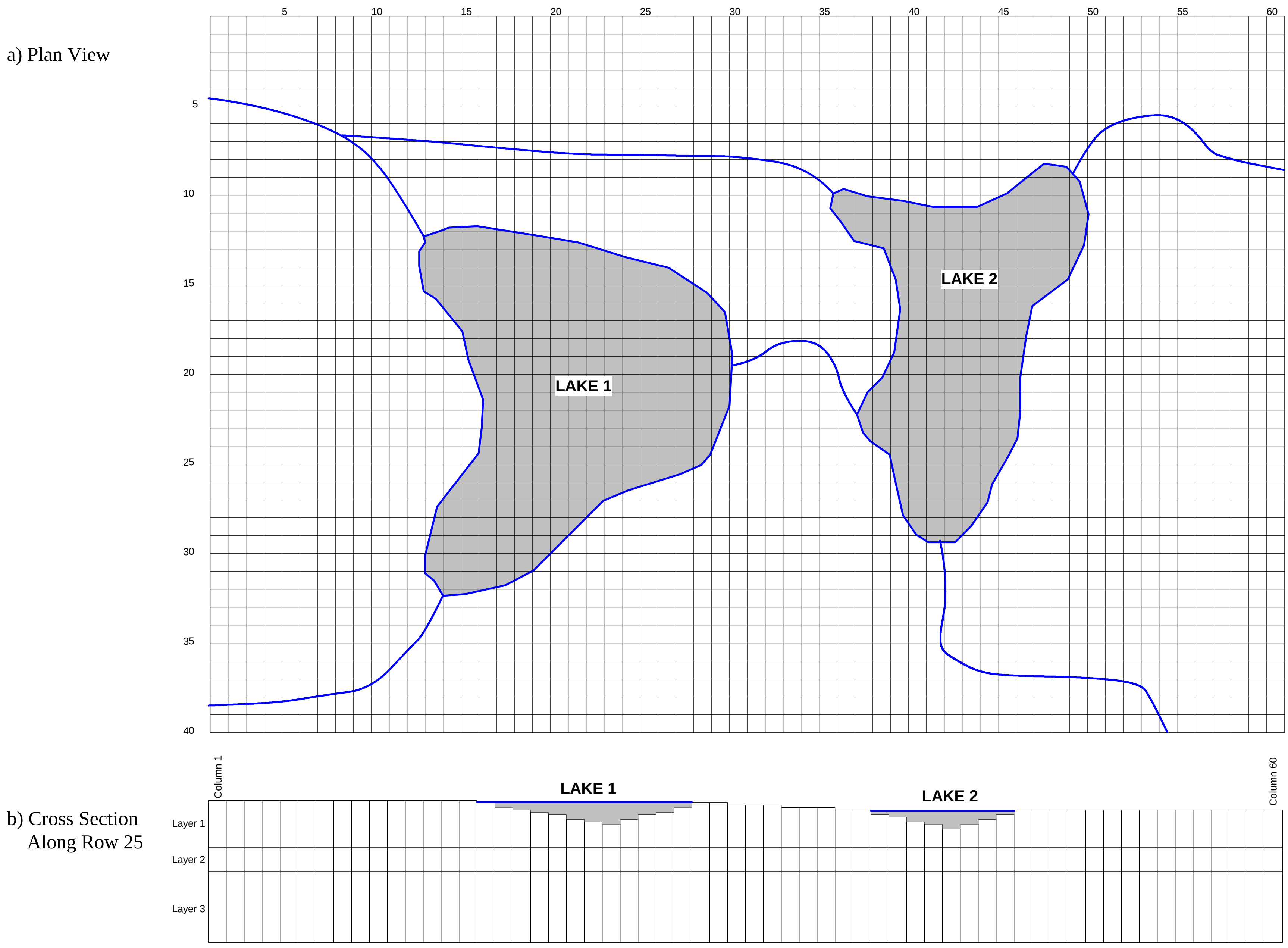


Figure 2. Example MODFLOW Model Grid with Two Lakes and Connecting Streams

1.3 Features

The following features are included in the Lake Package:

- simple integration in MODFLOW-96, including an interface with the Streamflow Routing Package (STR2)
- intuitive list-based input structure, similar to other MODFLOW boundary condition packages
- definition of lake cells specified for each simulation, definition of precipitation, evaporation, and runoff specified for each stress period
- treatment of head as an independent (known) or dependent (unknown) variable
- two specified stage simulation modes: constant stage and smoothly varying (interpolated) stage
- two calculated stage modes: transient and steady-state
- lake area (number of cells that represent the lake) and lake volume changes as the stage changes
- assignment of a default recharge rate to a shore cell (where stage is below the top of lakebed)
- precipitation/evaporation calculated based on either the maximum lake area (including shore cells) or the current area (not including shore cells)
- allowance for user-specified runoff inflow or direct withdrawal rates
- stream outflow calculated using flexible user-defined stage-outflow relationships
- flexible output options including printing of lake budget information in the list file, printing of stage/budget information to a separate output file, and printing of cell-by-cell lake-groundwater fluxes
- allowance for horizontal (side-cell) connections as well as vertical (top-cell) connections
- limitation of in seepage outflow where groundwater head falls below the bottom of lakebed
- allowance for groundwater inflow to shore cell where groundwater head goes above the top of lakebed

1.4. Development History

The Modular Finite-Difference Ground-Water Flow Model (MODFLOW) was developed by McDonald and Harbaugh of the USGS in 1988. The code was reissued with added features in 1996 as MODFLOW-96. The code solves (via iterative approximations in discretized space and time) the groundwater flow equation, which is a combination of the continuity equation and Darcy's Law:

$$\frac{\partial}{\partial x}\left(K_x \frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y \frac{\partial h}{\partial y}\right) + \frac{\partial}{\partial z}\left(K_z \frac{\partial h}{\partial z}\right) = S_s \frac{\partial h}{\partial t} \qquad (1)$$

In equation (1) the dependent variable, h, is potentiometric head [L], a function of space and time. The independent variables are the spatially variable hydraulic conductivity (K_x, K_y and K_z) [L/T] and specific storage (S_s) [L^{-1}] fields. Together with initial conditions for potentiometric head and various boundary conditions, MODFLOW uses a discretized, algebraic form of equation (1) to solve for the potentiometric head at every model cell at time steps within each simulated stress period.

Boundary conditions allowed by MODFLOW include the specified-head, specified-flux, and head-dependent flux types. Boundary conditions are specified in MODFLOW through many of its various packages, or modules, which originally included: Recharge (RCH), Well (WEL), River (RIV), Drain (DRN), and Evapotranspiration (EVT). Another package was later added for streamflow routing (STR1, revised to STR2, Prudic, 1989).

1.4.1 Packages to Simulate Lake-Groundwater Interaction

At least two previously-documented packages have been written to simulate lakes with MODFLOW: the Reservoir Package (RES1), and the original Lake Package (LAK1). The Reservoir Package (RES1, Fenske et al., 1996) works like the River (RIV) Package, but allows for a specified, linearly-varying (in time) lake stage. The known stage is used to determine the number of cells that are covered by the lake at each timestep, and to determine the amount of flow to and from the groundwater. The RES1 package was not designed to calculate the stage in response to environmental stresses. It is not connected to the Streamflow Routing Package.

The original Lake Package (LAK1) developed by Xiangxue Cheng and Mary Anderson (Cheng, 1994, Cheng and Anderson, 1993) includes many of the functions in the newer LAK2 package. In providing boundary conditions for equation (1), LAK1 also behaves like the River Package (RIV). Additionally, it calculates lake stage as a transient response to evaporation, precipitation, streamflow, and groundwater flux. The LAK1 package handles lake-stream interaction with a modified version of the original Streamflow Routing Package (STR1). The package does not provide for steady-state solution of lake stage, and requires the use of Manning's equation to calculate flow from a lake to an adjoining stream, based on the stage in the lake.

The new Lake Package (LAK2) described here includes all of the capabilities of the RES1 and LAK1 packages, and includes new features to allow for greater use. First, computation of steady-state lake stage is possible, using a modified version of Newton's Method. The steady-state lake stage represents the stage at which lake inflow (from precipitation on the lake, overland runoff, stream inflow, and groundwater flux) is balanced by lake outflow (to streams, groundwater, and the atmosphere via evaporation). Also, the relationship describing stream outflow has been made very general, to accommodate a wide variety of stage-outfall relationships. The LAK2 package is a completely new code, which improves the input file structure, output options, and memory requirements of the LAK1 package. It is easy to connect to MODFLOW, seamlessly connecting with the current Streamflow Routing Package (STR2) for stream-lake interaction.

1.4.2 Other Lake-Groundwater Models

Lake packages for other groundwater flow models have previously been developed, as documented in Cheng and Anderson, 1993. Additionally, fully integrated surface-water/groundwater models have been and are being developed that vary in terms of capabilities and complexity (see Yan and Smith, 1994 for example).

2. LAK2 Description

The LAK2 Lake Package provides two major functions: 1) it formulates boundary conditions for the system of equations MODFLOW uses to solve equation (1), and 2) it computes lake-wide budget and stage information. These two functions are related through the lakebed hydraulic conductance, which controls the degree of lake-groundwater interaction.

2.1 Formulation of the Groundwater Boundary Condition

The Lake Package formulates the head-dependent boundary condition for MODFLOW in a manner that is very similar to the River Package. There is a slight difference, however, at shore cells (where lake stage is below the lakebed top elevation). Here, the boundary condition is similar to that of the Drain Package — water is allowed to flow from the groundwater to the lake only when the groundwater head is greater than the lakebed top elevation.

In the Lake Package input file, the user specifies the active MODFLOW cells that are connected to each modeled lake (input file instructions are provided in Appendix B). Because they are groundwater boundary conditions, the modeled lakes technically lie outside of the MODFLOW groundwater domain (comprising instead the Lake Package domain). For this reason, if the MODFLOW discretization causes some cells to lie inside of a lake, then those cells should be specified as inactive (in the Basic Package IBOUND array), and the adjacent groundwater cells should be specified in the Lake Package input file (see section 3.1 for tips on modeling an "incising" lake). If instead the lake lies on top of layer 1 (as in Figure 2), then the underlying top-layer cells should be specified in the Lake Package input. The MODFLOW *groundwater* cells (specified layer, row, column) that are connected to a lake, are heretofore termed "lake cells" to identify them as being connected to a lake in the Lake Package (much as river cells are included in the River Package and drain cells are included in the Drain Package).

Along with each lake cell's layer-row-column position, the user also specifies the surface area of the lake cell (A) (projected on a horizontal plane), the top (Top) and bottom (Bot) elevations of the lakebed and either the hydraulic conductivity (K) or the hydraulic conductance ($Cond$) of the lakebed. If the hydraulic conductivity is specified, then the Lake Package calculates the lakebed hydraulic conductance as:

$$Cond = \frac{KA}{Top - Bot} \tag{2}$$

During each MODFLOW solution iteration, the Lake Package reformulates the lake boundary for each specified cell using the current values of the lake stage (S) and groundwater head (h) in the connected cell. As shown in Figure 3, the lake-to-groundwater flux is dependent on the values of stage and head relative to the lakebed top and bottom elevations. Consequently,

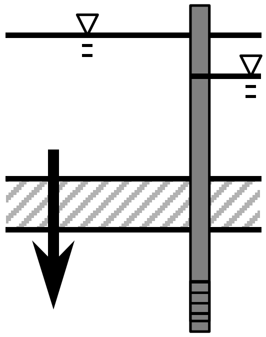
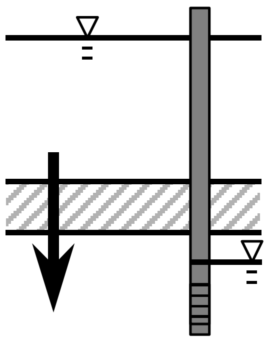
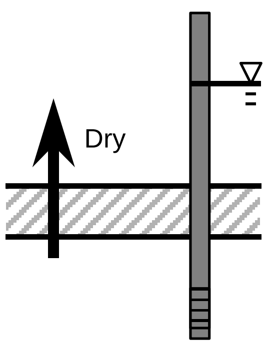
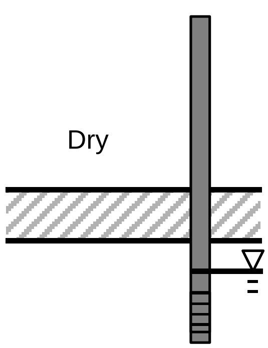
Illustration (cell *i*)	Condition	Flow to Aquifer	Formulation
Connected 	$S \geq TOP_i$ AND $h_i > BOT_i$	$Q_i = COND_i(S - h_i)$	$RHS_i = RHS_i - COND_i * S$ $HCOF_i = HCOF_i - COND_i$
Maximum Seepage 	$S \geq TOP_i$ AND $h_i \leq BOT_i$	$Q_i = COND_i(S - BOT_i)$	$RHS_i = RHS_i - COND_i(S - BOT_i)$
Shore Drain 	$S < TOP_i$ AND $h_i > TOP_i$	$Q_i = -COND_i(h_i - TOP_i)$	$RHS_i = RHS_i - COND_i * TOP_i$ $HCOF_i = HCOF_i - COND_i$
Shore Disconnected 	$S < TOP_i$ AND $h_i \leq TOP_i$	$Q_i = 0$	NO FORMULATION

Figure 3. Computation of Lake-to-Groundwater Flux and MODFLOW Boundary Condition Formulation

the formulation of the boundary condition also depends on these elevations. The MODFLOW variables *HCOF* and *RHS* are updated at each lake cell *i* according to the equations listed in Figure 3. Note that sign on the terms added to *RHS* and *HCOF* follow MODFLOW conventions.

2.2 Volumetric Budget Components

Unlike the standard MODFLOW boundary-condition packages, the Lake Package treats each modeled lake as a separate domain connected to the MODFLOW groundwater domain through any number of lake cells. Depending on the lake simulation mode chosen by the user, the Lake Package can automatically adjust the lake stage due to fluxes into or out of the lake. Regardless of the simulation mode chosen, the Lake Package calculates the volumetric water budget for each modeled lake, much as MODFLOW calculates the groundwater volumetric budget. The components of a lake budget are shown in Figure 1. The general volumetric-balance equation for a lake is:

$$\frac{\partial V}{\partial t} = Q_P + Q_{RO} + Q_{STRIN} + Q_{GW}(S) + Q_E(S) + Q_{STROUT}(S) \qquad (3$$

Equation (3) simply states that the time rate of change in lake volume (*V*) is equal to the sum of all inflows and ouflows from precipitation (Q_P), runoff (Q_{RO}), inflowing streams (Q_{STRIN}), groundwater (Q_{GW}), evaporation (Q_E), and outflowing streams (Q_{STROUT}). Note in this equation that all terms are positive for flows into the lake. Also note that Q_{GW}, Q_E, and Q_{STROUT} are functions of stage (*S*), whereas Q_P, Q_{RO}, and Q_{STRIN} are independent of stage.

In the Lake Package, the user specifies a precipitation rate and an evaporation rate (in length/time units) through the *PRECIP* and *EVAP* input variables for each stress period. The Lake Package calculates the precipitation flux as the *PRECIP* rate times the total area of the lake, regardless of lake stage. The Lake Package calculates the evaporation flux as the *EVAP* rate times the current “wetted” lake area (the sum of the areas of all cells having a lakebed top elevation below the current stage). The *EVAP* value should be specified as a negative number to indicate a lake outflow. In this formulation, evaporation occurs only on the lake surface, whereas precipitation falling either on the lake surface or on the lake shore is added to the lake (i.e. precipitation on the lake shore runs off into the lake). As described in section 3.3, the Lake Package input can be modified slightly for a conceptualization where only direct precipitation on the lake surface is included in the lake budget.

The user can also specify a runoff inflow rate for each stress period (in cubic length/time units) via the *RUNOFF* variable. The *RUNOFF* variable can also be used to add a known stream inflow to a lake’s budget (instead of using a stream inflow calculated by the Streamflow Routing Package). This variable can also be used to represent a direct withdrawal from the lake (a negative value should be specified in that case). The value of *RUNOFF* is added directly to the lake’s budget.

The Lake Package determines stream inflow by summing the values of the Streamflow Routing Package's *ARTRIB* array for all specified stream inflow segments. The *ARTRIB* array contains the streamflow out of the last reach (cell) of each stream segment.

Stream outflow is calculated based on user-defined stage-discharge relationships (also termed "rating equations") for any number of outflowing streams. The computed outflow to any stream is then (optionally) inserted as the inflow to a Streamflow Routing Package segment by the Lake Package (the *STRM* array is used). In the Lake Package, rating equations must have the general form:

$$Q_{STROUT} = C(S - E)^{P} \tag{4}$$

In equation (4), the multiplicative constant, *C*, the reference elevation, *E*, and the exponent, *P*, are specified in the Lake Package via the input variables *CONST*, *ELEV*, and *EXPNT*. Also, a lower stage limit is supplied for the equation, via the *CUTOFF* variable, such that the equation is only valid when the lake stage is above the equation's lower limit. The value of *CUTOFF* is often the same as the value of *ELEV*. The form of equation (4) allows the use of many empirically-derived stage-discharge relationships, some standard weir equations, and Manning's equation as in the original LAK1 Lake Package. Also, the Lake Package allows the user to specify multiple rating equations for any stream outflow, where a different equation applies over different ranges of stage. This allows for a piece-wise linear stage-discharge relationship and other, more complicated relationships. Section 3.4 provides examples of stage-discharge relationships.

Groundwater seepage from the lake to the aquifer (and vice-versa) is based on the lake stage, groundwater head at lake cells, and the lake-cell conductances. Each cell's contribution to the total groundwater seepage is calculated according to the budget equations in Figure 3. Note that if the groundwater head falls below the lakebed bottom elevation, the seepage at that cell goes to a maximum value based on the difference between the stage and the lakebed bottom elevation (river-like behavior). Also, if the stage of the lake falls below the top elevation of a lake cell, then water can only flow from aquifer to lake, which happens only if the groundwater head is above the top elevation (drain-like behavior). In calculating the volumetric budget of a lake, flow from the groundwater to the lake is considered positive. Conversely, when the volumetric budget of the aquifer is computed (for MODFLOW's overall budget and cell-by-cell flow terms), flow from the lake to the groundwater is positive. Thus, a positive sign always indicates flow into the domain being considered.

The volume and (wetted) area of the lake are updated each time the stage changes. The area of the lake is the sum of individual lake cell areas that have a top elevation below the current stage. The volume of water in the lake is determined by summing the water above each wetted cell, which is calculated as the cell area times the height of the water over the top elevation.

2.3 Fixed Stage

If the lake stage is known, it can be treated as an independent variable and specified directly in the Lake Package. Two options are provided for specifying the known lake stage: constant stage and interpolated stage.

In constant stage mode (activated by specifying *ISIMMODE* = 0 for a lake in the Lake Package input file), the user inputs the stage for each stress period, and that stage is used for each time step in the stress period. In this mode, the behavior of the Lake Package is similar to the River Package in that the stage of the lake can only change at the end of each stress period.

In interpolated stage mode (*ISIMMODE* = 1), the user specifies an initial lake stage and the stage at the end of each stress period. The Lake Package calculates the stage for each time step within a stress period using linear interpolation. For each successive time step within a stress period, the stage is updated to change from the starting stage to the ending stage for the period. The stage for the last time step of the stress period is the value specified as the ending stage for that period. In order to calculate the interpolated lake stage at each time step, the Lake Package first calculates the rate of change in stage with respect to time for a stress period by subtracting the ending stage from the starting stage and dividing by the stress period length (*PERLEN*). Then, for each time step, the rate of stage change is multiplied by the MODFLOW time step length (*DELT*) and added to the previous stage to get the stage for the current time step. This behavior is similar to that of the Reservoir Package.

2.4 Steady-State Stage Solution

In a steady-state MODFLOW simulation, an equilibrium solution is desired for heads such that all inflows and outflows to the aquifer domain from boundary conditions are in perfect balance. Aquifer storage is set to zero for a steady-state MODFLOW simulation. The steady-state solution is the eventual, or long-term, head field that would result from unchanging boundary conditions. It is also often taken to be the average head field that results from average boundary conditions.

Steady-state lake stage can be similarly calculated by the Lake Package to determine the eventual or approximate average stage that would result from constant (or averaged) stresses, where the stresses are precipitation, evaporation, runoff, etc. At steady-state, a lake’s inflows and outflows are in perfect balance, and the rate of change in lake storage (volume) is zero:

$$Q_P + Q_{RO} + Q_{STRIN} + Q_{GW}(S) + Q_E(S) + Q_{STROUT}(S) = 0 \qquad (5)$$

Conceptually, if the stage-independent flows (precipitation, runoff, and stream inflow) are positive (in net) indicating flow into the lake, then the steady-state stage would be the stage at which the (net) negative outflow from groundwater seepage, evaporation, and stream outflow are equal in magnitude to the inflow. As the stage rises, the outflows generally become greater (the terms become more and more negative), allowing for a unique stage to be determined. If the stage-independent flows are not net positive, then it is possible that even at the minimum lake stage (stage just above the top elevation of the lowest lakebed cell) net flow will be out of the lake. This indicates a dry lake. A dry lake is thus generally indicated by a lake that has a volume of zero and a steady-state net flow out of the lake. Section 2.7.2 presents more information on how dry lakes are handled in the Lake Package.

The steady-state stage is determined at each formulation iteration using the latest estimates for groundwater head and stream inflow. With these values fixed, the net flow into the lake is treated as a decreasing, uniquely-valued, differentiable function of stage (Figure 4). Newton's method is then used, starting with the prior estimate of stage, to determine the stage at which the net flow to the lake is zero. Newton's method is an iterative procedure whereby the current derivative of the net flow with respect to the stage is determined, then the current value of net flow is divided by the derivative to determine the approximate change in stage needed to give a net flow of zero. This iterative process continues until the solution converges to the correct stage (within one-tenth of the stage convergence criterion). If Newton's method fails (due to non-continuity in the net flow function or calculation of a zero-valued derivative), then a less-efficient but more robust range-splitting procedure is used to calculate stage. In the range splitting procedure, the upper-bound stage is set to the lowest stage value that gave a net outflow from the lake, and the lower-bound stage is set to the lowest stage value that gave a net inflow to the lake (the minimum and maximum stage values are used if needed). A guess for lake stage is made that is halfway between the upper and lower bounds. If this stage produces a net outflow, the upper bound is reset, and if it produces a net inflow then the lower bound is reset. This procedure continues until the stage converges.

The stage solution will not converge in two cases: an empty lake (net outflow at the minimum stage) and a full lake (net inflow at the maximum stage). The minimum stage is determined by the Lake Package as the lowest value of the *TOP* for all lake cells. The maximum lake stage is set by the user in the Lake Package input file, with the variable *STAGEMX*. For an empty lake, the stage is set to its minimum value and any cells that have that *TOP* elevation remain wetted (see section 3.2). Because net flow is negative, a substantially negative steady-state mass balance error will be reported in the lake's budget. Conversely, for a full lake, the stage is set to its maximum, a positive net flow is reported, and the steady-state error is substantially positive.

2.5 Transient Stage Solution

In transient mode, this Lake Package is very similar to the original Lake Package of Cheng and Anderson. As opposed to the steady-state mode, the lake stage is held constant for all iterations in a time step in transient mode. At the end of the time step, the flow rates into the lake are accumulated using the newly-estimated head values and the old stage value to get a net flow into the lake. The net flow is multiplied by the length of the current time step to get the change

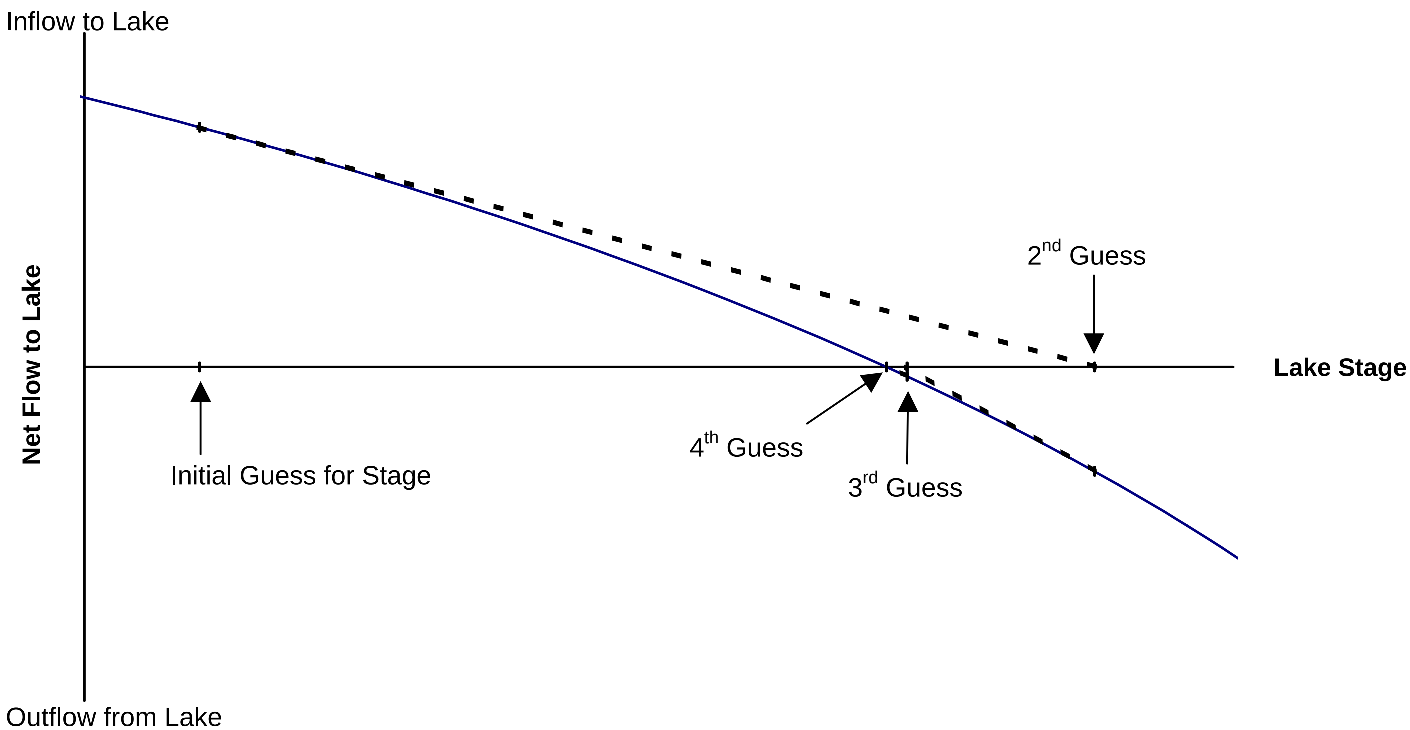


Figure 4. Generalized Relationship Between Stage and Flow, and Illustration of Newton's Method for Finding the Steady-State Stage

in lake volume, and the new "target" volume of the lake. The lake stage is then iteratively adjusted until the target volume is obtained. At each iteration, the next guess for the correct stage is calculated by subtracting the target volume from the current volume, dividing by the current wetted area, and adding that amount to the current stage (this is Newton's method with the wetted area as the derivative of volume with respect to stage). If the lake was initially dry (volume = 0) then the first guess for stage is always halfway between the lake's minimum and maximum stage. If the target volume is zero, then the lake's stage is set to its minimum elevation (equal to the minimum lakebed top elevation) to indicate a dry lake.

Since the lake stage is held constant during the MODFLOW iteration loop, the transient simulation mode is somewhat more stable than the steady-state mode. However, the implicit assumption is that the stage does not change much over the time step. This assumption can necessitate the use of very small time steps.

2.6 Other Features

2.6.1 Top Active Connection

Many times a modeled lake sits on top of the modeled aquifer. In these situations, the user can specify lake cells in layer zero. This will result in the Lake Package determining which layer (for the specified row and column) is the highest-active layer in the model, and connecting the lake to that cell. This is particularly convenient if a cells in the top layer go dry during the simulation, but seepage from the lake should still be allowed to occur.

2.6.2 Sub-Time-Steps

The Lake Package allows the use of sub-time-steps when the transient stage solver is used. Sub-time-steps are activated by setting *NSUBSTEPS* greater than 1 in the Lake Package input file. Each MODFLOW time step is broken down into the specified number of (uniform) sub-time-steps for transient stage updating. The use of sub-time-steps may be useful in some situations. See section 3.3 for suggestions on selecting time step sizes for transient simulations.

2.6.3 Shore-Cell Recharge — NOT SUPPORTED

Note that the Lake Package includes a variable, *DRYRCH*, for applying recharge-type inflow to the aquifer beneath shore cells. This variable was included to accomodate a future code enhancement. At the current time, **the *DRYRCH* option is not fully implemented in the Lake Package code, and is therefore not recommended for use**.

2.7 Limitations

2.7.1 Instability

The Lake Package can decrease the stability of a MODFLOW model, especially when lakes are simulated in steady-state mode. This is because the stage changes (potentially) each MODFLOW iteration, leading to a highly nonlinear aquifer boundary condition. To reduce instability, the user should select very good initial conditions for aquifer head (this can often be done by first running the model with the lake stage fixed), and the user should carefully choose MODFLOW solver and lake-stage solver parameters. Often, more iterations and lower convergence criteria will result in a better solution of head and stage.

2.7.2 Dry Lakes

A lake with no water is not a lake at all. The Lake Package does not model a completely dry lake with any accuracy. If the lake stage drops to its minium value and the lake volume goes to zero, the user should note that the lake is dry. The Lake Package will keep at least one cell active for (potential) seepage so that the lake can reform in later time steps. However, keeping a cell active in the dry condition can lead to overprediction of seepage from the lake to the aquifer, which could lead to an erroneous head solution. The user should be aware of lakes that go dry during a simulation. One method for getting around seepage overprediction for this situation is presented in section 3.2.

2.7.3 Coalescing Lakes

Sometimes two nearby lakes will merge to form a single large lake when the stage rises to a certain level. The Lake Package cannot automatically handle this condition of coalescing lakes. Often, the modeler can set up two models instead — one for the two-lake situation and one for the single-lake situation.

2.7.4 Transient Decoupling

Because the lake and aquifer domains are completely decoupled in transient mode, the time step must often be quite short. If the change in stage or the change in lake-cell heads are large for a time step, then the decoupling assumption is invalid and a smaller time step must be chosen. If a smaller time step yields nearly identical results, then the original time step is appropriate.

3. Suggestions for Use

3.1 Start With Fixed-Stage Mode

For most applications, the first simulation with a MODFLOW model is made in steady-state mode. Even if the analysis focuses on the transient response to stresses, a starting equilibrium condition must first be obtained. For the first steady-state simulation, the modeler often chooses to start from a rather arbitrary initial condition (e.g. uniform initial head or initial head equal to ground surface elevation). Under these circumstances, if the Lake Package is to be used, it is recommended that the modeler begin with the lakes set to fixed-stage mode (*ISIMMODE* = 0), with a reasonable guess made for lake stage.

If instead the modeler uses the steady-state stage solver mode (*ISIMMODE* = 2), and the initial conditions are far from equilibrium, the Lake Package can lead to undesired results. In the first MODFLOW iteration, the lake stage will reset to a value that is in equilibrium with the initial head field. Further iterations may be adversely affected by this initially-calculated stage.

A better plan of action is generally to make the first MODFLOW simulation with all lakes in fixed-stage mode. Then, using the resulting head field as starting heads, a second simulation can be made in steady-state stage solver mode.

The steady-state stage solver is most useful in impact simulations, with initial heads and stages from a pre-stressed simulation, when the long-term (maximum) change in stage as a result of an imposed stress is to be determined. For the impact simulation, the lakes can usually be set to steady-state mode without difficulty. However, if the modeler has trouble getting the simulation to converge, the two-step procedure (fixed-stage, then steady-state stage) may be required.

3.2 Specifying Stage-Discharge Relationships

The form of stage-discharge (outfall) equations for stream outflow have purposefully been made very general to allow for many different types of outfall relationships to be used. The stage-discharge relationship consists of one or more power-function equations in the form shown equation (4). Different equations may apply over different ranges of stage.

For example, consider a beaver dam that allows flow when the stage rises above an elevation of 30 m, with the following relationship (S = Stage, Q = stream outflow in m^3/d):

$$Q = 2.1(S - 30.0)^{3.2} \quad \text{if } S > 30.0$$
$$Q = 0 \quad \text{if } S \le 30.0$$

In this case, one outfall equation is entered with: *CONST* = 2.1, *ELEV* = 30.0, *EXPNT* = 3.2, and *CUTOFF* = 30.0. However, in some cases, another outflow equation may be more correct for low stages such that:

$$Q = 2.1(S - 30.0)^{3.2} \quad \text{if } S > 31.0$$
$$Q = 2.1(S - 30) \quad \text{if } 30.0 < S < 31.0$$
$$Q = 0 \quad \text{if } S \leq 30.0$$

In this case, two outfall equations are entered: 1) *CONST* = 2.1, *ELEV*= 30.0, *EXPNT* = 3.2, and *CUTOFF* = 31.0; and 2) *CONST* = 2.1, *ELEV*= 30.0, *EXPNT* = 1.0, and *CUTOFF* = 30.0. Note that the equations must be specified in decreasing order of *CUTOFF*. The difference between the variables *ELEV* and *CUTOFF* is sometimes subtle; *ELEV* is generally a reference elevation in the stage-discharge equation, and *CUTOFF* defines the lowest elevation at which the equation applies. Often the values of *ELEV* and *CUTOFF* will be the same (as in the first example and the second equation of the second example).

As another example, consider a stream outflow that is governed by Manning’s equation (as required by the original Lake Package — LAK1 — of Cheng and Anderson, 1993):

$$Q = \frac{CW}{n}\left[\frac{2}{3}(S - B)^{5/3}\right]\sqrt{S_c}$$

In this equation, C is a unit-conversion constant, W is the width of the outflowing stream reach, n is the stream’s roughness coefficient, B is the stream bottom (or streambed top) elevation, and S_c is the slope of the stream. This equation can be transformed to the required form of equation (4) by making the following substitutions:

$$CONST = \frac{CW}{n}\left(\frac{2}{3}\right)^{5/3}\sqrt{S_c}$$
$$ELEV = CUTOFF = B$$
$$EXPNT = 5/3$$

The example problem used to illustrate the LAK1 Lake Package (Cheng and Anderson, 1993), includes this type of stream outflow. That problem has been reformulated with this (LAK2) Lake Package and is included as an example problem, along with the Lake Package code and MODFLOW program, in electronic format as part of Appendix D.

Figure 5 contains other examples of stage-discharge relationships, including the specification of input variables in the Lake Package input file.

3.3 Time Stepping

With any transient MODFLOW model, time-step size is an important input parameter. The step size must be small enough to achieve convergence and mass balance throughout the simulation. Furthermore, the step size should be small enough to make the simulation results insensitive to further reductions in step size. In other words, if the resulting head function, $h(x,y,z,t)$ is substantially different when smaller time steps are taken, then the original time steps are too large.

When the Lake Package is used in transient stage solver mode (*ISIMMODE* = 3), selection of time step size becomes even more important. A smaller time step is often required because the transient stage solving procedure effectively decouples the lake stage solution from the groundwater head solution at each time step (the previous time-step's stage is used to calculate heads and the resulting heads are then used to calculate the stage at the end of the current time step). Consequently, if the lake stage or groundwater heads at lake cells change much during any time step, the simulation results will be less accurate.

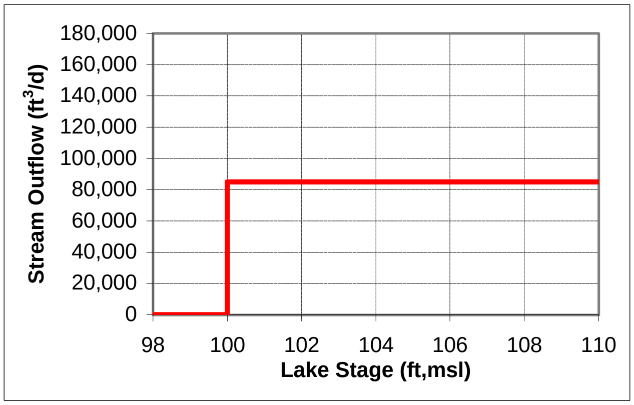
One way to test whether the time stepping used is appropriate is to re-run a simulation with much finer time stepping. If the results (stage, head, and flux vs. time) don't change, then the original time stepping is probably sufficient.

The variable *NSUBSTEPS* can be used to divide each MODFLOW time step into more than one equal sub-time steps. This causes the transient stage solver to take smaller time steps when it updates the stage, which may produce better results when the stage changes rapidly compared to changes in the groundwater head. However, the use of sub-time steps may lead to model instability in some cases, especially when the lakebed conductance is relatively high. Also, even when sub-steps are used, the modeler should still carefully select and test the size of the MODFLOW time steps. If sub-steps seem to cause problems with a model, *NSUBSTEPS* should be set to 1 (or 0) and adjustments in the MODFLOW time step size should be made.

3.4 Modeling Incising Lakes

In some cases, a lake incises the modeled aquifer to such an extent that the lake actually replaces one or more of the modeled layers, instead of just sitting on top of the uppermost model layer (Figure 6). In this case, the following guidelines should be followed:

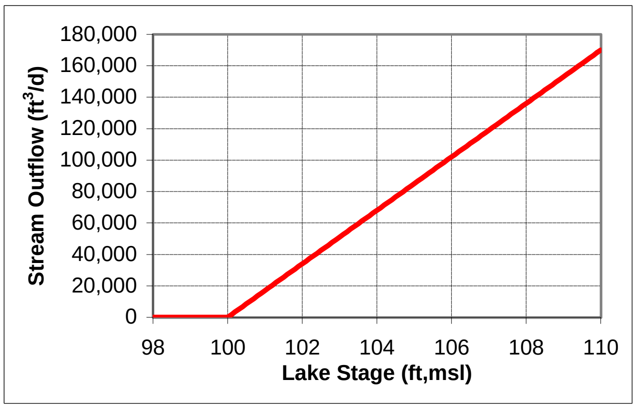
1) Set *IBOUND* = 0 where the lake replaces aquifer. The Lake Package will compute the (uniform) head inside the lake -- it should therefore not be part of the MODFLOW (aquifer) solution.



Constant Outflow Above Cufoff Elevation

Q = 85000 for S > 100
Q = 0 for S <= 100

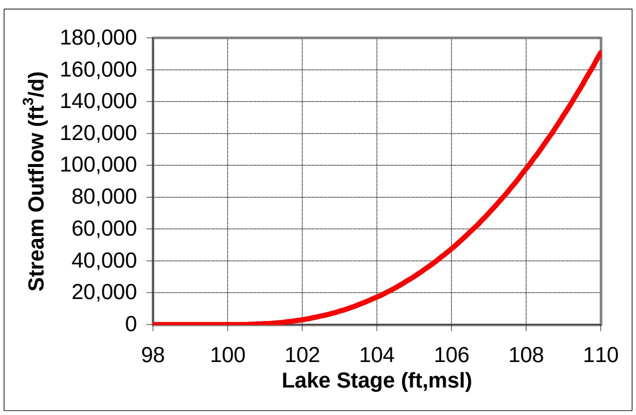
	CUTOFF	CONST	ELEV	EXPNT
1	100	85000	Arbitrary	0



Linear Stage-Outflow Relationship

Q = 17000(S-100) for S > 100
Q = 0 for S <= 100

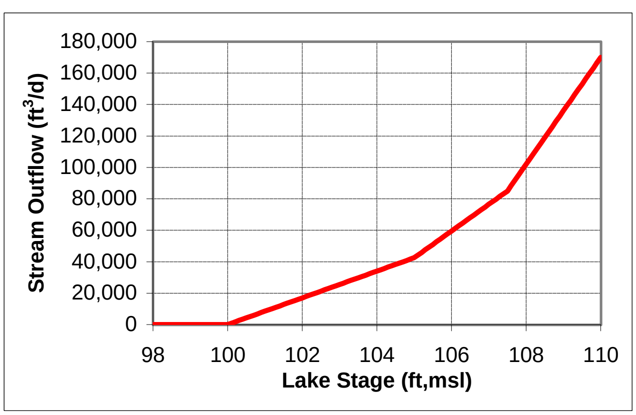
	CUTOFF	CONST	ELEV	EXPNT
1	100	17000	100	1



Weir-Type Rating Equation

$Q = 540(S-100)^{2.5}$ for S > 100
Q = 0 for S <= 100

	CUTOFF	CONST	ELEV	EXPNT
1	100	540	100	2.5



Piece-Wise Linear Relationship

Q = 34000(S-105) for S > 107.5
Q = 17000(S-102.5) for 105 < S < 107.5
Q = 8500(S-100) for 100 < S < 105
Q = 0 for S <= 100

	CUTOFF	CONST	ELEV	EXPNT
1	107.5	34000	105	1
2	105	17000	102.5	1
3	100	8500	100	1

Figure 5. Examples of Stage-Discharge Relationships and Specification of Lake Package Variables.

2) In the Lake Package, specify all active (*IBOUND*>0) cells that are connected to the lake — either horizontally (side nodes) or vertically (bottom nodes).

3) Use conductances, not conductivities. Specify *ICONDOP* = 1 and calculate all lake-cell conductances prior to running the model. This allows the setting of the *AREA* variable to zero (see guideline #8) without also forcing the conductance to zero.

4) Calculate the conductance of the low-conductivity lakebed, if it exists, and use this conductance for all lake connections.

5) If there is no low-conductivity lakebed for a connection, compute the conductance from the lake to the center of the connected cell using the properties of the connected cell.

6) The *TOP* variable will be used to determine if the lake connection is active (Active if Stage >= *TOP*). For side nodes, *TOP* should generally be set to the center elevation of the connected cell. If the side connection is to be always active, then the *TOP* variable can be set arbitrarily low.

7) The *BOT* variable is used to limit the flow from the lake when a desaturated area forms below the lakebed (similar to *RBOT* in the River Package). For side nodes, *BOT* should be set at or below the bottom of the connected cell, and can be arbitrarily low.

8) *AREA* is used to determine the precipitation and evaporation fluxes. For side nodes, *AREA* should be set to zero.

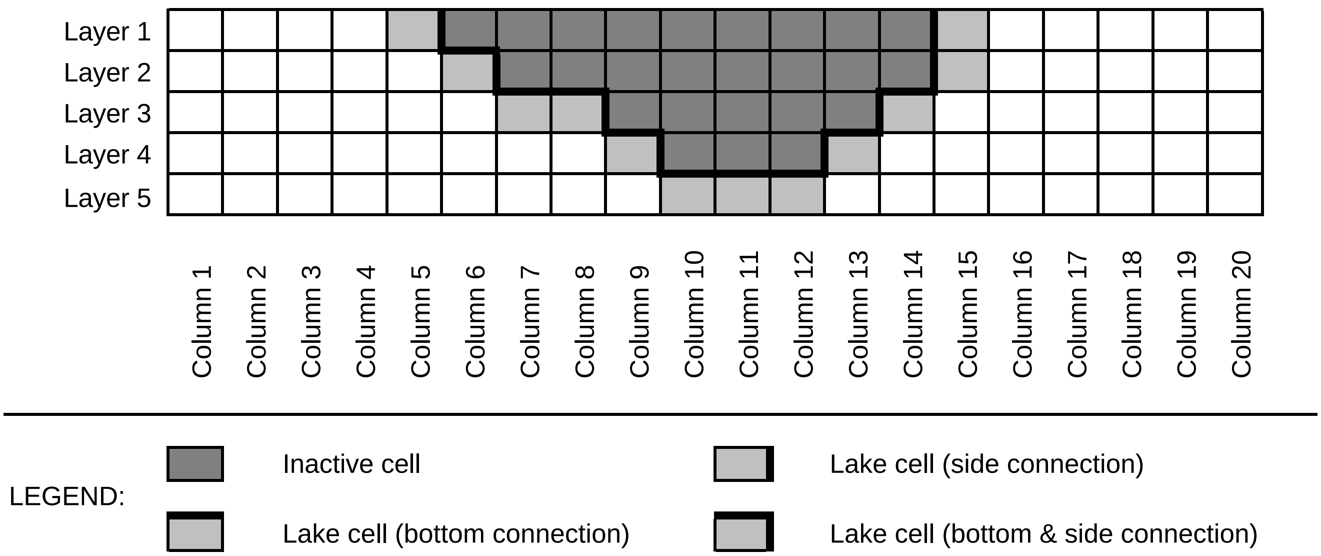
Figure 6 shows a hypothetical cross-sectional model with a lake that incises three model layers (unlike the lakes of Figure 2 which are located on top of model layer 1). The shading of the cells indicates which cells are specified as inactive, and which are included in the Lake Package. Note that some groundwater cells are connected to the lake both horizontally and vertically, and should be listed twice in the Lake Package input file.

3.5 Making Precipitation a Stage-Dependent Inflow

Precipitation and evaporation rates are specified (in length/time units) in the Lake Package input file using the *PRECIP* and *EVAP* variables. In the Lake Package budget routine (*LAKEBUD* subroutine), the *PRECIP* rate is multiplied by the lake's total (maximum) area, and the *EVAP* rate is multiplied by the current wetted area. Both flux terms are then added to the lake's volumetric budget.

Under the designed conceptualization, the *PRECIP* variable has a positive value corresponding to the areal precipitation rate, and the *EVAP* variable has a negative value equal to the areal evaporation rate (the user is responsible for applying the correct sign to the input values, positive always indicating inflow to the lake). With this conceptualization, precipitation that

Example Cross-Sectional Model (1 Row)



Inactive Cells (IBOUND = 0)

L	R	C
1	1	6
1	1	7
1	1	8
1	1	9
1	1	10
1	1	11
1	1	12
1	1	13
1	1	14
2	1	7
2	1	8
2	1	9
2	1	10
2	1	11
2	1	12
2	1	13
2	1	14
3	1	9
3	1	10
3	1	11
3	1	12
3	1	13
4	1	10
4	1	11
4	1	12

Lake Package Cells

L	R	C	Connection
2	1	6	Bottom
3	1	7	Bottom
3	1	8	Bottom
4	1	9	Bottom
5	1	10	Bottom
5	1	11	Bottom
5	1	12	Bottom
4	1	13	Bottom
3	1	14	Bottom
1	1	5	Side
2	1	6	Side
3	1	8	Side
4	1	9	Side
4	1	13	Side
3	1	14	Side
2	1	15	Side
1	1	15	Side

Figure 6. Example of an Incising Lake and Specification of Lake Cells.

falls on a lake's shore cells (lakebed *TOP* above stage) runs off directly into the lake. Conversely, evaporation only occurs over the water surface (wetted area).

In some cases, an alternative conceptualization may be desired where precipitation should only be applied to the wetted lake area (like evaporation). This can be accomplished by specifying a zero for *PRECIP* and specifying the net precipitation (precipitation minus evaporation) for *EVAP*. Under this conceptualization and input procedure, precipitation falling on shore cells is not accounted for in the lake volumetric budget (it may be considered to immediately evaporate), and the net precipitation is applied over the current wetted lake area. With this formulation, the *EVAP* variable can be positive (precipitation greater than evaporation) or negative (evaporation greater than precipitation). In the budget printing routine, lake inflows and outflows are reported separately. Therefore, a positive *EVAP* value results in a lake inflow that is labeled precipitation (it is the net precipitation inflow), and a negative *EVAP* value results in a lake outflow that is labeled evaporation (it is the net evaporation outflow),

Note that the Lake Package includes a variable, *DRYRCH*, for applying recharge-type inflow to the aquifer beneath shore cells. At the current time, **the *DRYRCH* option is not fully implemented in the Lake Package code, and is therefore not recommended for use**.

3.6 Simulating Dry Lakes with a Dummy Cell

As mentioned in Section 2.7.2, the Lake Package was not designed to simulate dry lakes with great accuracy. A dry lake can be identified in the Lake Package output by a reported lake volume of zero and a reported stage equal to the lake's minimum stage.

The lake's minimum stage is determined by the lowest value of the input lakebed *TOP* elevation. When this elevation is reached, the lake cell or cells having this *TOP* elevation will continue to be wetted (all others will be dry) and will therefore behave like River Package cells having a stage equal to *TOP*. That means that seepage from the lake to the groundwater will remain possible for the cell or cells, depending on the cell conductance(s), the aquifer head and the specified *BOT* values. With this behavior, the lake will reform (volume will go above zero in steady-state or transient stage solver mode) when inflows to the lake exceed the seepage outflow (if any) at the minimum stage.

If a lake is dry and lake inflows are small or zero, the Lake Package may therefore simulate an aquifer inflow that may not be real. This inflow can lead to higher simulated aquifer heads than would be expected if the lake were not present at all.

To circumvent this potential misrepresentation, a "dummy" lake cell can be added to the definition of the lake. The dummy cell should have a conductance and area of zero, and a *TOP* elevation lower than all other cells for the lake. Because it has a conductance of zero, the dummy cell can be any cell in the model (layer 1, row 1, column 1 will always work). The dummy cell's purpose is solely to redefine the minimum lake stage without changing any other properties of the lake. With the dummy node, when the lake goes dry, no flow from the lake to the groundwater will be allowed. The lake will reform if any lake inflow (such as precipitation, runoff, stream inflow, or drain-like flow to the lake) occurs.

3.7 Increasing Discretization of Lake Cells

Generally speaking, the Lake Package's stage solvers will be more effective when the modeled lakes are represented by many cells, and the lakebed *TOP* elevations vary from cell to cell such than a small change in stage will not cause an overly large change in wetted area. If a modeled lake covers only a few MODFLOW cells (Figure 7a), the modeler may consider dividing the MODFLOW cell into many lake cells to better capture the smaller scale variations in the *TOP* elevation and to improve stage-solver stability (Figure 7b). Note that many lake cells will have the same specified layer, row, column coordinates in this formulation (which is allowed by the MODFLOW code), and will have the same value of groundwater head in leakage calculations.

Also, for side-node connections from an incising lake to a tall model cell (Figure 7c), the modeler can improve the stage-conductance function by similarly subdividing the side connection into several lake cells, each with a different *TOP* elevation (Figure 7d). As the stage in the lake and/or the head in the adjacent cell rises, more lake cells actively contribute to flow. If both the lake stage and the groundwater head are below *TOP* then there is no flow for that lake cell.

Of course, rediscretizing the MODFLOW model grid into smaller cells will also lead to more lake cells and a more accurate description of lakes. Increasing the MODFLOW discretization will also provide for greater resolution in the groundwater head solution.

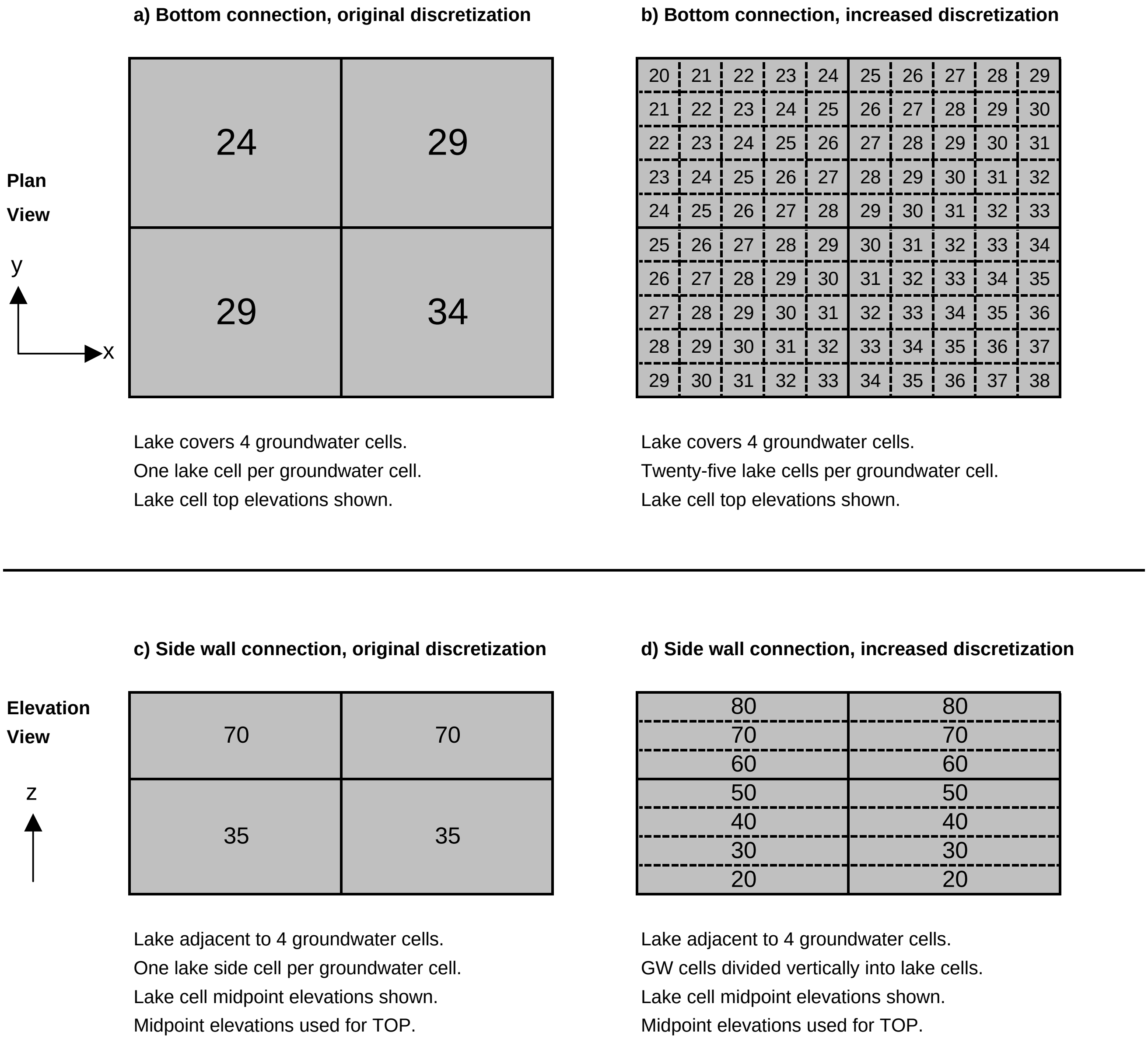


Figure 7. Examples of increasing the discretization for lake cells.

4. Code Design

4.1 Input

The Lake Package is activated in the same manner as other MODFLOW packages, either by specification of a positive value for the appropriate element of the *IUNIT* array in the Basic Package (older MODFLOW versions), or by specifying an input file as type "LAK" in the MODFLOW name file (in MODFLOW-96). See Appendix C for instructions on file opening and *IUNIT* specification in MODFLOW.

The Lake Package input file contains basic data to describe the simulation methods, physical data for each lake for specification of stream-lake connections and lake nodes, and stress period data to define data (such as precipitation, evaporation, and runoff rates) that vary each stress period. The format of the Lake Package input file is described in detail in Appendix B.

4.2 Output

The Lake Package may send output to (up to) three different files: the main (printed, or list) MODFLOW output file, a cell-by-cell flow file, and a separate Lake Package ASCII (stage) output file. The type and frequency of output is controlled largely by values set by the user in the Lake Package and Output Control input files.

As the Lake Package input values are read, the data can be echoed back to the main output file. If the input variable *IECHO* is positive, all input data and summary information (such as the total area of a lake, the total conductance, and the computed minimum stage) will be printed in the main output file for checking. If *IECHO* is zero, only the summary information will be printed (data for individual lake cells will not be printed). Specifying a negative value for *IECHO* suppresses echoing of input data to the main output file.

The variable *ILKCBC*, if positive, specifies a unit number for (binary) saving of cell-by-cell fluxes to the groundwater. The cell-by-cell fluxes are saved using the MODFLOW utility subroutine *UBUDSV* whenever the Output Control file calls for cell-by-cell saving. Note that *UBUDSV* saves a lake-to-groundwater flux for every cell in the MODFLOW model, even though many of the cells may not be connected to lakes and will therefore have a flux of zero. Positive values in the cell-by-cell flux output file indicate flow from the lake to the groundwater (consistent with other MODFLOW boundary condition packages). Unlike the convention of other boundary condition packages, a negative value for *ILKCBC* does not activate the printing of cell-by-cell printing to the main output file. Instead, the *IOUTOP* variable (discussed below), which can be different for each lake, controls printing of cell-by-cell fluxes to the output file.

The input variable *ILKOUT*, if positive, specifies a unit number for the Lake Package ASCII output file (also called the stage output file). This file may contain stage vs. time and lake inflow/outflow components vs. time depending on values set with the *IOUTOP* variable. A header line in the output file defines the data fields (columns) to be saved, which are (in order) lake name, simulation time, stage (prior to transient solution updating), precipitation inflow, evaporation outflow, runoff/withdrawal flow, stream inflow, stream outflow, net groundwater seepage, wetted area, volume, and stage (after transient solution updating). In the stage output file, positive flow values always indicate flow into the lake (positive seepage is from groundwater to lake). If *IOUTOP* is set so that only stage is saved, then only the first three columns are used. The columnar format of the stage output file allows the data to be easily imported into a spreadsheet or database.

A value for *IOUTOP* is specified by the user for each lake in each stress period. At the end of each time step in the period, the current value of *IOUTOP* determines what type of output the Lake Package will send to the main output file and the stage output file. The user can specify a value of 1 to print cell-by-cell fluxes to the output file, 2 to print a summary of the lake volumetric budget in the main output file, 4 to save the lake stage in the stage output file, or 8 to save both flows and stage in the stage output file. Multiple options can be activated by summing the values for the desired individual options. For instance, a value of 11 (1+2+8) calls for cell-by-cell printing, a volumetric budget summary, and printing flows and stage to the stage file.

As with other MODFLOW boundary package files, the Lake Package creates an entry in the MODFLOW aquifer budget to report the total aquifer inflow and outflow for all lakes, and for summing the total volume of water that flows into or out of the aquifer via the lakes. As with other terms in the aquifer budget, the inflows refer to flows into the aquifer (outflow from the lakes). The flows are labeled “lake leakage” in the aquifer budget.

4.3 Lake Package Arrays, and Memory requirements

Six arrays (*ILAKE*, *RLAKE*, *RLNODE*, *ISTROUT*, *RATEQ*, and *ISTRIN*) require space in the main MODFLOW *X* array. During the initial allocation subroutine (*LAK2AL*), these arrays are located within the *X* array for use in the other routines of the Lake Package. The allocation routine reads in the number of lakes (*NLAKES*) and skims the Lake Package input file to determine the total number of lake cells (*MXNODE*), the total number of stream inflows (*MXSTRIN*) and outflows (*MXSTROUT*), and the total number of rating equation components (*MXRATEQ*). The procedure then sets aside 9 elements of *X* for each lake for the *ILAKE* integer array, 18 elements of *X* for each lake for the *RLAKE* double-precision array, 5 elements of *X* for each lake cell for the *RLNODE* real array, 3 elements of *X* for each stream outflow for the *ISTROUT* integer array, 4 elements of *X* for each rating equation component for the *RATEQ* real array, and 1 element of *X* for each stream inflow for the *ISTRIN* integer array. The data stored in these arrays are described in Table 1. The total number of (4-byte) elements of *X* required for a Lake Package simulation is then:

Table 1. Arrays used by the Lake Package and the information they contain.

Array Name & Dimensions	Pos.	Local Variable Name	Description
ILAKE (9,NLAKES)	(1,K)	NODES	Number of nodes for lake K
	(2,K)	NSTRIN	Number of stream inflows to lake K
	(3,K)	NSTROUT	Number of stream outflows fro lake K
	(4,K)	LRLNODE	Position in RLNODE of first cell in lake K
	(5,K)	LISTRIN	Position in ISTRIN of first stream inflow to lake K
	(6,K)	LISTROUT	Position in ISTROUT of first stream outflow for lake K
	(7,K)	ISIMMODE	Simulation mode for lake K
	(8,K)	ITERLAKE	Maximum stage solver iterations for lake K
	(9,K)	IOUTOP	Output option flag for lake K
RLAKE (9,NLAKES) *double-precision*	(1,K)	STAGE	Current lake stage, lake K
	(2,K)	STAGEMX	Maximum stage of lake K
	(3,K)	STAGEMN	Minimum stage of lake K
	(4,K)	PRECIP	Total-area-dependant inflow/outflow for lake K
	(5,K)	EVAP	Wetted-area-dependant inflow/outflow for lake K
	(6,K)	RUNOFF	Direct inflow/outflow for lake K
	(7,K)	DRYRCH	Shore cell recharge rate for lake K
	(8,K)	CONVCRIT/DSDT	Stage solver convergence criterion or stage rate of change for lake K
	(9,K)	TOTAREA	Total (maximum) area of lake K
RLNODE (5,MXNODE)	(1,L)	POS	absolute position of cell L
	(2,L)	TOP	lakebed to elevation for cell L
	(3,L)	BOT	lakebed bottom elevation for cell L
	(4,L)	AREA	horizontal area of cell L
	(5,L)	COND	lakebed conductance for cell L
ISTROUT (3,MXSTROUT)	(1,J)	ISEG	Stream segment of lake outflow J
	(2,J)	NRATEQ	Number of rating equations used to describe flow vs. stage for outflow J
	(3,J)	LRATEQ	Position in RATEQ of the first rating equation component for outflow J
RATEQ (4,MXRATEQ)	(1,M)	CUTOFF	Minimum stage for application of rating equation component M
	(2,M)	CONST	Multiplicative constant in rating equation component M
	(3,M)	ELEV	Reference elevation in rating equation component M
	(4,M)	EXPNT	Exponential constant in rating equation component M
ISTRIN(MXSTRIN)	(N)	ISEG	Stream segment of lake inflow N

27	x	Number of Lakes	
+	5	x	Number of Lake cells
+	3	x	Number of Stream Outflows
+	5	x	Number of Rating Equation Components
+			Number of Stream Inflows
=			Number of Elements in *X* Required

Additionally, a character array, *LAKENAME*, is used to keep track of the names for each simulated lake. Each name is a 10-character string. The number of elements required in this array is equal to the number of lakes in the simulation.

4.4 Procedure Descriptions & Flow Charts

This section describes the inner workings of the Lake Package code, practically line by line. There are 13 subroutines and 3 functions that make up the Lake Package. These 16 FORTRAN procedures are described in detail in the next 16 subsections. Each section includes a step-by-step narrative of the code calculations, a chart that shows procedural flow, and a table of variables used in the procedure.

Six of the Lake Package subroutines are called by the main MODFLOW routine (Figure 8). The other 10 procedures are sub-modules that are called by other Lake Package procedures. One MODFLOW utility subroutine, *UBUDSV*, is called by the Lake Package to save cell-by-cell fluxes from lakes to the aquifer.

4.4.1 Subroutine *LAK2AL*

The *LAK2AL* subroutine reads and scans the input file and sets up the arrays to be stored in *X*. In this procedure, much of the Lake Package input file is scanned so that the total number of lake cells, stream inflows, stream outflows, and rating equation components can be determined. The input file is then repositioned at the third line for processing by the *LAK2RP1* procedure. This subroutine is called once, near the beginning of a MODFLOW simulation. A flow chart for this procedure is presented in Figure 9, and a list of variables used is presented in Table 2. The steps of this procedure are as follows:

1) Write a line to the output file that identifies the Lake Package and the input file unit number

2) Read the first line in the input file. If that line does not begin with “/*LAK2.2”, then the input file is not a valid Lake Package input file — stop. Otherwise continue and read the second line of simulation parameters (*NLAKES*, *ILKCBC*, *ILKOUT*, *IECHO*, *NSUBSTEPS*)

Table 2. List of Variables in Subroutine LAK2AL.

Variable Name	Scope	Description
IECHO	Package	Level of input echoing: <0 no input echoing, 0 summary of input, >0 full echoing of input
ILKCBC	Package	Lake cell-by-cell flux flag and unit number: >0 unit number for cell-by-cell fluxes (saved whenever the ICBCFL variable of the Output Control package is non-zero), ≤0 do not save cell-by-cell fluxes
ILKOUT	Package	Lake stage/budget output flag and unit number: >0 stage/budget unit number (ASCII output recorded depending on the value of IOUTOP in each stress period), ≤0 Do not write stage/budget records
IN	Package	Unit number of Lake Package input file
IOUT	Global	Unit number of main (list) output file
ISOPEN	Procedure	Logical flag indicating whether or not a unit number is connected to an open file
ISP	Procedure	Number of elements (words) in the X array needed for the various subarrays
ISPT	Procedure	Total number of elements (words) in the X array needed for the Lake Package
ISUM	Global	Index number of the lowest element in the X array that is not yet allocated
K	Procedure	Counter for lakes
K1	Procedure	Counter for stream inflows, stream outflows, and lake cells
K2	Procedure	Counter for rating equation components
LAKID	Procedure	Eight-character variable set to first line of Lake Package input file; must evaluate to “/*LAK2.2"
LCILAKE	Package	Location in the X array of the first element in the ICLAKE array
LCISTRIN	Package	Location in the X array of the first element in the ISTRIN array
LCISTROUT	Package	Location in the X array of the first element in the ISTROUT array

Table 2. List of Variables in Subroutine LAK2AL (continued).

Variable Name	Scope	Description
LCRATEQ	Package	Location in the X array of the first element in the RATEQ array
LCRLAKE	Package	Location in the X array of the first element in the RLAKE array
LCRLNODE	Package	Location in the X array of the first element in the RLNODE array
MXNODE	Package	Total number of lake cells (all lakes)
MXRATEQ	Package	Total number of rating equation components (all stream outflows, all lakes)
MXSTRIN	Package	Total number of stream inflows (all lakes)
MXSTROUT	Package	Total number of stream outflows (all lakes)
NLAKES	Package	Number of lakes
NODES	Procedure	Number of lake cells in current lake
NRATEQ	Procedure	Number of rating equation components for current stream outflow
NSTRIN	Procedure	Number of stream inflows for current lake
NSTROUT	Procedure	Number of stream outflows for current lake
NSUBSTEPS	Package	Number of sub-time steps used for transient stage simulation

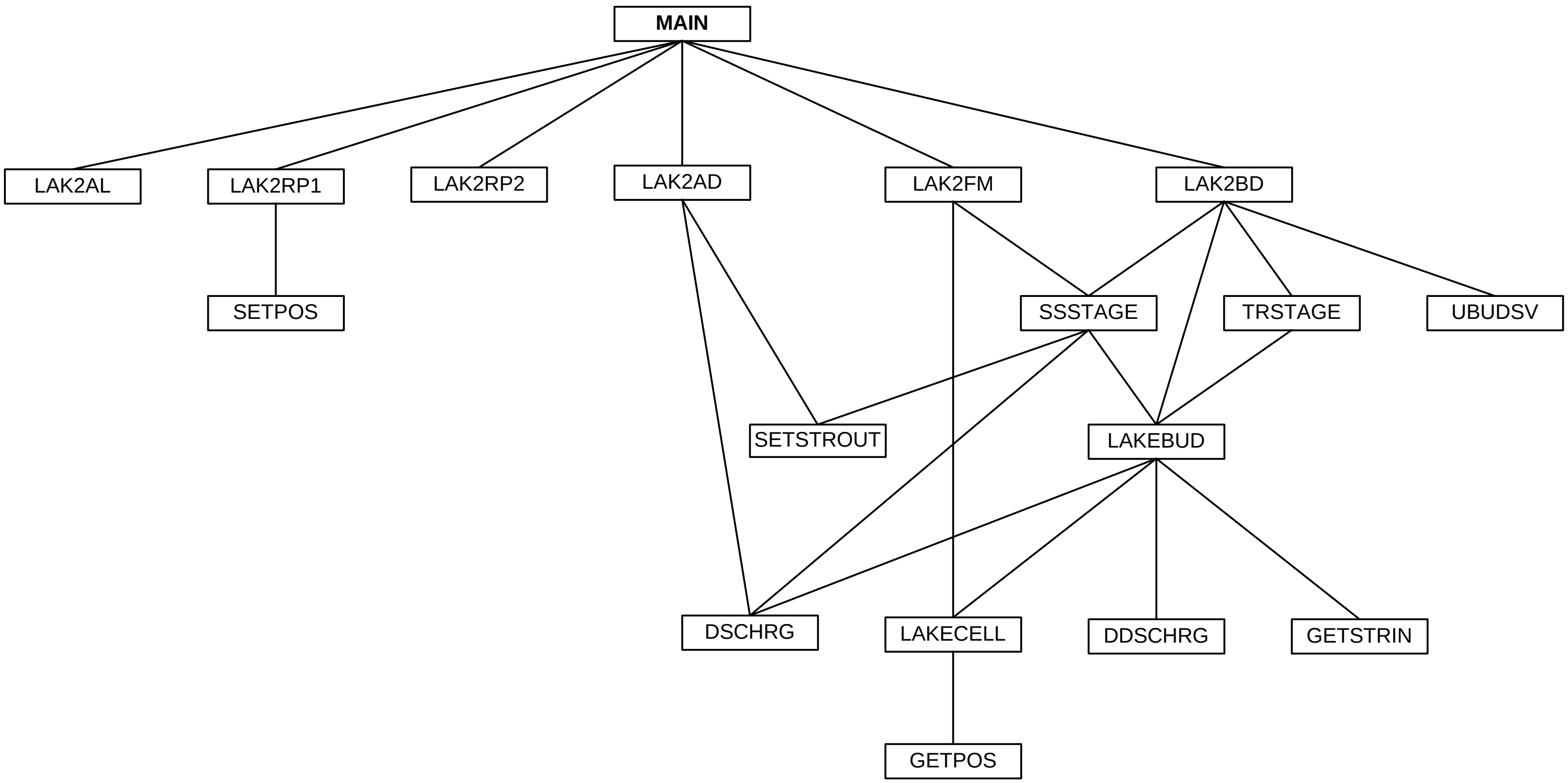


Figure 8. Relationship Between Procedures in the Lake Package

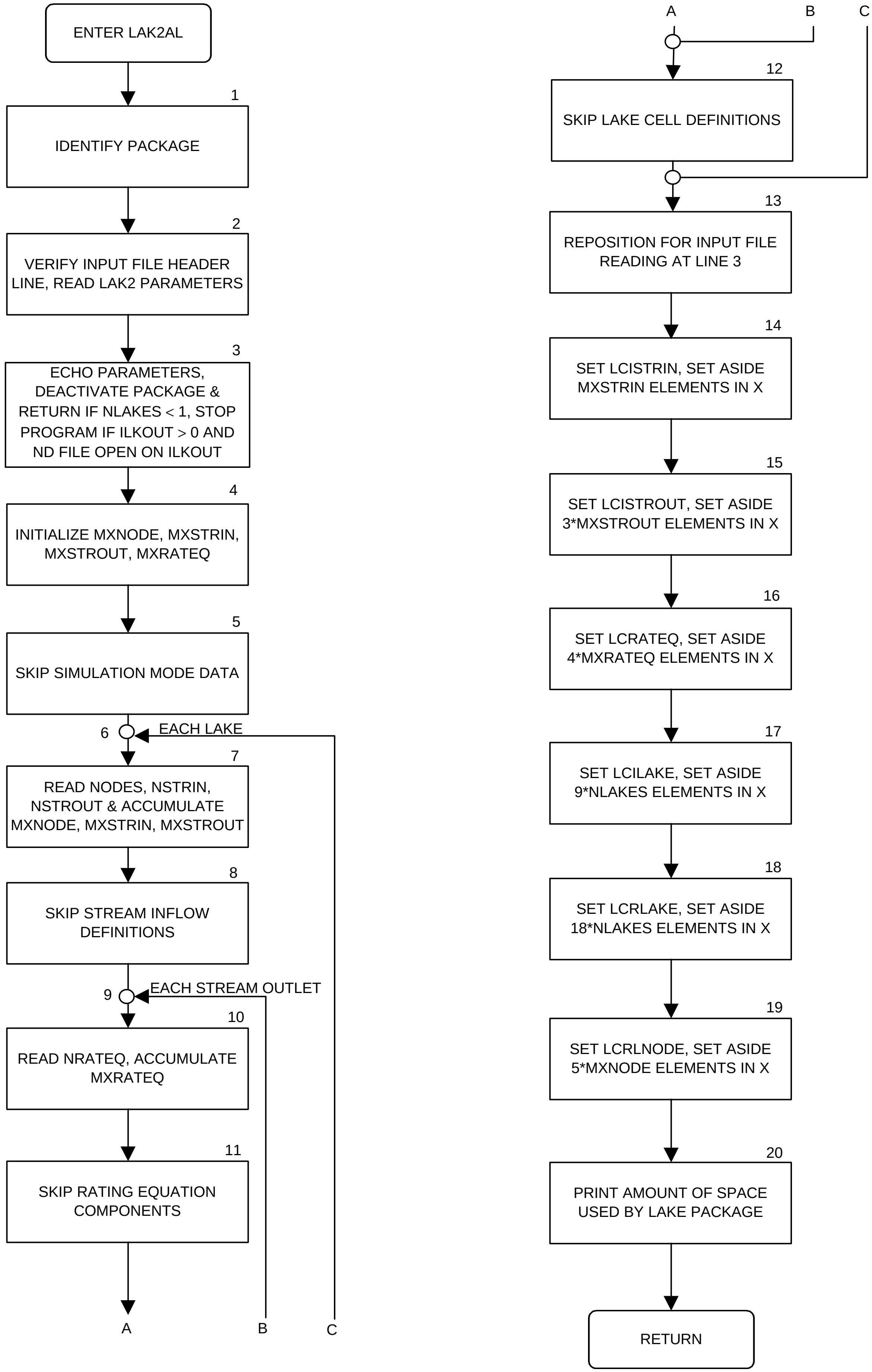


Figure 9. Flowchart for Subroutine LAK2AL

3) Echo the number of lakes to the output file. If the number of lakes is less than 1, return. If *ILKCBC* is positive, echo a message identifying the unit number for cell-by-cell lake fluxes, else echo a message that cell-by-cell lake fluxes will not be saved. If the variable *ILKOUT* is positive, echo a message identifying the unit number for the stage output file and make sure that a file is open on that unit (stop if not). Based on *IECHO* print a message regarding the level of input echoing. If *NSUBSTEPS* is less than 1, set it to 1. Echo the number of sub-time-steps.

4) Initialize the total number of lake cells (*MXNODE*), stream inflows (*MXSTRIN*), stream outflows (*MXSTROUT*), and rating equation components (*MXRATEQ*) to zero.

5) Skip a line in the input file for each lake (the lines that identify the names, simulation modes, etc. for each lake.

6) Repeat steps 7 through 12 for each lake to accumulate the total number of lake cells, stream inflows and outflows, and rating equation components

7) Read the number of lake cells (*NODES*), stream inflows (*NSTRIN*), and stream outflows (*NSTROUT*) for the lake. Add these values to *MXNODE*, *MXSTRIN*, and *MXSTROUT,* respectively.

8) Skip a line in the input file for each stream inflow.

9) For each stream outflow, repeat steps 10 and 11 to accumulate the total number of rating equation components

10) Read the number of rating equation components for this stream outflow and add to *MXRATEQ*.

11) Skip a line in the input file for each rating equation component.

12) Skip a line in the input file for each lake cell.

13) Rewind the input file and skip the first two lines to position for *LAK2RP1*.

14) Set *LCISTRIN*, the *X* array address for *ISTRIN* to the first available position in *X* (*ISUM*). The number of elements in *X* used by this array is *MXSTRIN*. Add this value to the total space used by the Lake Package (*ISPT*) and add this value to *ISUM*.

15) Set *LCISTROUT*, the *X* array address for *ISTROUT* to the first available position in *X* (*ISUM*). The number of elements in *X* used by this array is (*MXSTROUT* x 3). Add this value to the total space used by the Lake Package (*ISPT*) and add this value to *ISUM*.

16) Set *LCRATEQ*, the *X* array address for *RATEQ* to the first available position in *X* (*ISUM*). The number of elements in *X* used by this array is (*MXRATEQ* x 4). Add this value to the total space used by the Lake Package (*ISPT*) and add this value to *ISUM*.

17) Set *LCILAKE*, the *X* array address for *ILAKE* to the first available position in *X* (*ISUM*). The number of elements in *X* used by this array is (*NLAKES* x 9). Add this value to the total space used by the Lake Package (*ISPT*) and add this value to *ISUM*.

18) Set *LCRLAKE*, the *X* array address for *RLAKE* to the first available position in *X* (*ISUM*). The number of elements in *X* used by this array is (*NLAKES* x 9 elements x 2 for double-precision). Add this value to the total space used by the Lake Package (*ISPT*) and add this value to *ISUM*.

19) Set *LCRLNODE*, the *X* array address for *RLNODE* to the first available position in *X* (*ISUM*). The number of elements in *X* used by this array is (*NODES* x 5). Add this value to the total space used by the Lake Package (*ISPT*) and add this value to *ISUM*.

20) To avoid zero-dimension array error messages for unused arrays, check the values of *MXSTRIN*, *MXSTROUT*, and *MXRATEQ*, and for any that are zero, reset to 1. Print a message in the output file listing the number of *X* elements required for the Lake Package, and the total elements required by all packages so far. If the total so far is greater than the total length of *X*, print a warning message to the output file.

4.4.2 Subroutine *LAK2RP1*

The *LAK2RP1* subroutine reads the simulation mode and physical data for each lake (the data that does not vary by stress period) starting on line 3 of the Lake Package input file. This subroutine is called once by the main MODFLOW routine, after *X* array allocation but before the stress period loop. A flow chart for this procedure is presented in Figure 10, and a list of variables used is presented in Table 3. The steps of this procedure are as follows:

1) Initialize the array locator variables *LRLNODE*, *LISTRIN*, *ILISTROUT*, and *LRATEQ* to 1.

2) For each lake, read the simulation data line (*LNAME*, *ISIMMODE*, *STAGE*, *ITERLAKE*, and *CONVCRIT*). Store *LNAME* in the *LAKENAME* array, and store the other variables in the appropriate positions in *ILAKE* and *RLAKE*. If full or summary echoing (*IECHO* nonnegative), print the simulation mode and the pertinent simulation data to the main output file.

3) For each lake, repeat the remaining steps to read and store all lake physical data.

Table 3. List of Variables in Subroutine LAK2RP1.

Variable Name	Scope	Description
AREA	Procedure	Horizontal area of lakebed in current lake cell
BOT	Procedure	Bottom elevation of lakebed at current lake cell
COND	Procedure	Hydraulic conductivity/conductance of lakebed at current lake cell
CONST	Procedure	Multiplicative constant for rating equation component
CONVCRIT	Procedure	Convergence criterion for stage solution (double precision)
CUTOFF	Procedure	Cutoff elevation for rating equation component
ELEV	Procedure	Reference elevation for rating equation component
EXPNT	Procedure	Exponential constant for rating equation component
ICOL	Procedure	Column number of model cell connected to lake
ICONDOP	Package	Conductance option: 1 hydraulic conductances are input, <>1 hydraulic conductivities are input
IECHO	Package	Level of input echoing: <0 no input echoing, 0 summary of input, >0 full echoing of input
ILAKE	Package	Integer array of lake parameters, dimension (9,NLAKES)
ILAY	Procedure	Layer number of model cell connected to lake
IN	Package	Unit number of Lake Package input file
IOUT	Global	Unit number of main (list) output file
IROW	Procedure	Row number of model cell connected to lake
ISEG	Procedure	Stream segment number for stream inflow or outflow
ISIMMODE	Procedure	Input simulation mode for a lake: 0 fixed-constant, 1 fixed-interpolated, 2 steady-state, 3 transient
ISTRIN	Package	Array of inflow stream segment numbers, dimension (MXSTRIN)
ISTROUT	Package	Array of outflow stream parameters, dimension (3,MXSTROUT)
ITERLAKE	Package	Maximum iterations allowed for stage solution

Table 3. List of Variables in Subroutine LAK2RP1 (continued).

Variable Name	Scope	Description
K	Procedure	Counter for lakes
K1	Procedure	Counter for stream inflows, stream outflows, and lake cells
K2	Procedure	Counter for rating equation components
LAKENAME	Package	Array of lake names, 10-character string, dimension (NLAKES)
LISTRIN	Procedure	Location in the ISTRIN array of the first stream inflow segment for the current lake
LISTROUT	Procedure	Location in the ISTROUT array of the first stream outflow definition for the current lake
LNAME	Procedure	Current lake name, 10-character string
LRATEQ	Procedure	Location in the RATEQ array of the first rating equation component definition for the current stream outflow
LRLNODE	Procedure	Location in the RLNODE array of the first node definition for the current lake
MXNODE	Package	Total number of lake cells (all lakes)
MXRATEQ	Package	Total number of rating equation components (all stream outflows, all lakes)
MXSTRIN	Package	Total number of stream inflows (all lakes)
MXSTROUT	Package	Total number of stream outflows (all lakes)
NCOL	Global	Number of columns in model grid
NLAKES	Package	Number of lakes
NLAY	Global	Number of layers in model grid
NODES	Procedure	Number of lake cells in current lake
NRATEQ	Procedure	Number of rating equation components for current stream outflow
NROW	Global	Number of rows in model grid
NSTRIN	Procedure	Number of stream inflows for current lake
NSTROUT	Procedure	Number of stream outflows for current lake

Table 3. List of Variables in Subroutine LAK2RP1 (continued).

Variable Name	Scope	Description
RATEQ	Package	Array of rating equation parameters, dimension (4,MXRATEQ)
RLAKE	Package	Real (double precision) array of lake parameters, dimension (9,NLAKES)
RLNODE	Package	Array of lake node parameters, dimension (5,MXNODE)
STAGE	Procedure	Lake stage (double precision)
STAGEMN	Procedure	Minimum lake stage (double precision)
STAGEMX	Procedure	Maximum lake stage (double precision)
TOP	Procedure	Top elevation of lakebed at current lake cell
TOTAREA	Procedure	Total (maximum) lake area (double precision)
TOTCOND	Procedure	Total (maximum) lake conductance

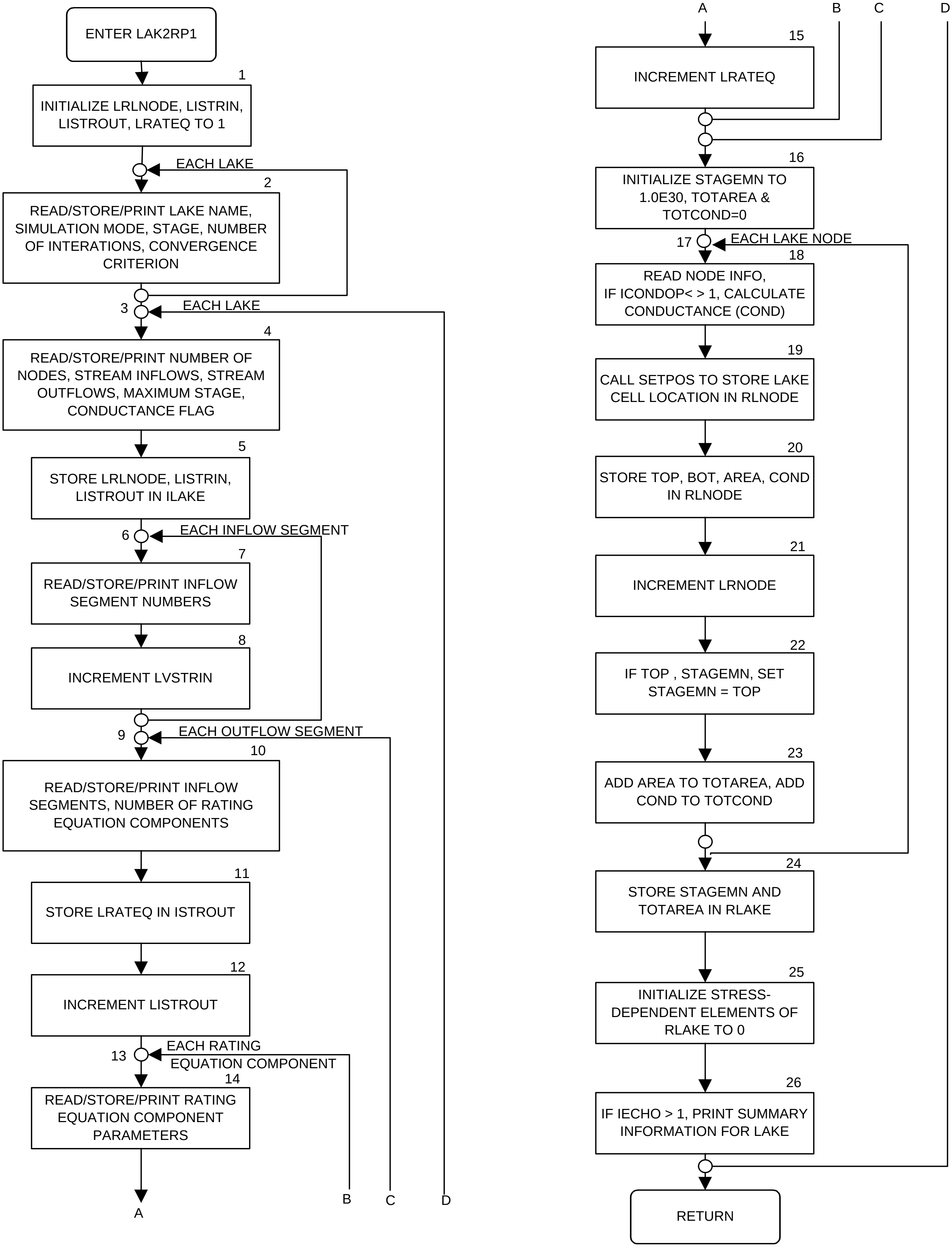


Figure 10. Flowchart for Subroutine LAK2RP1

4) Read the lake dimensions and parameters: *NODES*, *NSTRIN*, *NSTROUT*, *STAGEMX*, *ICONDOP*. Store these values in the appropriate positions in *ILAKE* and *RLAKE*.

5) Store current value of *LRLNODE* in *ILAKE* as location in *RLNODE* of the first lake cell definition for this lake. Store current value of *LISTRIN* in *ILAKE* as location in *ISTRIN* of the first stream inflow definition for this lake. Store current value of *LSTROUT* in *ILAKE* as location in *STROUT* of the first stream outflow definition for this lake.

6) Repeat steps 7 and 8 for each stream inflow for this lake to read and set the stream inflow segment information.

7) Read the stream inflow segment, store in *ISTRIN*, and echo the data to the output file (if full echoing enabled).

8) Increment *LISTRIN* by 1.

9) Repeat steps 10 through 15 for each stream outflow for this lake to read and set stream outflow information.

10) Read the segment number and number of rating equation components for this stream outflow. Store in *ISTROUT*, and print to output file (if full echoing)

11) Store current value of *LRATEQ* in *ISTROUT* as location in *RATEQ* of the first rating equation component for this stream outflow.

12) Increment *LISTROUT* by 1.

13) Repeat steps 14 and 15 for each rating equation component for this stream outflow to read and store the rating equation parameters

14) Read the 4 rating equation parameters (*CUTOFF*, *CONST*, *ELEV*, and *EXPNT*) and store in *RATEQ*. Echo to the output file (if full echoing)

15) Increment *LRATEQ* by 1.

16) Initialize the lake’s minimum stage (*STAGEMN*) to 1.0E30, an arbitrarily high number. Initialize the lake’s total area (*TOTAREA*) and total conductance (*TOTCOND*) to zero.

17) For each cell in the lake, repeat steps 18-23 to process information on each lake cell.

18) Read layer, row, column, *TOP*, *BOT*, *AREA*, and *COND* for this cell. If *COND* represents a hydraulic conductivity, multiply it by *AREA* and divide by (*TOP* - *BOT*) to compute conductance. Echo the lake cell information (if full echoing).

19) Call *SETPOS* to store in the first position of *RLNODE* the absolute position of the lake cell.

20) Also store *TOP*, *BOT*, *AREA*, and conductance in *RLNODE*.

21) Increment *LRLNODE* by 1.

22) If this is the lowest *TOP* elevation so far for this lake, reset *STAGEMN* to *TOP*.

23) Add the cell *AREA* and conductance to *TOTAREA* and *TOTCOND*, respectively.

24) Store the final values of *STAGEMN* and *TOTCOND* in *RLAKE*.

25) Initialize the stress-dependent elements of *RLAKE* to zero.

26) If full or summary echoing is enabled, print summary information for this lake: the number of cells, inflow and outflow streams, minimum and maximum stage, total area, and total conductance.

4.4.3 Subroutine *LAK2RP2*

The *LAK2RP2* subroutine reads the part of the Lake Package input file that varies by stress period. This subroutine is called once each stress period near the beginning of the stress period loop. A flow chart for this procedure is presented in Figure 11, and a list of variables used is presented in Table 4. The steps of this procedure are as follows:

1) Read the stress period flag, *ITMP*.

2) Check *ITMP*. If negative, proceed to step 3, otherwise go to step 4.

3) Print a message that data from the previous stress period will be used for all lakes. For any lake that is in interpolated-stage mode (*ISIMMODE* = 1), set the rate of change in stage (8^{th} position of *RLAKE*) to zero. Return.

4) For each lake, repeat the remaining steps to process the stress period data.

5) Read the stress period data (*PRECIP*, *EVAP*, *RUNOFF*, *DRYRCH*, *IOUTOP*, and *STAGE*). Store in appropriate positions in *RLAKE* and *ILAKE*.

6) If the simulation mode is 1 (interpolated stage) then print the stress rates and the final stage. Calculate the stage rate of change for the stress period and store in the 8^{th} position of *RLAKE*. Skip steps 7 and 8.

Table 4. List of Variables in Subroutine LAK2RP2.

Variable Name	Scope	Description
DRYRCH	Procedure	Recharge rate for shore cells (double precision)
DSDT	Procedure	Time rate of change in stage (double precision)
EVAP	Procedure	Evaporation (wetted-area dependent) rate (double precision)
ILAKE	Package	Integer array of lake parameters, dimension (9, NLAKES)
IN	Package	Unit number of Lake Package input file
IOUT	Global	Unit number of main (list) output file
IOUTOP	Procedure	Flag indicating type of output desired
ITMP	Procedure	Flag indicating whether stresses will be read (>0) or re-used (≤0)
K	Procedure	Counter for lakes
LAKENAME	Global	Array of lake names, 10-character string, dimension (NLAKES)
NLAKES	Package	Number of lakes
PRECIP	Procedure	Precipitation (total-area dependent) rate (double precision)
RLAKE	Package	Real (double precision) array of lake parameters, dimension (9,NLAKES)
RUNOFF	Procedure	Runoff (withdrawal) flow rate (double precision)
STAGE	Procedure	Lake stage (double precision)

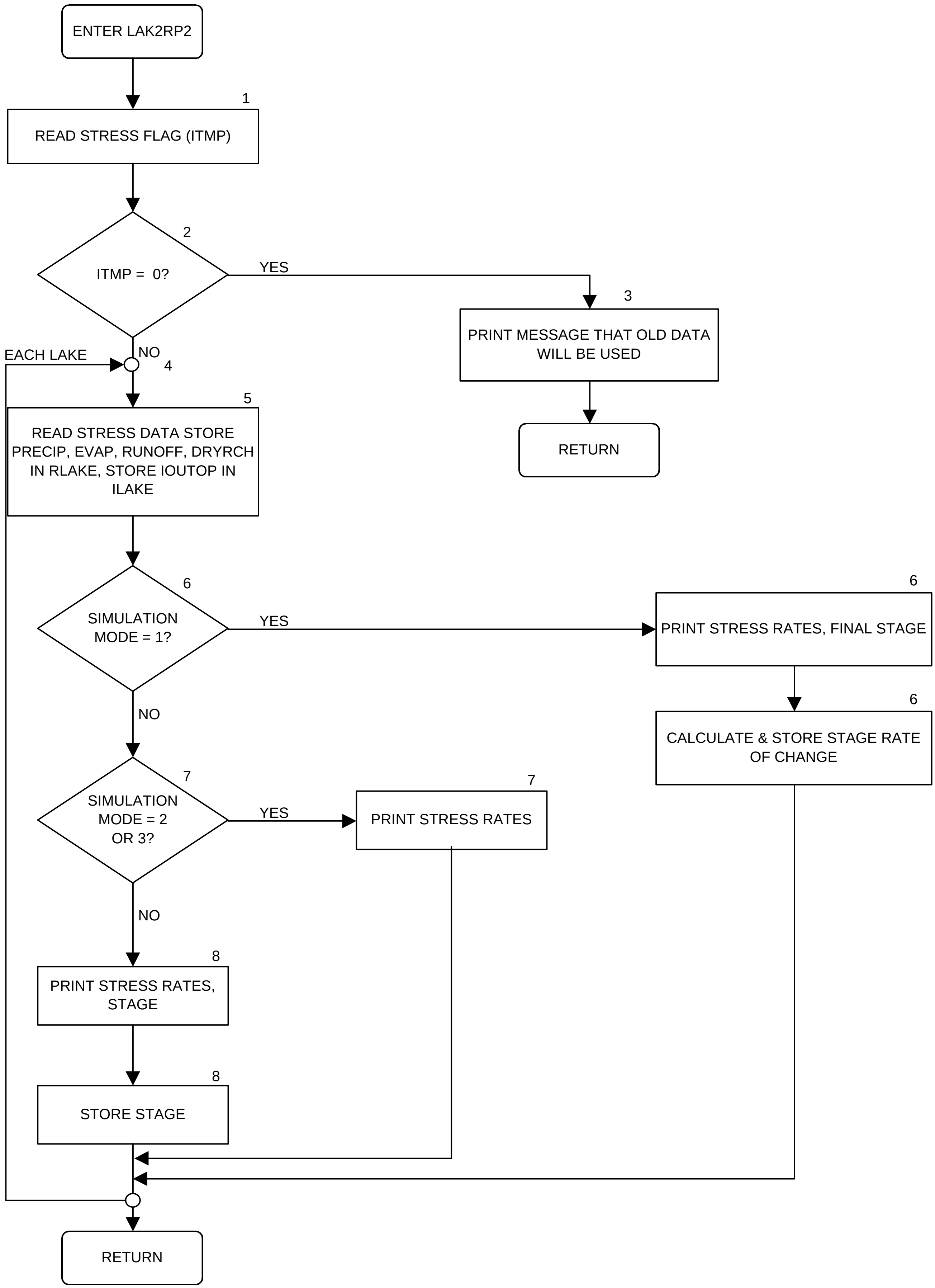


Figure 11. Flowchart for Subroutine LAK2RP2

7) If the simulation mode is 2 or 3 (steady-state or transient stage solver) then print the stress rates (*STAGE* is ignored). Skip step 8.

8) The simulation mode is constant stage. Print the stress rates and *STAGE*. Store *STAGE* in the 1st position of *RLAKE*.

4.4.4 Subroutine *LAK2AD*

The *LAK2AD* subroutine is called near the beginning of each time step (before the iteration loop) to set the lake stage for the time step for lakes in interpolated-stage mode and to set the lake-stream outflows for the Stream Package. A flow chart for this procedure is presented in Figure 12, and a list of variables used is presented in Table 5. The steps of this procedure are as follows:

1) For each lake, perform all remaining steps.

2) If the simulation mode is 1 (interpolated stage) multiply the stage rate of change by the current time step and add to the previous stage. Store this new stage value in *RLAKE*.

3) Set *NSTROUT*, *LSTROUT*, and *STAGE* from *ILAKE* and *RLAKE*.

4) Repeat the remaining steps for each stream outflow.

5) Set *ISEGOUT*, *NRATEQ*, and *LRATEQ* from *ISTROUT* using *LISTROUT* to address.

6) Increment *LISTROUT* by 1.

7) Use function *DSCHRGE* and the current set of rating equations (beginning with component *LRATEQ*) to determine the stream outflow rate (*QSTROUT*) at the current stage.

8) Call subroutine *SETSTROUT* to apply *QSTROUT* as the inflow to the first reach of the outflow segment *ISEGOUT*.

4.4.5 Subroutine *LAK2FM*

The *LAK2FM* subroutine formulates the lake boundary condition for the MODFLOW groundwater equations. For steady-state mode, *SSSTAGE* is called. This subroutine is called each MODFLOW iteration. A flow chart for this procedure is presented in Figure 13, and a list of variables used is presented in Table 6. The steps of this procedure are as follows:

1) For each modeled lake, repeat the remaining steps.

Table 5. List of Variables in Subroutine LAK2AD.

Variable Name	Scope	Description
DELT	Global	Length of the current time step
ILAKE	Package	Integer array of lake parameters, dimension (9,NLAKES)
ISEGOUT	Procedure	Stream outflow segment number
ISTRM	Stream	For each stream reach: layer, row, column, segment number, reach number, dimension (5,NSTREM)
ISTROUT	Package	Array of outflow stream parameters, dimension (3,MXSTROUT)
K	Procedure	Counter for lakes
K1	Procedure	Counter for stream outflows
LISTROUT	Procedure	Location in the ISTROUT array of the first stream outflow definition for the current lake
LRATEQ	Procedure	Location in the RATEQ array of the first rating equation component definition for the current stream outflow
MXRATEQ	Package	Total number of rating equation components (all stream outflows, all lakes)
MXSTROUT	Package	Total number of stream outflows (all lakes)
NLAKES	Package	Number of lakes
NRATEQ	Procedure	Number of rating equation components for current stream outflow
NSTREM	Stream	Number of stream reaches
NSTROUT	Procedure	Number of stream outflows for current lake
QSTROUT	Procedure	Flow rate to an individual stream outflow
RATEQ	Package	Array of rating equation parameters, dimension (4,MXRATEQ)
RLAKE	Package	Real (double precision) array of lake parameters, dimension (9,NLAKES)
STAGE	Procedure	Lake stage (double precision)

Table 5. List of Variables in Subroutine LAK2AD (continued).

Variable Name	Scope	Description
STRM	Stream	For each stream reach: segment inflow, stage, conductance, bottom, top, width, slope, roughness, reach outflow, reach inflow, leakage, dimension (11,NSTREM)

Table 6. List of Variables in Subroutine LAK2FM.

Variable Name	Scope	Description
ARTRIB	Stream	Streamflow out of last reach in each segment, dimension (NSS)
DRYRCH	Procedure	Recharge rate for shore cells (double precision)
DUM	DUM	Procedure
HCOF	Global	Coefficient of head in the finite difference equation, dimension (NCOL,NROW,NLAY)
HEAD	Global	Current aquifer head at each cell (double precision), dimension (NCOL,NROW,NLAY)
IBOUND	Global	Cell status flag, dimension (NCOL,NROW,NLAY)
ILAKE	Package	Integer array of lake parameters, dimension (9,NLAKES)
ISIMMODE	Procedure	Input simulation mode for a lake: 0 fixed-constant, 1 fixed-interpolated, 2 steady-state, 3 transient
ISTRIN	Package	Array of inflow stream segment numbers, dimension (MXSTRIN)
ISTRM	Stream	For each stream reach: layer, row, column, segment number, reach number, dimension (5,NSTREM)
ISTROUT	Package	Array of outflow stream parameters, dimension (3,MXSTROUT)
ITERLAKE	Procedure	Maximum iterations allowed for stage solution
K	Procedure	Counter for lakes
LISTRIN	Procedure	Location in the ISTRIN array of the first stream inflow segment for the current lake
LISTROUT	Procedure	Location in the ISTROUT array of the first stream outflow definition for the current lake
LRLNODE	Procedure	Location in the RLNODE array of the first node definition for the current lake
MXNODE	Package	Total number of lake cells (all lakes)
MXRATEQ	Package	Total number of rating equation components (all stream outflows, all lakes)

Table 6. List of Variables in Subroutine LAK2FM (continued).

Variable Name	Scope	Description
MXSTRIN	Package	Total number of stream inflows (all lakes)
MXSTROUT	Package	Total number of stream outflows (all lakes)
NCOL	Global	Number of columns in model grid
NLAKES	Package	Number of lakes
NLAY	Global	Number of layers in model grid
NODES	Procedure	Number of lake cells in current lake
NROW	Global	Number of rows in model grid
NSS	Stream	Number of stream segments
NSTREM	Stream	Number of stream reaches
NSTRIN	Procedure	Number of stream inflows for current lake
NSTROUT	Procedure	Number of stream outflows for current lake
RATEQ	Package	Array of rating equation parameters, dimension (4,MXRATEQ)
RHS		
RLAKE	Package	Real (double precision) array of lake parameters, dimension (9,NLAKES)
RLNODE	Package	Array of lake node parameters, dimension (5,MXNODE)
STAGE	Procedure	Lake stage (double precision)
STRM	Stream	For each stream reach: segment inflow, stage, conductance, bottom, top, width, slope, roughness, reach outflow, reach inflow, leakage, dimension (11,NSTREM)

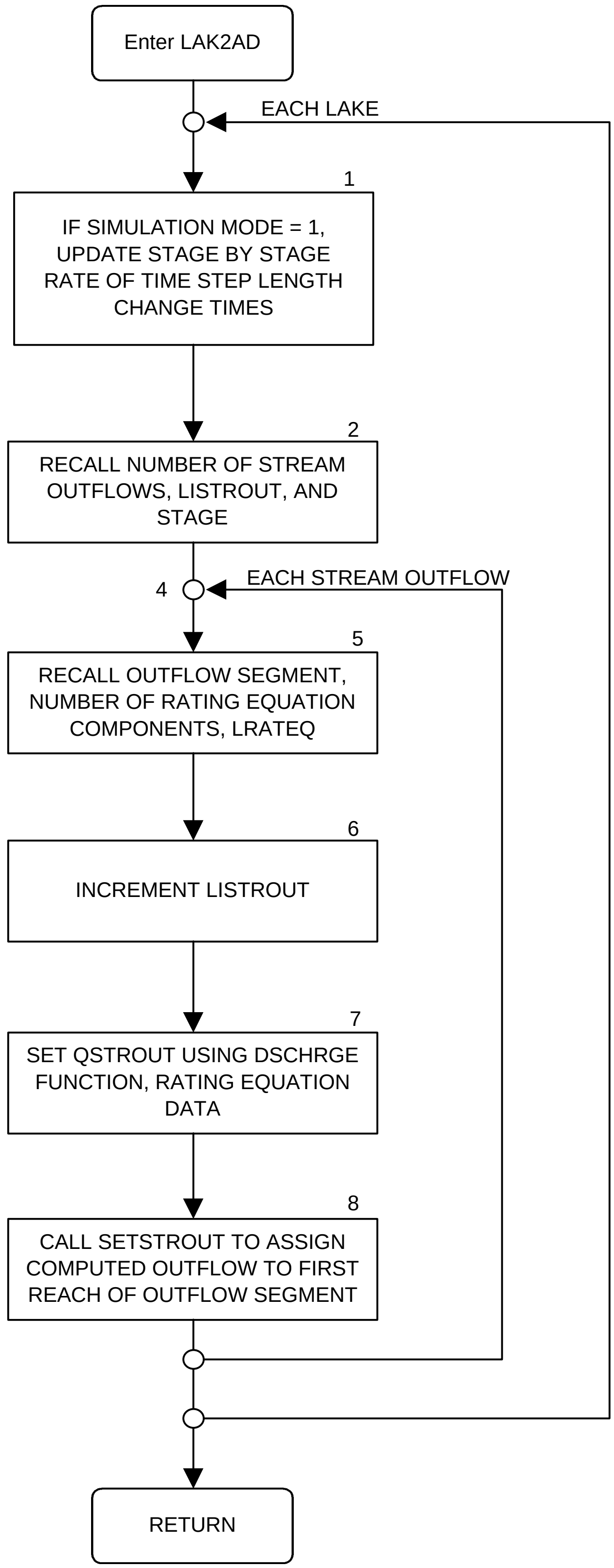


Figure 12. Flowchart for Subroutine LAK2AD

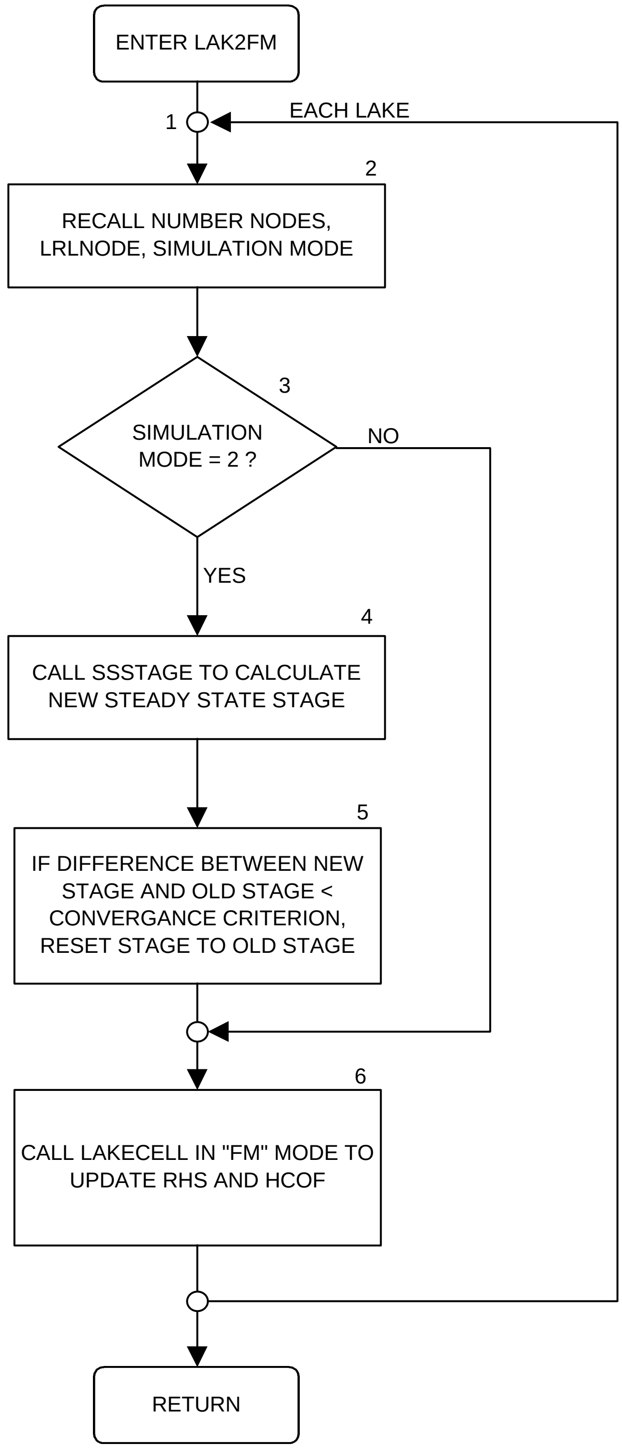


Figure 13. Flowchart for Subroutine LAK2FM

2) Recall the number of lake cells (*NODES*), the address of the first cell in the lake (*LRLNODE*), and the simulation mode (*ISIMMODE*).

3) If the simulation mode is 2 (steady-state stage solver) proceed to step 4. Otherwise, skip to step 6.

4) Call *SSSTAGE* for this lake to update the stage.

5) If the difference between the newly-computed steady-state stage and the previous stage is lower than the convergence criterion (stored in the 8^{th} position of *RLAKE*), then reset the stage to the previous value.

6) Call *LAKECELL* in "FM" mode to update *HCOF* and *RHS* at the individual cells of this lake.

4.4.6 Subroutine *LAK2BD*

The *LAK2BD* subroutine reports the volumetric budget of each modeled lake. MODFLOW calls this subroutine each time step, after the aquifer heads have been determined. A flow chart for this procedure is presented in Figure 14, and a list of variables used is presented in Table 7. The steps of this procedure are as follows:

1) Determine if cell-by-cell flows from the lake to the aquifer will be saved this time step (only if *ICBCFL* non-zero and *ILKCBC* positive). Set the logical flag *CBCSAVE* accordingly. If cell-by-cell flows will be saved, clear the buffer (*BUFF*) to zero. Initialize the sums of aquifer flows, *TOTIN* and *TOTOUT*, to zero.

2) For each simulated lake, repeat steps 3 through 14.

3) Set local variables from *ILAKE*. Add 16 to *IOUTOP* if cell-by-cell flows will be saved.

4) If cell-by-cell terms will be printed to the main output file (*IOUTOP* is odd), print a header line for the lake.

5) If the simulation mode is 2 (steady-state stage solver), call *SSSTAGE* to get the steady-state lake stage.

6) Call *LAKEBUD* in "BD" mode to compute volumetric budget components and cell-by-cell flows.

7) If the simulation mode is 3 (transient stage solver) proceed to step 8. Otherwise skip to step 10.

Table 7. List of Variables in Subroutine LAK2BD.

Variable Name	Scope	Description
ARTRIB	Stream	Streamflow out of last reach in each segment, dimension (NSS)
BUDGET	Procedure	Lake-wide volumetric budget components, dimension (2,5)
BUFF	Global	Buffer used to accumulate cell-by-cell flows for saving to disk, dimension (NCOL,NROW,NLAY)
CBCSAV	Procedure	Logical flag indicating whether cell-by-cell flows will be saved
DELT	Global	Length of the current time step
DT	Procedure	Length of a sub-time step
DUM	Procedure	Dummy variable
HEAD	Global	Current aquifer head at each cell (double precision), dimension (NCOL,NROW,NLAY)
I	Procedure	Counter for rows
IBOUND	Global	Cell status flag, dimension (NCOL,NROW,NLAY)
ICBCFL	Global	Output-control flag to signal whether cell-by-cell flows will be saved ($\neq$0)
ILAKE	Package	Integer array of lake parameters, dimension (9,NLAKES)
ILKCBC	Package	Flag to signal whether lake cell-by-cell flows will be saved (>0) when ICBCFL is set
ILKOUT	Package	Lake stage/budget output flag and unit number: >0 stage/budget unit number (ASCII output recorded depending on the value of IOUTOP in each stress period), $\leq$0 Do not write stage/budget records
IOUT	Global	Unit number of main (list) output file
IOUTOP	Procedure	Flag indicating type of output desired
ISIMMODE	Procedure	Input simulation mode for a lake: 0 fixed-constant, 1 fixed-interpolated, 2 steady-state, 3 transient
ISTRIN	Package	Array of inflow stream segment numbers, dimension (MXSTRIN)

Table 7. List of Variables in Subroutine LAK2BD (continued).

Variable Name	Scope	Description
ISTRM	Stream	For each stream reach: layer, row, column, segment number, reach number, dimension (5,NSTREM)
ISTROUT	Package	Array of outflow stream parameters, dimension (3,MXSTROUT)
ITERLAKE	Procedure	Maximum iterations allowed for stage solution
J	Procedure	Counter for columns
K	Procedure	Counter for layers, lakes
KPER	Global	Stress period number
KSTP	Global	Time step number
LAKENAME	Global	Array of lake names, 10-character string, dimension (NLAKES)
LISTRIN	Procedure	Location in the ISTRIN array of the first stream inflow segment for the current lake
LISTROUT	Procedure	Location in the ISTROUT array of the first stream outflow definition for the current lake
LRLNODE	Procedure	Location in the RLNODE array of the first node definition for the current lake
MSUM	Global	Current number of aquifer budget components
MXNODE	Package	Total number of lake cells (all lakes)
MXRATEQ	Package	Total number of rating equation components (all stream outflows, all lakes)
MXSTRIN	Package	Total number of stream inflows (all lakes)
MXSTROUT	Package	Total number of stream outflows (all lakes)
NCOL	Global	Number of columns in model grid
NLAKES	Package	Number of lakes
NLAY	Global	Number of layers in model grid
NODES	Procedure	Number of lake cells in current lake
NROW	Global	Number of rows in model grid

Table 7. List of Variables in Subroutine LAK2BD (continued).

Variable Name	Scope	Description
NSS	Stream	Number of stream segments
NSTREM	Stream	Number of stream reaches
NSTRIN	Procedure	Number of stream inflows for current lake
NSTROUT	Procedure	Number of stream outflows for current lake
NSUBSTEPS	Package	Number of sub-time steps used for transient stage simulation
PCTERROR	Procedure	Steady-state volumetric balance percentage error for a lake
QEVAP	Procedure	Precipitation/Evaporation outflow from lake
QNET	Procedure	Net inflow to lake
QPRECIP	Procedure	Precipitation/Evaporation inflow to lake
QRUNOFF	Procedure	Runoff (withdrawal) (net) flow to lake
QSEEPAGE	Procedure	Groundwater flux (net) to lake
QSTRIN	Procedure	Total stream inflow to lake
QSTROUT	Procedure	Total stream outflow from lake
RATEQ	Package	Array of rating equation parameters, dimension (4,MXRATEQ)
RLAKE	Package	Real (double precision) array of lake parameters, dimension (9,NLAKES)
RLNODE	Package	Array of lake node parameters, dimension (5,MXNODE)
STAGE	Procedure	Lake stage (double precision)
STAGENW	Procedure	New lake stage (double precision) after update in transient mode
STRM	Stream	For each stream reach: segment inflow, stage, conductance, bottom, top, width, slope, roughness, reach outflow, reach inflow, leakage, dimension (11,NSTREM)
TEXT	Procedure	Label for use in cell-by-cell saving and budget printing, 4-character string, dimension (4)
TOTIM	Global	Total elapsed simulation time
TOTIN	Procedure	Total flow from lakes to aquifer

Table 7. List of Variables in Subroutine LAK2BD (continued).

Variable Name	Scope	Description
TOTOUT	Procedure	Total flow from aquifer to lakes
VBNM	Global	Labels for aquifer volumetric budget, 4-character string, dimension (4,20)
VBVL	Global	Aquifer volumetric budget entries, dimension (4,20), for each budget component N: (1,N) cumulative volume into aquifer, (2,N) cumulative volume out of aquifer, (3,N) current flow rate into aquifer, (4,N) current flow rate out of aquifer
VOL	Procedure	Lake volume
WETAR	Procedure	Lake wetted area

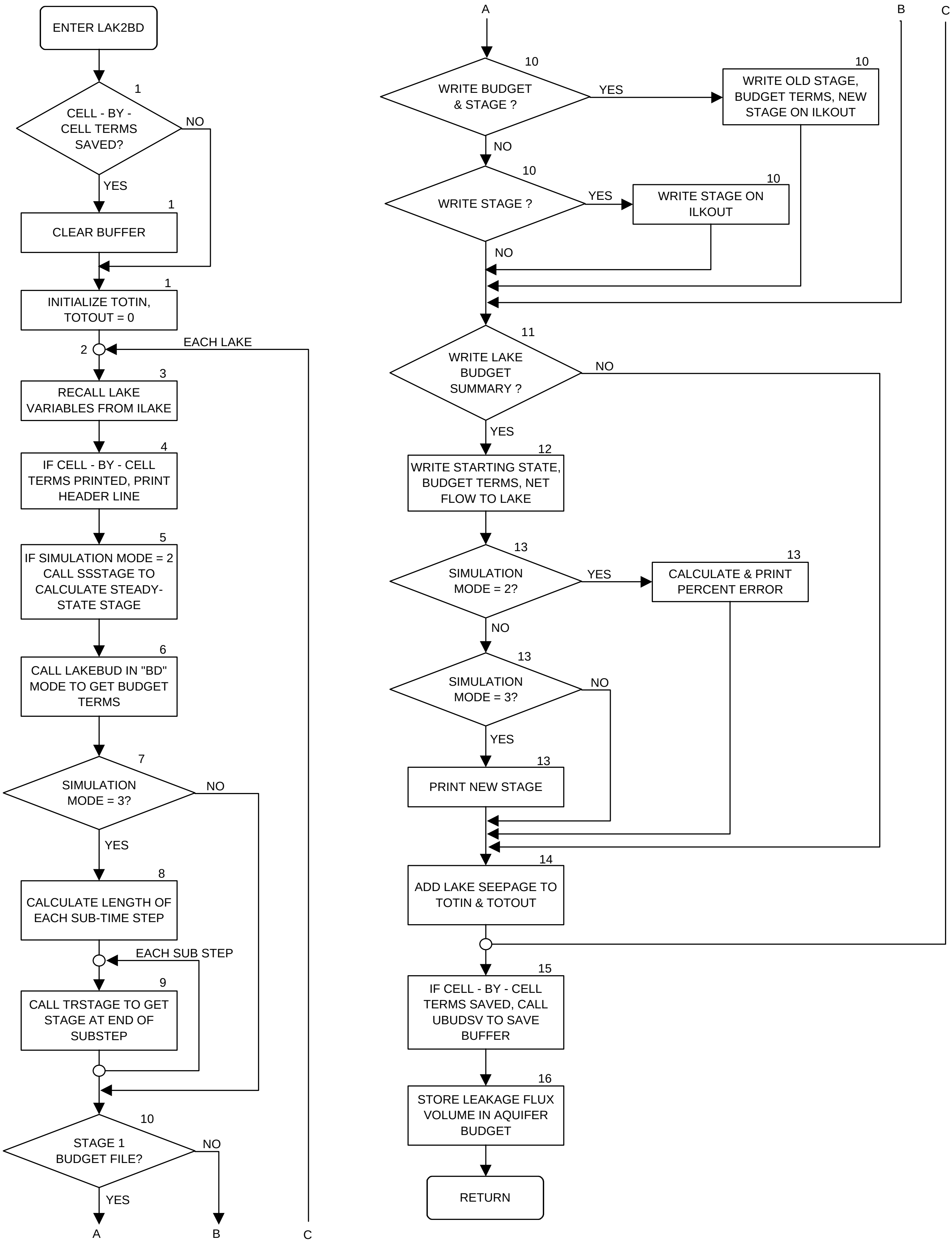


Figure 14. Flowchart for Subroutine LAK2BD

8) Calculate the sub-time-step length as the current MODFLOW time-step length (*DELT*) divided by the number of sub-steps (*NSUBSTEPS*).

9) For each sub-time-step, call *TRSTAGE* to calculate the stage at the end of each sub-step.

10) Set flow rates from calculated components of the *BUDGET* array. If there is output to the stage file (*ILKOUT* > 0) and writing of stage and budget is enabled (*IOUTOP* includes an 8), record the lake stage and budget for this time step, else if there is a stage output file and only stage is to be recorded (*IOUTOP* contains a 4) then record the stage for this time step.

11) If the volumetric budget is to be printed for this time step (*IOUTOP* contains a 4), then proceed to step 12. Otherwise skip to step 14.

12) Print the stage (starting stage for transient solver mode), area, and volume of the lake to the main output file. Also print each budget term along with the total inflow to the lake and total outflow from the lake, from the *BUDGET* array. Print the net inflow to the lake.

13) If the lake is in steady-state solver mode (*ISIMMODE* = 2), calculate and print the mass balance percentage error. If the lake is in transient mode (*ISIMMODE* = 3), print the new stage (for the next time step).

14) Add the total lake cell flux from the lake to the groundwater to *TOTIN*. Add the groundwater-to-lake flux to *TOTOUT*.

15) If cell-by-cell flows are to be saved, call *UBUDSV* to save them to unit *ILKCBC*.

16) Store the lake leakage flow rate and accumulate lake leakage volume in the current slot in the MODFLOW aquifer budget. Increment the variable *MSUM* for the next package to store fluxes and volumes.

4.4.7 Subroutine *LAKEBUD*

The *LAKEBUD* subroutine calculates all of the volumetric budget components for a single modeled lake. It is called by *LAK2BD* (“BD” mode) for reporting the volumetric budget at the end of a time step, by *SSSTAGE* (“SS” mode) and to determine the net flow to the lake and the derivative of the net flow with respect to stage, and by *TRSTAGE* (“TR” mode) to determine the current volume and wetted area. A flow chart for this procedure is presented in Figure 15, and a list of variables used is presented in Table 8. The steps of this procedure are as follows:

1) Get local lake variables from *RLAKE*. Note that *RLAKE* and *ILAKE* are 9-element arrays in this procedure. The calling procedures call *LAKEBUD* with the correct address for these variables for this lake.

Table 8. List of Variables in Subroutine LAKBUD.

Variable Name	Scope	Description
ARTRIB	Stream	Streamflow out of last reach in each segment, dimension (NSS)
BUDGET	Procedure	Lake-wide volumetric budget components, dimension (2,5)
BUFF	Global	Buffer used to accumulate cell-by-cell flows for saving to disk, dimension (NCOL,NROW,NLAY)
DQDS	Procedure	Derivative of flow with respect to stage
DRYRCH	Procedure	Recharge rate for shore cells (double precision)
EVAP	Procedure	Evaporation (wetted-area dependent) rate (double precision)
FCT	Procedure	Function indicator (2-character string): ‘FM’ formulate groundwater equations, ‘BD’ record budget, ‘SS’ steady-state solve, ‘TR’ transient-stage solve
HEAD	Global	Current aquifer head at each cell (double precision), dimension (NCOL,NROW,NLAY)
IBOUND	Global	Cell status flag, dimension (NCOL,NROW,NLAY)
IFLAG	Procedure	Flag indicating type of output desired
IOUT	Global	Unit number of main (list) output file
ISTRIN	Package	Array of inflow stream segment numbers, dimension (MXSTRIN)
ISTROUT	Package	Array of outflow stream parameters, dimension (3,MXSTROUT)
K	Procedure	Counter for budget terms, stream inflows, stream outflows
LRATEQ	Procedure	Location in the RATEQ array of the first rating equation component definition for the current stream outflow
MXRATEQ	Package	Total number of rating equation components (all stream outflows, all lakes)
NCOL	Global	Number of columns in model grid
NLAY	Global	Number of layers in model grid
NODES	Procedure	Number of lake cells in current lake

Table 8. List of Variables in Subroutine LAKBUD (continued).

Variable Name	Scope	Description
NRATEQ	Procedure	Number of rating equation components for current stream outflow
NROW	Global	Number of rows in model grid
NSS	Global	Number of stream inflows for current lake
NSTRIN	Procedure	Number of stream inflows for current lake
NSTROUT	Procedure	Number of stream outflows for current lake
PRECIP	Procedure	Precipitation (total-area dependent) rate (double precision)
QNET	Procedure	Net inflow to lake
RATEQ	Package	Array of rating equation parameters, dimension (4,MXRATEQ)
RLAKE	Package	Real (double precision) array of lake parameters, dimension (9)
RLNODE	Package	Array of lake node parameters, dimension (5,NODES)
RUNOFF	Procedure	Runoff (withdrawal) flow rate (double precision)
STAGE	Procedure	Lake stage (double precision)
TOTAREA	Procedure	Total (maximum) lake area (double precision)
VOL	Procedure	Lake volume
WETAR	Procedure	Lake wetted area

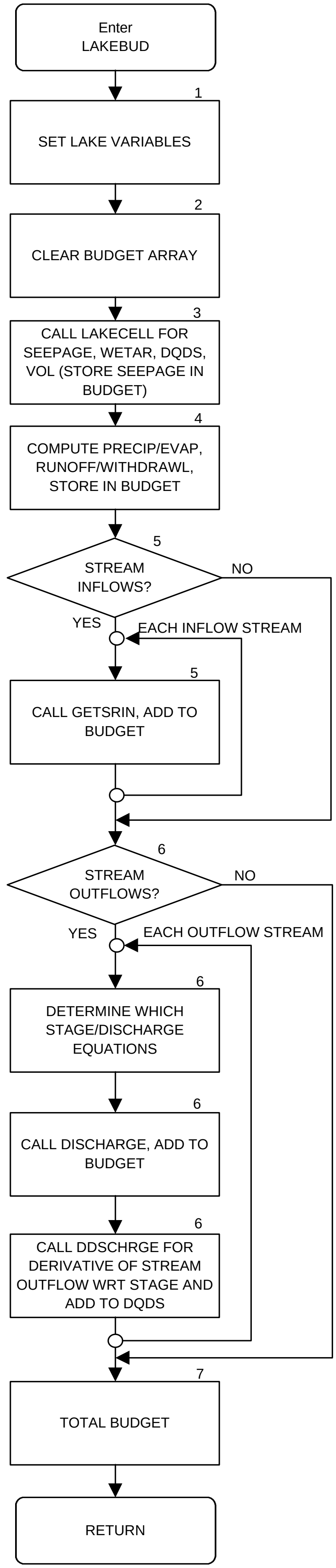


Figure 15. Flowchart for Subroutine LAKEBUD

2) Clear all elements of *BUDGET* to zero. *BUDGET* is a dimension (2,5) array. The inner index is 1 for lake inflows and 2 for lake outflows. The outer index is 1 for precipitation and evaporation, 2 for runoff (or direct withdrawal), 3 for stream inflows and outflows, 4 for lake-groundwater flux, and 5 for total inflows and outflows.

3) Call *LAKECELL* to calculate wetted area, volume, lake-groundwater flux, and the derivative of lake-groundwater flux with respect to stage. Store total groundwater-to-lake flux in *BUDGET*(1,4) and total lake-to-groundwater flux in *BUDGET*(2,4).

4) Multiply *PRECIP* by the total lake area and add to *BUDGET* (add to *BUDGET*(1,1) if inflow, or *BUDGT*(2,1) if outflow). Multiply *EVAP* by the wetted area and add to *BUDGET* (add to *BUDGET*(1,1) if inflow, or *BUDGT*(2,1) if outflow). Add *RUNOFF* to *BUDGET*(1,2) if positive, or *BUDGET*(2,2) if negative.

5) For each of the lake's stream inflows, call *GETSTRIN* to retrieve the inflow rate from the Stream Package variables. Add to *BUDGET*(1,3).

6) For each of the lake's stream outflows, determine the number and location (in *RATEQ*) of rating equation components. Call *DSCHRGE* to determine the stream outflow rate and add to *BUDGET*(2,3). Call *DDSCHRGE* to calculate the derivative of stream outflow with respect to stage, and add to *DQDS*.

7) Sum lake inflows in *BUDGET*(1,5). Sum lake outflows in *BUDGET*(2,5). Set *QNET* to net inflow to lake.

4.4.8 Subroutine *LAKECELL*

The *LAKECELL* subroutine loops through all cells in a single lake to accumulate groundwater flows, formulate the MODFLOW equations, determine the wetted area and volume, and determine the derivative of lake-groundwater flow with respect to stage. This subroutine is called from *LAKEBUD* (in "BD", "SS", or "TR" mode), and from *LAK2FM* (in "FM" mode). A flow chart for this procedure is presented in Figure 16, and a list of variables used is presented in Table 9. The steps of this procedure are as follows:

1) Initialize lake inflow (*SEEPIN*) and lake outflow (*SEEPOUT*) to zero. Also initialize the wetted area (*WETAR*), volume (*VOL*), and derivative (*DQDS*) to zero.

2) Perform the remaining steps in this procedure for each lake node. Note that *RLNODE* is correctly positioned by the calling procedure such that the first element of *RLNODE* is the first cell in the current lake.

Table 9. List of Variables in Subroutine LAKECELL.

Variable Name	Scope	Description
BOT	Procedure	Bottom elevation of lakebed at current lake cell
COND	Procedure	Hydraulic conductivity/conductance of lakebed at current lake cell
CSEEP	Procedure	Seepage rate from lake to groundwater cell
DQDS	Procedure	Derivative of flow with respect to stage
DRYRCH	Procedure	Recharge rate for shore cells (double precision)
FCT	Procedure	Recharge rate for shore cells (double precision)
H	Procedure	Groundwater head of cell
HCOF	Global	Coefficient of head in the finite difference equation, dimension (NCOL,NROW,NLAY)
HEAD	Global	Current aquifer head at each cell (double precision), dimension (NCOL,NROW,NLAY)
IBCELL	Procedure	IBOUND value for cell
IBOUND	Global	Cell status flag, dimension (NCOL,NROW,NLAY)
IC	Procedure	Column number
IFLAG	Procedure	Flag indicating type of output desired
IL	Procedure	Layer number
IOUT	Global	Unit number of main (list) output file
IR	Procedure	Row number
K	Procedure	Counter for rating equation components
NCOL	Global	Number of columns in model grid
NLAY	Global	Number of layers in model grid
NODES	Procedure	Number of lake cells in current lake
NROW	Global	Number of rows in model grid
POS	Procedure	Absolute cell position
RCH	Procedure	Recharge rate applied at shore cell

Table 9. List of Variables in Subroutine LAKECELL (continued).

Variable Name	Scope	Description
RHS	Global	Right-hand side array for MODFLOW equations, dimension (NCOL,NROW,NLAY)
RLNODE	Package	Array of lake node parameters, dimension (5,NODES)
SEEPIN	Procedure	Total flow to lake from groundwater
SEEPOUT	Procedure	Total flow to groundwater from lake
STAGE	Procedure	Lake stage (double precision)
TOP	Procedure	Top elevation of lakebed at current lake cell
VOL	Procedure	Lake volume
WETAR	Procedure	Lake wetted area

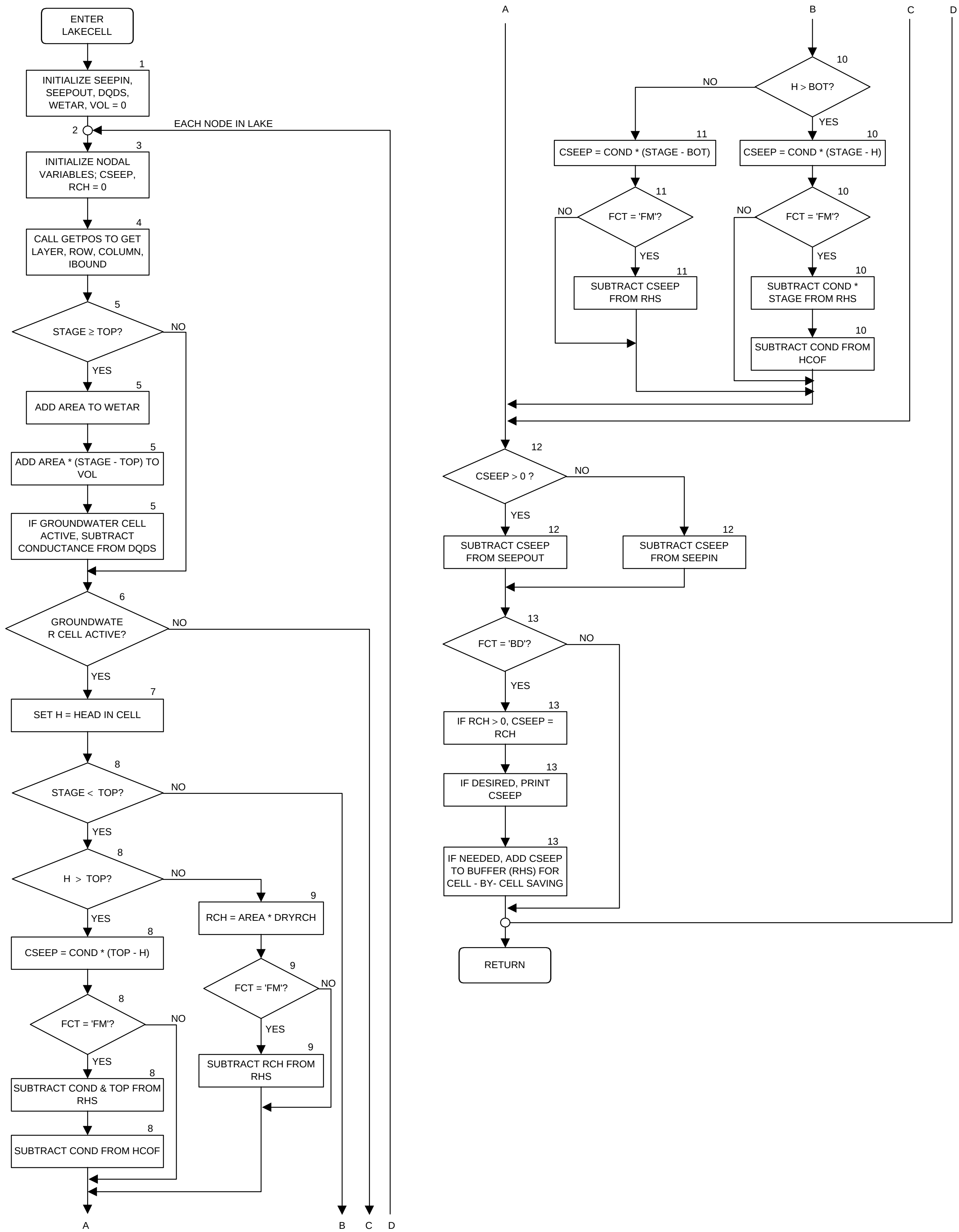


Figure 16. Flowchart for Subroutine LAKECELL

3) Initialize cell seepage (*CSEEP*) and recharge (*RCH*) to zero. Set *POS* equal to the coordinate number stored in the 1st position in *RLNODE*. Set *TOP*, *BOT*, *AREA*, and *COND* from *RLNODE*.

4) Call *GETPOS* to determine, from *POS*, the layer, row, and column of the lake cell, as well as the groundwater cell status, *IBCELL*.

5) If the lake *STAGE* is greater than or equal to the lakebed *TOP*, then add *AREA* to *WETAR*, add *AREA**(*STAGE* - *TOP*) to *VOL*, and subtract *COND* from *DQDS* if the groundwater cell is active.

6) If the groundwater cell is active (*IBCELL* > 0) then continue with step 7, otherwise skip the remaining steps for this node.

7) Store the groundwater cell head in *H*.

8) If the cell is a shore cell (*STAGE* < *TOP*) and the head is above the lakebed (*H* > *TOP*), then this cell is a drain. Set *CSEEP* to *COND**(*TOP* - *H*). If the mode is “FM”, subtract *COND***TOP* from *RHS*, and subtract *COND* from *HCOF*.

9) If the cell is a shore cell (*STAGE* < *TOP*) and the head is at or below the lakebed top (*H* <= *TOP*), then there is no lake-groundwater flow, but dry recharge can be applied. Set *RCH* to *AREA***DRYRCH*. If the mode is “FM”, subtract *RCH* from *RHS*.

10) If the cell is a wetted cell (*STAGE* >= *TOP*) and the head is above the lakebed bottom (*H* > *BOT*), then this cell is a connected river-type cell. Set *CSEEP* to *COND**(*STAGE* - *H*). If the mode is “FM”, subtract *COND***STAGE* from *RHS*, and subtract *COND* from *HCOF*.

11) If the cell is a wetted cell (*STAGE* >= *TOP*) and the head is at or below the lakebed bottom (*H* <= *BOT*), then this cell is a disconnected (maximum seepage) river-type cell. Set *CSEEP* to *COND**(*STAGE* - *BOT*). If the mode is “FM”, subtract *CSEEP* from *RHS*.

12) If *CSEEP* is positive, subtract it from *SEEPOUT*. If *CSEEP* is negative, subtract it from *SEEPIN*.

13) If in “BD” mode, check the output option flag (*IFLAG* also *IOUTOP* in other procedures). If the flag is odd, print the lake-to-groundwater seepage (or dry recharge) rate to the output file. If the flag contains a 16, add the seepage or dry recharge to the buffer (passed in as *RHS*).

4.4.9 Function *DSCHRGE*

The *DSCHRGE* function calculates and returns the outflow rate for a particular stream outflow using the appropriate rating equation component. This function is called by *LAK2AD*, *SSSTAGE*, and *LAKEBUD*. A flow chart for this procedure is presented in Figure 17, and a list of variables used is presented in Table 10. The steps of this procedure are as follows:

1) Step through this stream outflow's rating equation components. Stop as soon as one of the equations has a cutoff elevation (stored in the 1st position in *OUTFALL*) less than or equal to the current lake stage, and proceed to step 2. If all of the equations have cutoff elevations above the current stage, go to step 3.

2) Set the stream outflow rate using this rating equation. The outflow rate is equal to *CONST**(*STAGE* - *ELEV*)***EXPNT*. Where *CONST*, *ELEV*, and *EXPNT*, are stored in *OUTFALL*. Set *DSCHRGE* to this rate and return.

3) The outflow is not active. Set *DSCHRGE* to zero and return.

4.4.10 Function *DDSCHRGE*

The *DDSCHRGE* function calculates and returns the derivative of the outflow rate with respect to stage for a particular stream outflow using the appropriate rating equation component. This function is called by *LAKEBUD*. A flow chart for this procedure is presented in Figure 18, and a list of variables used is presented in Table 11. The steps of this procedure are as follows:

1) Step through this stream outflow's rating equation components. Stop as soon as one of the equations has a cutoff elevation (stored in the 1st position in *OUTFALL*) less than or equal to the current lake stage, and proceed to step 2. If all of the equations have cutoff elevations above the current stage, go to step 3.

2) Set the derivative of the outflow rate with respect to stage using this rating equation. The derivative is equal to *EXPNT***CONST**(*STAGE* - *ELEV*)**(*EXPNT*-1). Where *CONST*, *ELEV*, and *EXPNT*, are stored in *OUTFALL*. Set *DDSCHRGE* to this derivative and return.

3) The outflow is not active. Set *DDSCHRGE* to zero and return.

4.4.11 Function *GETSTRIN*

The *GETSTRIN* function determines and returns the inflow rate for a particular stream inflow by looking up the outflow from the specified segment in the Stream Package's *ARTRIB* array. The function is called by *LAKEBUD*. A flow chart for this procedure is presented in

Table 10. List of Variables in Subroutine DSCHRGE.

Variable Name	Scope	Description
K	Procedure	Counter for rating equation components
NRAT	Procedure	Number of rating equation components
OUTFALL	Procedure	Rating equation parameters, dimension (4,NRAT)
STAGE	Procedure	Lake stage (double precision)

Table 11. List of Variables in Subroutine DDSCHRGE.

Variable Name	Scope	Description
K	Procedure	Counter for rating equation components
NRAT	Procedure	Number of rating equation components
OUTFALL	Procedure	Rating equation parameters, dimension (4,NRAT)
STAGE	Procedure	Lake stage (double precision)

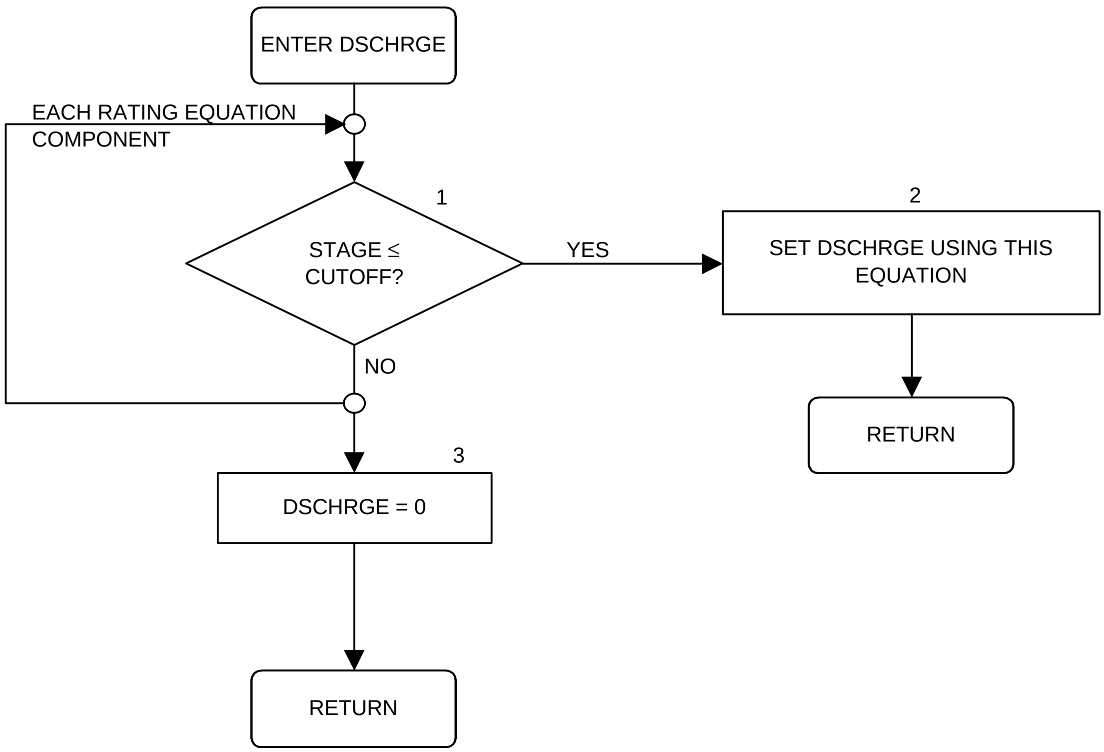


Figure 17. Flowchart for Function DSCHRGE

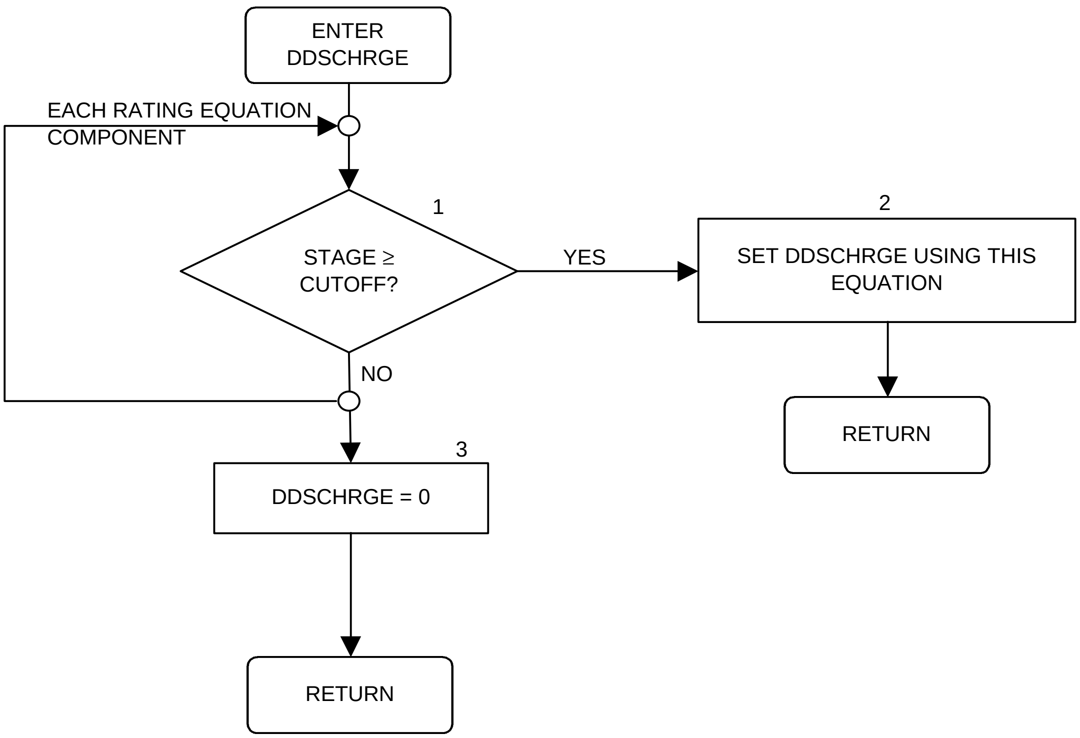


Figure 18. Flowchart for Function DDSCHRGE

Figure 19, and a list of variables used is presented in Table 12. The steps of this procedure are as follows:

1) Determine if the specified stream segment is out of range (less than 1 or greater than the current number of stream segments). If so, return.

2) Set *GETSTRIN* to the appropriate value from the *ARTRIB* array.

4.4.12 Subroutine ***SETSTROUT***

The *SETSTROUT* subroutine sets the lake-to-stream flow rate as the inflow rate to the appropriate stream segment, using the Stream Package's *STRM* array. A flow chart for this procedure is presented in Figure 20, and a list of variables used is presented in Table 13. The steps of this procedure are as follows:

1) Determine if the stream outflow segment is less than 1 (not connected to a Stream Package segment). If it is, return.

2) Step through each stream reach. Look for the first reach that has the appropriate segment number identified in *ISTRM*. If it if found, go to step 4.

3) The specified outflow segment is not a valid stream segment. Print an error message to the screen and stop.

4) Set the inflow to the identified reach to the lake-to-stream flow rate using the 1^{st} position in *STRM*.

4.4.13 Subroutine ***SETPOS***

The *SETPOS* subroutine stores the absolute cell position in a single integer (4-byte) variable (which is actually the 1^{st} position of *RLNODE* in the calling procedure, *LAK2RP1*) . A flow chart for this procedure is presented in Figure 21, and a list of variables used is presented in Table 14. The steps of this procedure are as follows:

1) If the layer number is zero, set it to -1.

2) Using the absolute value of the layer number, set *IPOS* to the absolute cell position (numbered from 1 to *NCOL***NROW***NLAY*). The absolute position is found as (ABS(*ILAY*)-1)**NROW***NCOL* + (*IROW*-1)**NCOL* + *ICOL*.

3) If the layer number is less than zero, set *IPOS* to -*IPOS*. A negative position indicates that the lake cell will be connected to the specified layer if it is active, or the next highest layer below the specified layer that is active.

Table 12. List of Variables in Subroutine GETSTRIN.

Variable Name	Scope	Description
ARTRIB	Stream	Streamflow out of last reach in each segment, dimension (NSS)
ISEG	Procedure	Stream segment number for stream inflow or outflow
NSS	Global	Number of stream inflows for current lake

Table 13. List of Variables in Subroutine SETSTROUT.

Variable Name	Scope	Description
ISEG	Procedure	Stream segment number for stream inflow or outflow
ISTRM	Stream	For each stream reach: layer, row, column, segment number, reach number, dimension (5,NSTREM)
K	Procedure	Counter for stream reaches
NSTREM	Stream	Number of stream reaches
QSTROUT	Procedure	Total stream outflow from lake
STRM	Stream	For each stream reach: segment inflow, stage, conductance, bottom, top, width, slope, roughness, reach outflow, reach inflow, leakage, dimension (11,NSTREM)

Table 14. List of Variables in Subroutine SETPOS.

Variable Name	Scope	Description
ICOL	Procedure	Column number of model cell connected to lake
ILAY	Procedure	Layer number of model cell connected to lake
IPOS	Procedure	Stored cell number
IROW	Procedure	Row number of model cell connected to lake
NCOL	Global	Number of columns in model grid
NLAY	Global	Number of layers in model grid

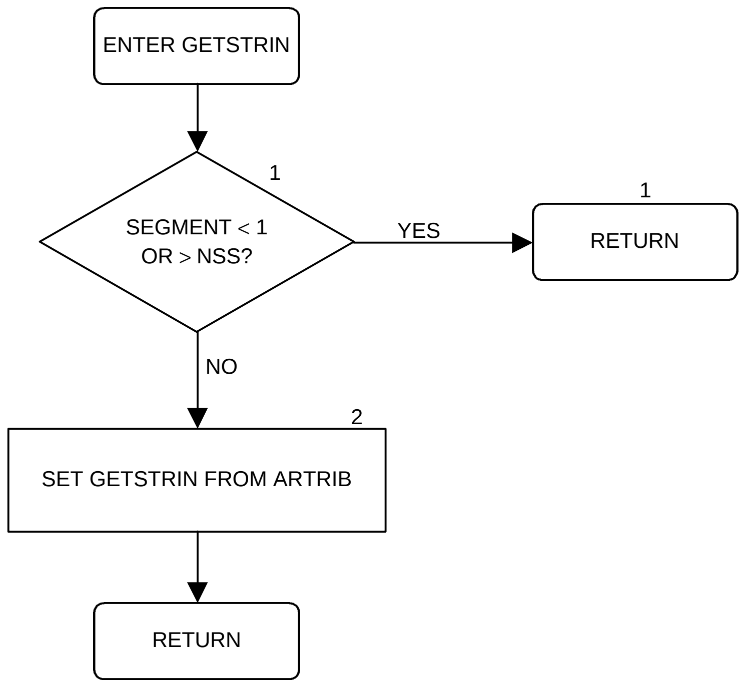


Figure 19. Flowchart for Function GETSTRIN

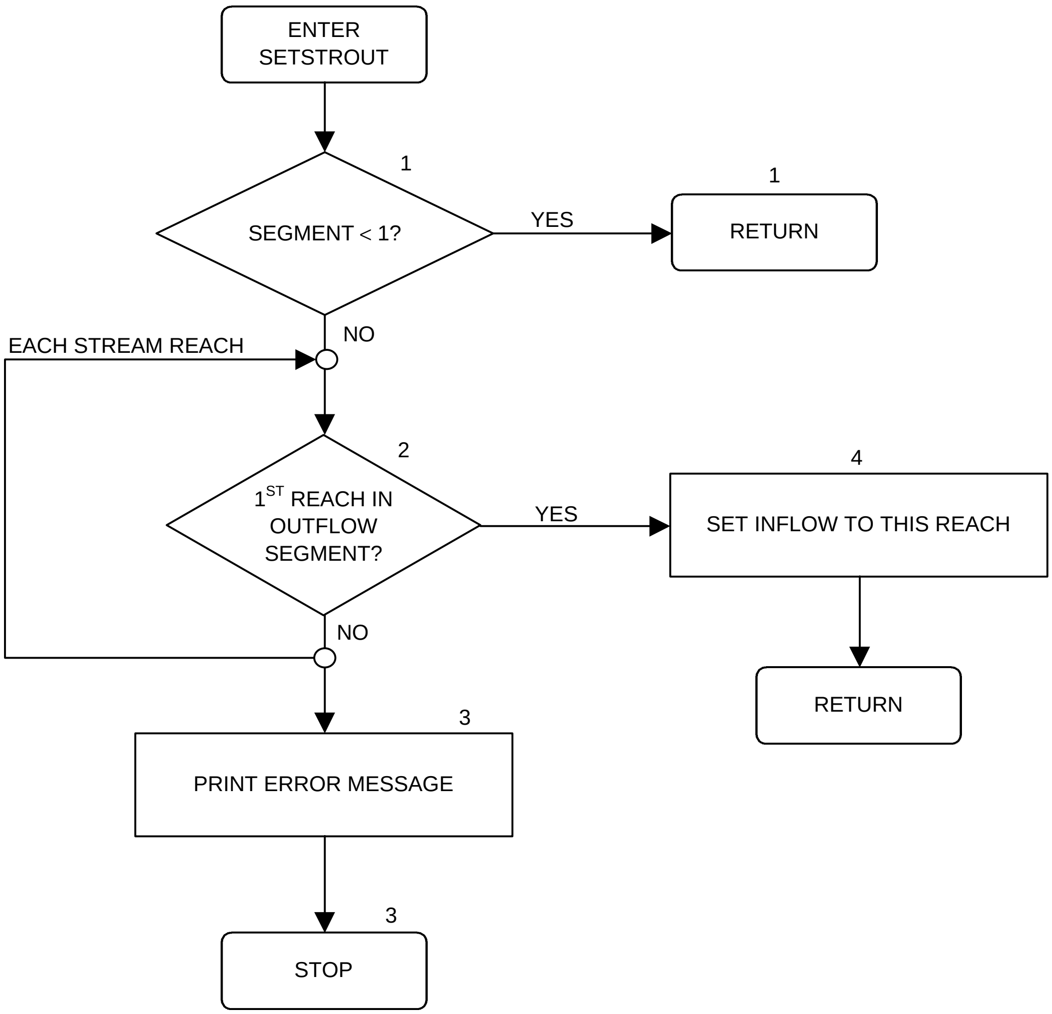


Figure 20. Flowchart for Subroutine SETSTROUT

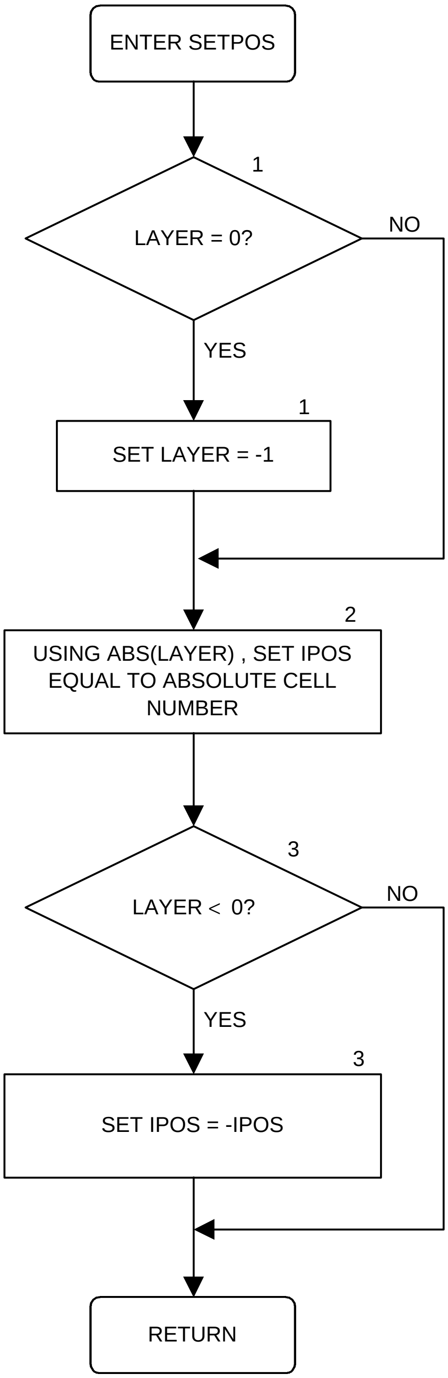


Figure 21. Flowchart for Subroutine SETPOS

4.4.14 Subroutine *GETPOS*

The *GETPOS* subroutine determines the layer, row, and column stored in the absolute position number (the 1st position in *RLNODE*). This subroutine is called by *LAKECELL* for each lake cell. A flow chart for this procedure is presented in Figure 22, and a list of variables used is presented in Table 15. The steps of this procedure are as follows:

1) Using the absolute value of *IPOS*, determine the layer, row and column.

2) If *IPOS* is positive, set *IBCELL* to the value of *IBOUND* at the specified cell and return.

3) Step from the current layer to *NLAY*, assigning *IBCELL* from *IBOUND*, and returning when the cell is not inactive (return when *IBCELL* <> 0).

4) If all cells from the original layer to *NLAY* are inactive, set the layer number to the original layer number.

4.4.15 Subroutine *SSSTAGE*

The *SSSTAGE* subroutine calculates the steady-state (equilibrium) stage of a lake. This subroutine is called for each lake by *LAK2FM* during each MODFLOW iteration, and by *LAK2BD* at the end of each time step (for lakes in steady-state stage solver mode). A flow chart for this procedure is presented in Figure 23, and a list of variables used is presented in Table 16. The steps of this procedure are as follows:

1) Initialize *STAGE*, *STAGEMN*, *STAGEMX*, and *CONVCRIT* from *RLAKE*. Set *CONVCRIT* 10 times lower than the user-specified value, so that the check made in *LAK2FM* will be made to a more accurate stage value.

2) Set *NEWTON* equal to TRUE. The default method for solving for steady-state stage is Newton's method.

3) For each stage-solver iteration, up to *ITERLAKE*, perform steps 4 through 11 to iteratively solve for lake stage.

4) Set *STAGEOLD* to *STAGE*.

5) Call *LAKEBUD* in "SS" mode to get net flow to lake and the derivative of net flow with respect to stage.

6) If the net flow is positive, then the stage is too low, reset *STAGEMN* to *STAGE*. Otherwise, set *STAGEMX* to *STAGE*.

Table 15. List of Variables in Subroutine GETPOS.

Variable Name	Scope	Description
IBCEL	Procedure	IBOUND value of lake cell
IBOUND	Global	Cell status flag, dimension (NCOL,NROW,NLAY)
ICOL	Procedure	Column number of model cell connected to lake
ILAY	Procedure	Layer number of model cell connected to lake
IROW	Procedure	Row number of model cell connected to lake
IPOS	Procedure	Stored cell number
ITMP	Procedure	Temporary variable for ILAY
NCOL	Global	Number of columns in model grid
NLAY	Global	Number of layers in model grid
NRC	Procedure	Number of cells in a layer
NROW	Global	Number of rows in model grid

Table 16. List of Variables in Subroutine SSSTAGE.

Variable Name	Scope	Description
ARTRIB	Stream	Streamflow out of last reach in each segment, dimension (NSS)
BUDGET	Procedure	Lake-wide volumetric budget components, dimension (2,5)
CONVCRIT	Procedure	Convergence criterion for stage solution (double precision)
DQDS	Procedure	Derivative of flow with respect to stage
HEAD	Global	Current aquifer head at each cell (double precision), dimension (NCOL,NROW,NLAY)
IBOUND	Global	Cell status flag, dimension (NCOL,NROW,NLAY)
ITERLAKE	Procedure	Maximum iterations allowed for stage solution
ISEGOUT	Procedure	Stream outflow segment number
ISTRIN	Package	Array of inflow stream segment numbers, dimension (MXSTRIN)
ISTRM	Stream	For each stream reach: layer, row, column, segment number, reach number, dimension (5,NSTREM)
ISTROUT	Package	Array of outflow stream parameters, dimension (3,MXSTROUT)
K	Procedure	Counter for iterations, stream outflows
LRATEQ	Procedure	Location in the RATEQ array of the first rating equation component definition for the current stream outflow
MXRATEQ	Package	Total number of rating equation components (all stream outflows, all lakes)
NCOL	Global	Number of columns in model grid
NEWTON	Procedure	Logical flag indicating whether Newton's method or the range-split method is used for steady-state stage solution
NLAY	Global	Number of layers in model grid
NODES	Procedure	Number of lake cells in current lake
NRATEQ	Procedure	Number of rating equation components for current stream outflow

Table 16. List of Variables in Subroutine SSSTAGE (continued).

Variable Name	Scope	Description
NROW	Global	Number of rows in model grid
NSS	Global	Number of stream inflows for current lake
NSTRIN	Procedure	Number of stream inflows for current lake
NSTREM	Stream	Number of stream reaches
NSTROUT	Procedure	Number of stream outflows for current lake
QNET	Procedure	Net inflow to lake
RATEQ	Package	Array of rating equation parameters, dimension (4,MXRATEQ)
RLAKE	Package	Real (double precision) array of lake parameters, dimension (9)
RLNODE	Package	Array of lake node parameters, dimension (5,NODES)
STAGE	Procedure	Lake stage (double precision)
STAGEMN	Procedure	Minimum lake stage (double precision)
STAGEMX	Procedure	Maximum lake stage (double precision)
STAGEOLD		Original lake stage (double precision)
STRM	Stream	For each stream reach: segment inflow, stage, conductance, bottom, top, width, slope, roughness, reach outflow, reach inflow, leakage, dimension (11,NSTREM)
VOL	Procedure	Lake volume
WETAR	Procedure	Lake wetted area

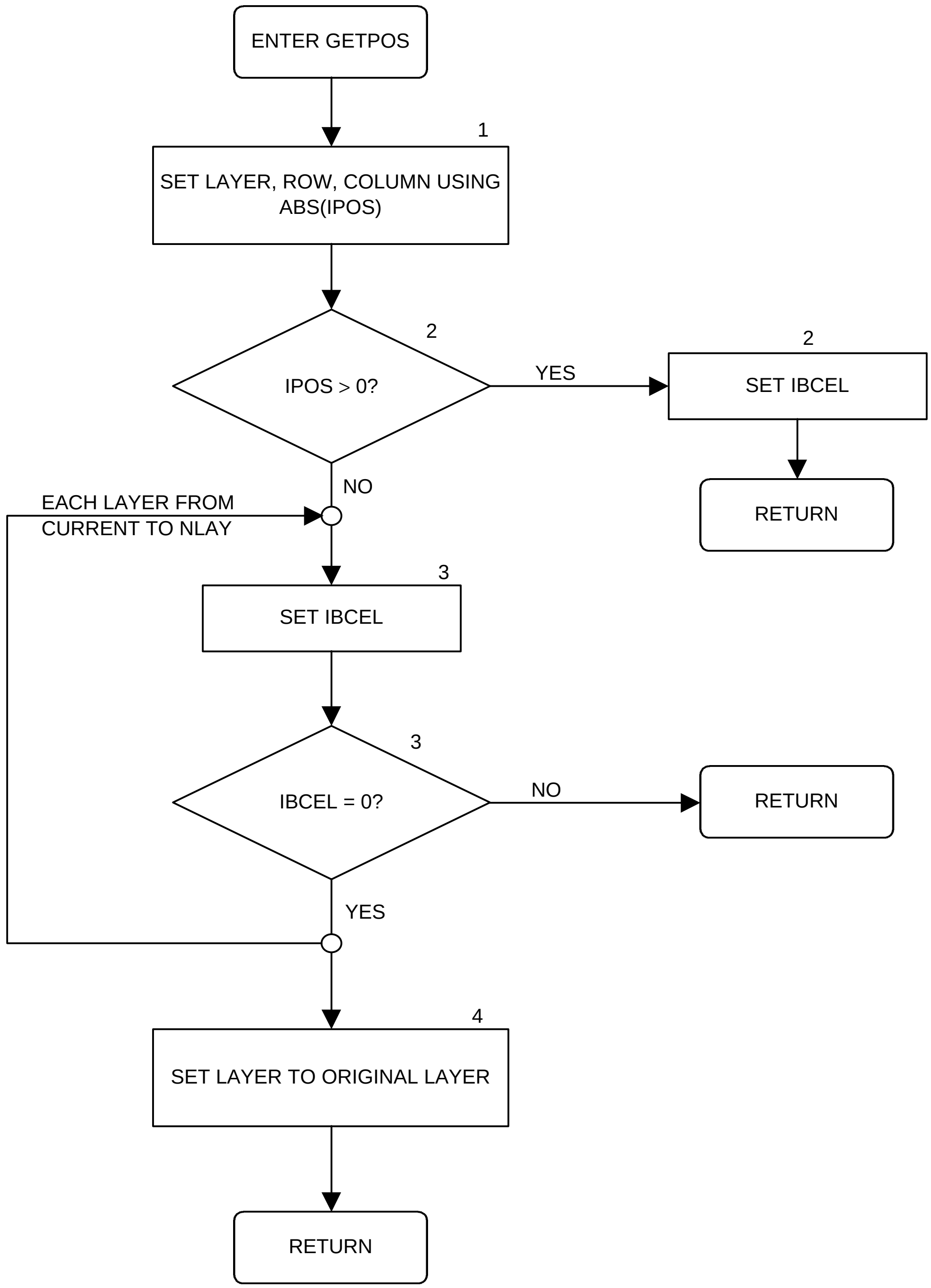


Figure 22. Flowchart for Subroutine GETPOS

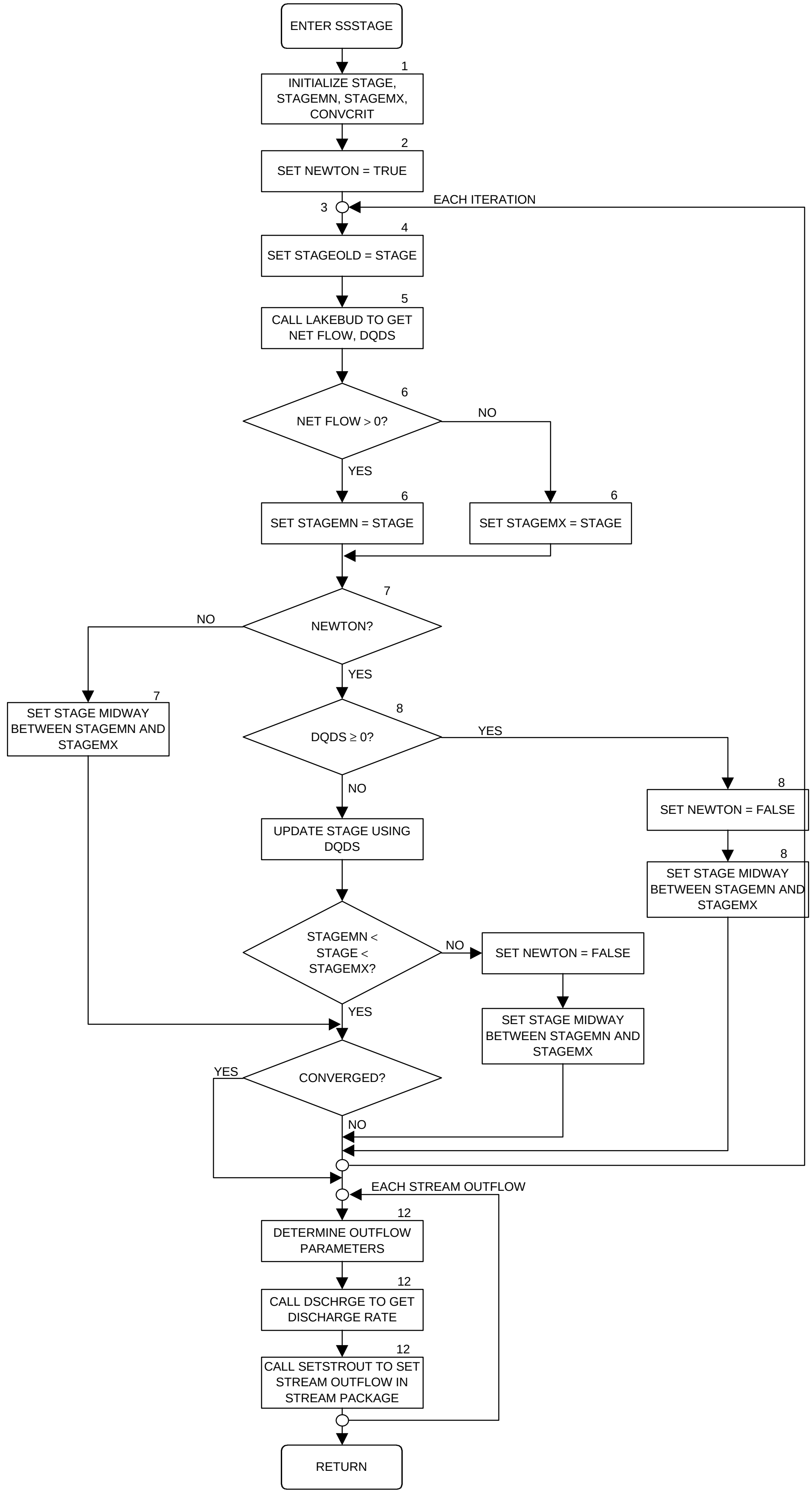


Figure 23. Flowchart for Subroutine SSSTAGE

7) Check *NEWTON* to see whether Newton's method is currently being used, if it is proceed with step 8. If not, use range splitting (set *STAGE* halfway between *STAGEMN* and *STAGEMX*) and skip to step 11.

8) If the derivative of net flow with respect to stage (*DQDS*) is not negative, there are no stream outflows or seepage connections and Newton's method will not work. Set *NEWTON* to FALSE, set *STAGE* halfway between *STAGEMN* and *STAGEMX*, and skip to the next iteration (skip steps 9 through 11).

9) Update the stage using Newton's method: *STAGE* = *STAGE* - *QNET*/*DQDS*

10) If the *STAGE* is not between *STAGEMN* and *STAGEMX*, then Newton's method has failed to converge. Set *NEWTON* to FALSE, set *STAGE* halfway between *STAGEMN* and *STAGEMX*, and skip to the next iteration (skip step 11).

11) Set the 1st position of *RLAKE* to the current stage. If the absolute difference between *STAGE* and *STAGEOLD* is less than or equal to *CONVCRIT*, the stage solution has converged, proceed to step 12. Otherwise continue with the next iteration at step 4.

12) The solution has converged or the maximum number of iterations has been reached. For each of the lake's stream outflows, set the flows by recalling *ISEGOUT*, *LRATEQ*, and *NRATEQ*, from *ISTROUT*, calling *DSCHRGE* to determine the outflow rate and then calling *SETSTROUT* to assign the rate to the appropriate segment in the Stream Package.

4.4.16 Subroutine *TRSTAGE*

The *TRSTAGE* subroutine updates the lake stage for lakes in transient stage solver mode. This subroutine is called by *LAK2BD* for each lake, for each (sub) time step. A flow chart for this procedure is presented in Figure 24, and a list of variables used is presented in Table 17. The steps of this procedure are as follows:

1) Set *STAGEMN*, *STAGEMX*, and *CONVCRIT* from *RLAKE*.

2) Call *LAKEBUD* in "TR" mode to get the current net flow to the lake, wetted area and volume.

3) Set the desired change in volume (*DVOL*) to the net flow times the (sub) time step legth.

4) Set the target volume, *TARGVOL*, to the current volume plus *DVOL*.

5) If the target volume is zero or less, the lake is emptying. Set the stage (1st position in *RLAKE*) to *STAGEMN* and return.

Table 17. List of Variables in Subroutine TRSTAGE.

Variable Name	Scope	Description
ARTRIB	Stream	Streamflow out of last reach in each segment, dimension (NSS)
BUDGET	Procedure	Lake-wide volumetric budget components, dimension (2,5)
CONVCRIT	Procedure	Convergence criterion for stage solution (double precision)
DQDS	Procedure	Derivative of flow with respect to stage
DSTAGE	Procedure	Change in lake stage
DT	Procedure	Length of a sub-time step
DVOL	Procedure	Change in lake volume
HEAD	Global	Current aquifer head at each cell (double precision), dimension (NCOL,NROW,NLAY)
IBOUND	Global	Cell status flag, dimension (NCOL,NROW,NLAY)
ITERLAKE	Procedure	Maximum iterations allowed for stage solution
ISTRIN	Package	Array of inflow stream segment numbers, dimension (MXSTRIN)
ISTROUT	Package	Array of outflow stream parameters, dimension (3,MXSTROUT)
K	Procedure	Counter for iterations
MXRATEQ	Package	Total number of rating equation components (all stream outflows, all lakes)
NCOL	Global	Number of columns in model grid
NLAY	Global	Number of layers in model grid
NODES	Procedure	Number of lake cells in current lake
NROW	Global	Number of rows in model grid
NSS	Global	Number of stream inflows for current lake
NSTRIN	Procedure	Number of stream inflows for current lake
NSTROUT	Procedure	Number of stream outflows for current lake
QNET	Procedure	Net inflow to lake

Table 17. List of Variables in Subroutine TRSTAGE (continued).

Variable Name	Scope	Description
RATEQ	Package	Array of rating equation parameters, dimension (4,MXRATEQ)
RLAKE	Package	Real (double precision) array of lake parameters, dimension (9)
RLNODE	Package	Array of lake node parameters, dimension (5,NODES)
STAGEMN	Procedure	Minimum lake stage (double precision)
STAGEMX	Procedure	Maximum lake stage (double precision)
TARGVOL	Procedure	Target volume of lake
VOL	Procedure	Lake volume
WETAR	Procedure	Lake wetted area

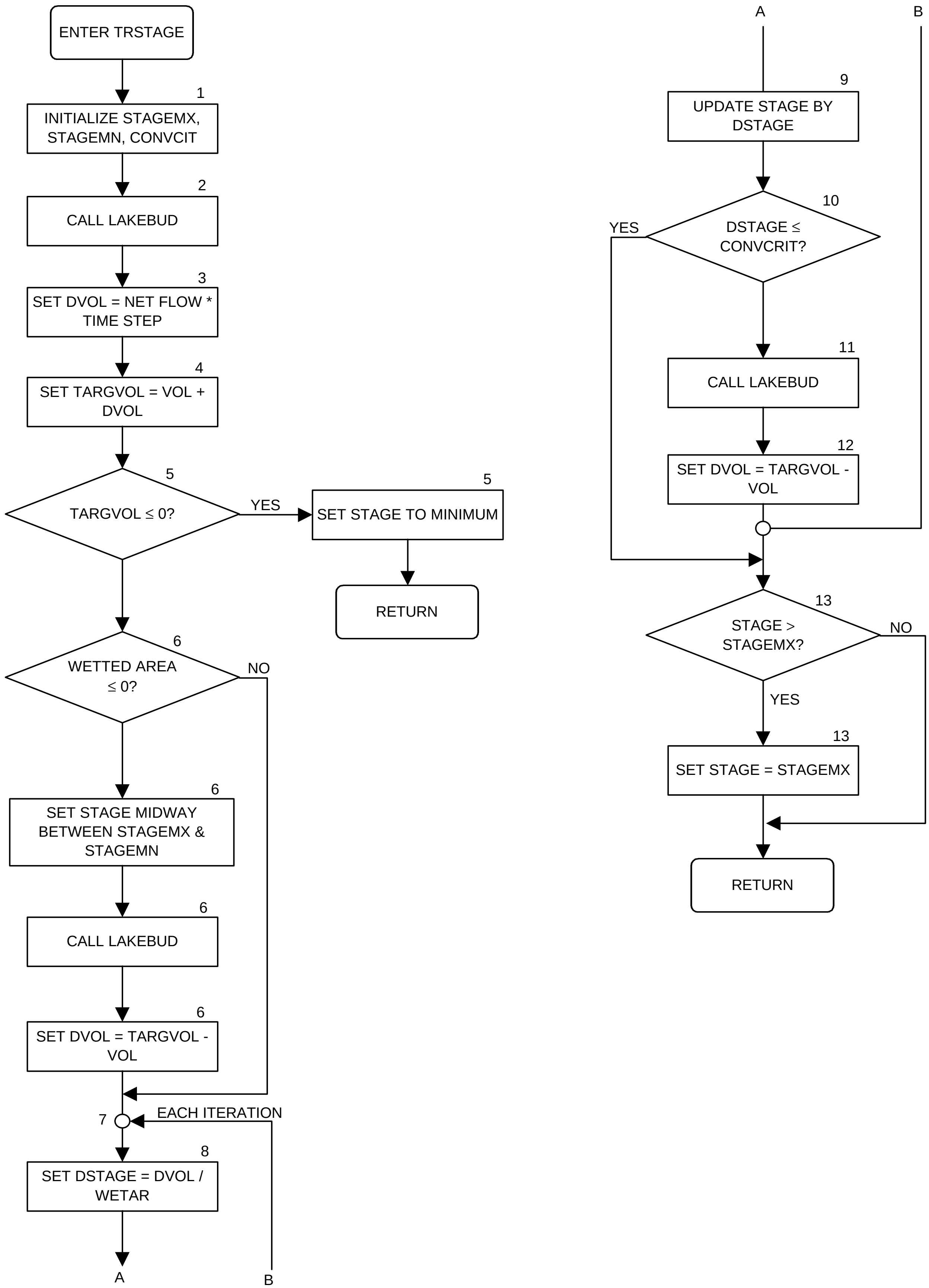


Figure 24. Flowchart for Subroutine TRSTAGE

6) If the current wetted area is not positive, then the lake starting with no wetted cells. As a first guess for the stage that will yield the (non-zero) target volume, set the stage halfway between *STAGEMN* and *STAGEMX*. Call *LAKEBUD* at this stage and set *DVOL* to the target volume minus the volume at the first-guess stage.

7) For each iteration up to *ITERLAKE*, perform steps 8 through 12 to make a new guess at the stage that will yield the target volume.

8) Set the change in stage, *DSTAGE*, to *DVOL*/*WETAR*.

9) Add *DSTAGE* to the current guess for stage.

10) If *DSTAGE* is less than *CONVCRIT* then the stage solver has converged — go to step 13.

11) Call *LAKEBUD* to get the new lake volume.

12) Reset *DVOL* to *TARGVOL* - *VOL* and proceed to the next iteration (step 8)

13) The solution has converged or the maximum iterations have been used. Check for a calculated stage above *STAGEMX*. If the stage is above its maximum, set the lake stage to its maximum.

4.5 Incorporation in MODFLOW

A version of MODFLOW-96 with the Lake Package included is provided along with this documentation (executable and source code, see Appendix D). If another version of MODFLOW is used, the Lake Package code should be added to the MODFLOW source code, and the program should be recompiled. Appendix C provides instructions for incorporating the Lake Package into an existing version of MODFLOW.

5. Code Testing and Verification

5.1 Verification of Lake Package Results Using an Analytical Solution and the Method of High-Conductivity, High Storage Lake Cells for a Circular Lake

A test problem was constructed similar to that reported by Nair and Wilsnack (1998), which consists of a circular lake surrounded by a constant-transmissivity aquifer of very large extent. The head in the aquifer is initially uniform, and the stage in the lake is initially higher than the aquifer head. The specific problem setup is illustrated in Figure 25. In the test problem, the lake is 1500 ft in diameter, the aquifer has a transmissivity of 50,000 ft^2/d and a storage coefficient of 0.001. The initial aquifer head is 80 ft (relative to an arbitrary elevation datum), and the lake stage is initially 82 ft.

This setup is analogous to a slug test in a fully-penetrating borehole in a confined aquifer of infinite extent. The solution for borehole (lake) water level versus time has been determined analytically by Cooper et al. (1967) as a function of borehole (lake) radius, the aquifer transmissivity and storage coefficient, and the initial height of water in the borehole (lake) above the aquifer head. Using a radius of 750 ft, a transmissivity of 50,000 ft^2/d, and a storage coefficient of 0.001, the borehole (lake) water level versus time curve was determined using the AQTESOLV program (Version 1.17 for Windows, HydroSOLVE, Inc., 1996).

A MODFLOW model was constructed for this test problem, having one confined layer, 220 rows, and 220 columns. The horizontal dimensions of the model were 256,900 ft by 256,900 ft. The lake was centered in the model, as shown in Figure 26. Grid spacing over most of the model was 1500 ft in both rows and columns, but was reduced down to 50 ft near the lake. The large size of the model relative to the lake diameter closely approximates the infinte extent assumption of the analytical solution.

In the first simulation, the lake was simulated using high conductivity, high storage cells. Inside the lake, the transmissivity of all cells was set three orders of magnitude higher than the aquifer cells, and the storage coefficient was set to 1.0. The model was run for a total of 100 days using 60 time steps and a time-step multiplier of 1.1. This time-stepping proved to be adequate — further reducing the time step size had little effect on results. The head in row 110, column 110 (near the lake center) was taken to be the lake stage. Because the transmissivity of the lake cells was much larger than the transmissivity of the aquifer cells, the head in the lake varied very little from cell to cell.

In the second simulation, the Lake Package was used. All of the cells inside the lake were made inactive in the IBOUND array. The adjacent aquifer cells were specified as lake cells, each with a lake-to-groundwater conductance of 100,000 ft^2/d (transmissivity times cell

a) Plan View

b) Elevation View

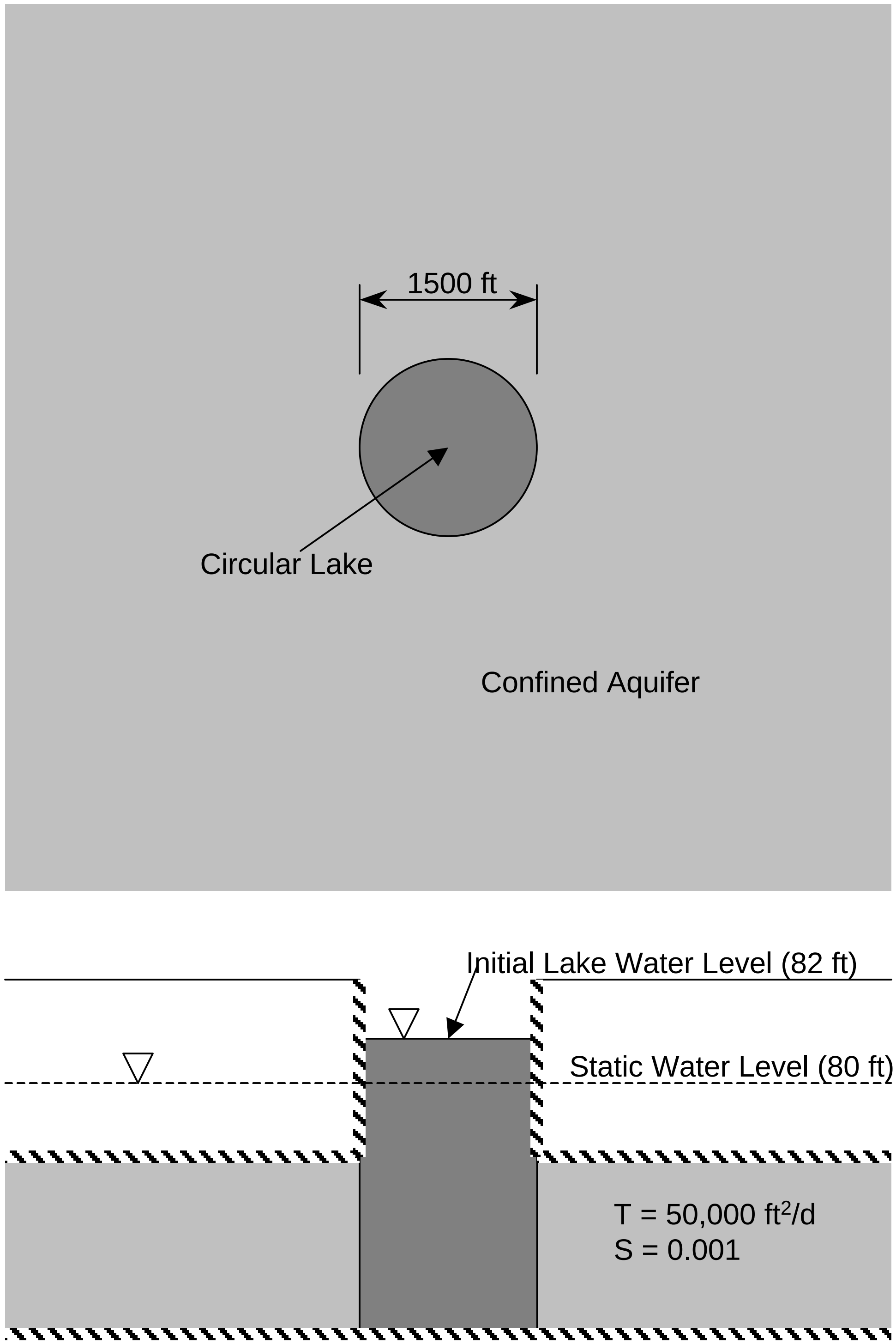


Figure 25. Circular Lake Problem Setup

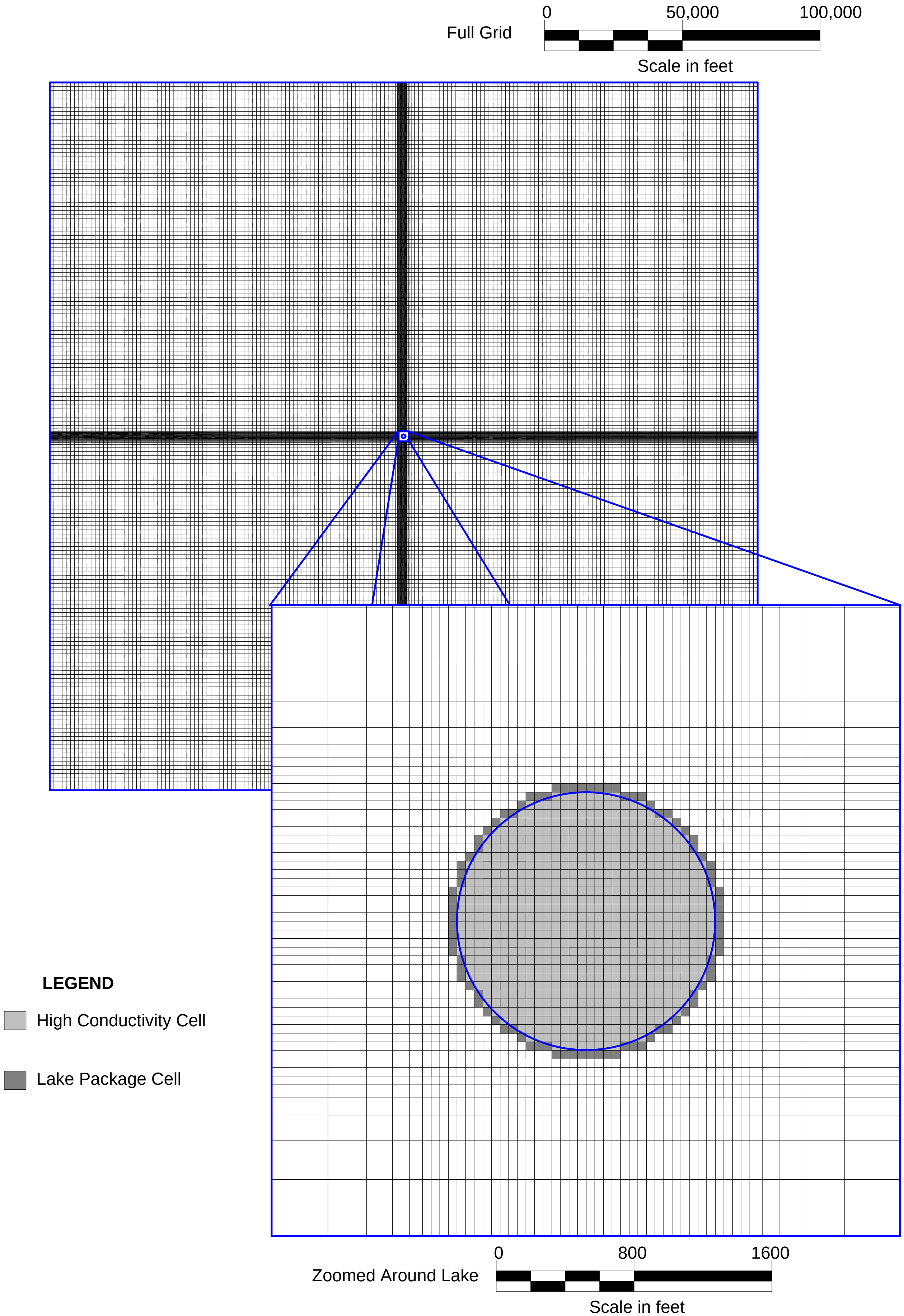


Figure 26. MODFLOW Model Setup for the Circular Lake Problem

width divided by half cell width), and an area of zero (vertical wall connections). A dummy node was also included for the lake bottom, having zero conductance and an area of 1,790,000 ft^2 (equal to the area of the 716 cells that were inactive in the model). The dummy node is necessary to give the lake the correct total area and allow for accurate computation of lake volume and stage changes. The lake was simulated in transient mode with an initial stage of 82 ft and no external stresses. Time stepping was the same as in the previously-described simulation. The Lake Package was set up to save the lake stage at each time step to an output file.

As shown in Figure 27, the three methods for calculating stage vs. time produced virtually identical results. Small discrepancies at late times are probably due to the influence of the no-flow boundary at the edge of the model (simulations with a smaller domain produced larger errors at late times).

This test problem shows that the Lake Package is able to match an analytical solution for transient stage vs. time, and is able to match another method for simulating lakes — the method of high conductivity, high storage lake cells. This problem also demonstrates that the transient stage solver is functional and that the groundwater boundary condition is appropriately formulated.

5.2 Verification of Lake Package Results Using the River Package, Drain Package, and External Calculations

In providing MODFLOW boundary conditions, the Lake Package is designed to behave like either the River Package (for wetted lake cells) or the Drain Package (for shore cells). Those packages can be used to test Lake Package functionality. Additionally, the volumetric budget calculations of the Lake Package can be checked externally using a calculator or spreadsheet program.

A simple test problem involving a hypothetical reservoir (Fish Lake) with one inflowing stream and one outflowing stream was designed to test many of the Lake Package functions. The problem, depicted in Figure 28, was simulated using the Lake Package in some simulations, and using the River and/or Drain Package in others. The model has 1 layer, 36 rows and 27 columns, with uniform grid spacing of 400 ft. The unconfined aquifer is assumed to have a constant transmissivity (largely unaffected by head variations) of 9800 ft^2/d and a specific yield of 5%. The reservoir is shallow relative to the aquifer with a uniform 5-ft thick lakebed having a vertical conductivity of 1.0 ft/d. Flowing River, at the southern edge of the model, is simulated as a constant-head boundary at 75 ft, msl. The northern edge of the model is a constant-flux boundary condition of 280 ft^3/d per foot of aquifer width, simulated using the Well Package. The eastern and western model edges are no-flow boundaries. The two segments of Rolling Stream are simulated using the Stream Package. The stage of Upper Rolling Stream varies form 110 ft, msl at the northern model edge to 101 ft, msl at the northern edge of Fish Lake (slope 0.0025). The stage of Lower Rolling Stream varies from 94 ft, msl at Fish Lake to 77 ft at Flowing River

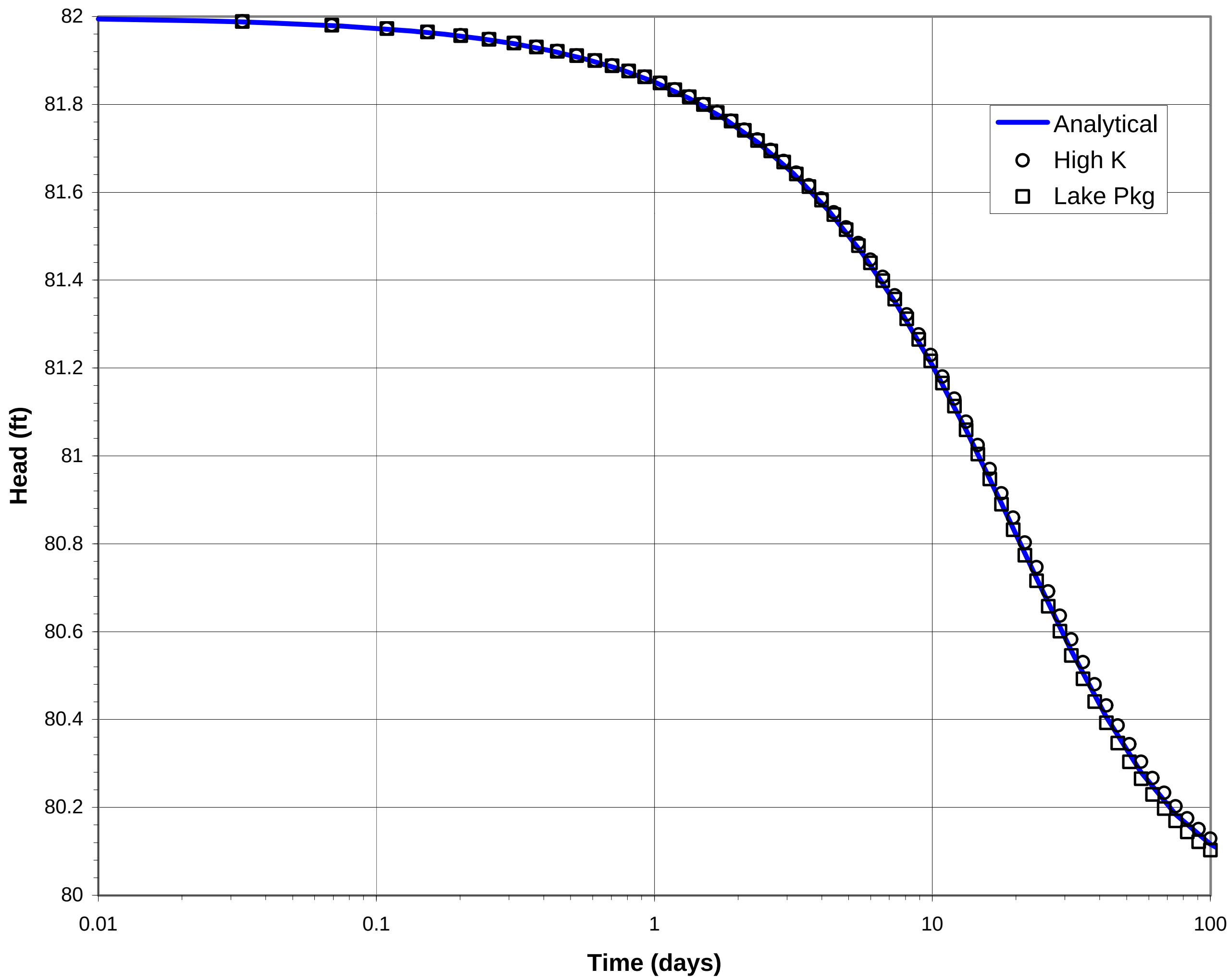


Figure 27. Stage vs. Time for the Circular Lake Problem, Comparison of Three Solutions

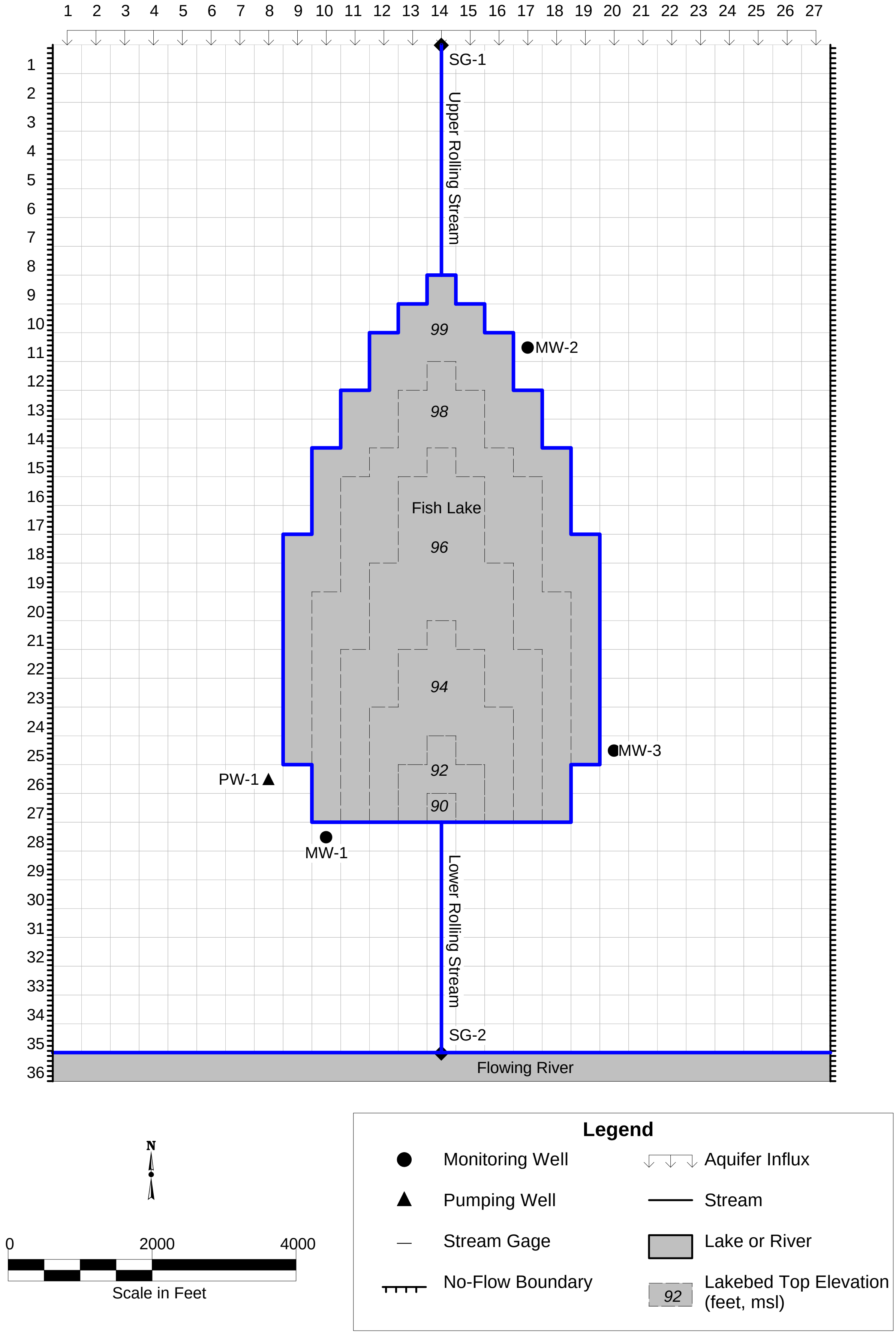


Figure 28. Model Design for Reservoir Test Problem

(slope 0.005). Both stream segments are 5 feet wide with a water depth of 2 ft and a uniform 2-ft thick streambed with a 1.0 ft/d conductivity. Flow into the northern end of Upper Rolling Stream is fixed at 302,400 ft^3/d, while flow into the northern end of Lower Rolling Stream is governed by the relationship: $Q = 2770\ H^{2.5}$, where Q is the outflow in ft^3/day, and H is the height of the water in Fish Lake over the outflow weir base elevation in feet. The weir at Fish Lake is set to a base elevation of 97 ft, msl. The areal precipitation rate is 0.002 ft/d, the evaporation rate is 0.001 ft/d, and the aquifer recharge rate is 0.001 ft/d. Overland runoff flows into the lake at a rate of 86,248 ft^3/d, and a nearby town withdraws 190,000 ft^3/d directly from the lake for water supply.

5.2.1 Steady-State Unstressed Model, Known Stage

In the first pair of simulations, the MODFLOW model is run in steady-state mode with a uniform initial head field. The lake stage is fixed at the known value of 99.9 ft, msl, which results in all lake cells being wetted. In one simulation, the Lake Package is used with Fish Lake simulated in constant stage mode (*ISIMMODE* = 0). In another simulation, Fish Lake is simulated with the River Package (the Drain Package is not needed because there are no shore cells). When the Lake Package is used, the inflow to Lower Rolling Stream is calculated by the Lake Package, and is not needed in the Stream Package input file. For the River Package simulation, however, the inflow to Lower Rolling Stream must be specified in the Stream Package input file. Based on the lake stage of 99.9 ft, msl, the flow to Lower Rolling Stream is 39671.1 ft^3/d.

The results of the two simulations are virtually identical. A side-by-side comparison of contoured aquifer head is shown in Figure 29. Cell-by-cell lake leakages and total lake flows were compared and found to be nearly identical. These simulations demonstrate that the Lake Package formulation of the MODFLOW boundary condition is essentially equivalent to the formulation of the River Package. Furthermore, an inspection of the main MODFLOW output file indicates that connections between the Stream Package and the Lake Package are working correctly. The outflow from the last reach of Upper Rolling Stream is reported as stream inflow in the volumetric budget of Fish Lake, and the stream outflow from Fish Lake is calculated to be 39671.1 ft^3/d, which appears also as the inflow to the first reach of Lower Rolling Stream.

5.2.2 Transient Stressed Model, Known Stage vs. Time

Additional simulations were made with the model in transient mode. In these simulations, a pumping well (PW-1) is simulated in row 26, column 8, pumping at 100,000 ft^3/d. As a result of this pumping, the aquifer heads decrease and the stage of Fish Lake decreases over a 4-year period. It is assumed, for now, that the stage of Fish Lake is a known (independent) variable, which can be interpolated from stage measurements made every three months (Table 18). The initial conditions for these simulations are taken from the results of the steady-state unstressed Lake Package simulation discussed above.

Table 18. Known (Measured) Stage of Fish Lake

Time	Measured Stage (ft, msl)
initial(0 months)	99.9
3 months	99.6818
6 months	99.4862
9 months	99.3192
12 months	99.1749
15 months	99.0492
18 months	98.9832
21 months	98.9118
24 months	98.8412
27 months	98.7808
30 months	98.7299
33 months	98.6871
36 months	98.6512

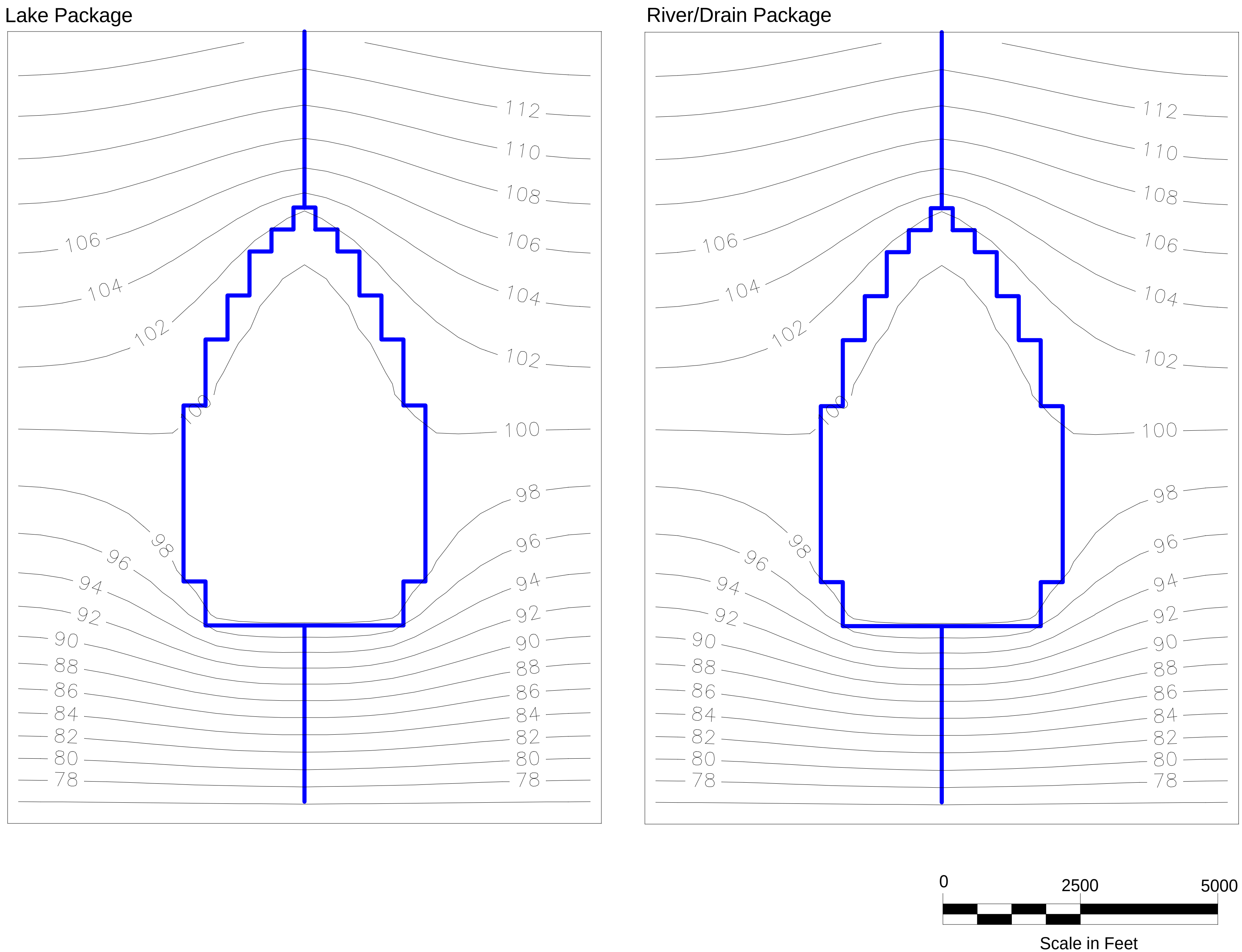


Figure 29. Comparison of Modeled Aquifer Head from the Lake Package and River/Drain Package

In the first simulation, the Lake Package is used with 12 3-month stress periods, each containing 12 uniform time steps. Fish Lake is simulated in interpolated stage mode (*ISIMMODE* = 0) using 99.9 ft, msl as the initial stage, and the values in Table 18 as the stages at the end of each period. Note that the stage falls below 99.0 ft, msl in this simulation, causing some cells along the edge of the lake to become shore cells during the simulation. The flow to Lower Rolling Stream is calculated by the Lake Package in this simulation.

In the second simulation, the River and Drain Packages are used with the same stress period and time step definitions. The final (measured) stage of the lake during each stress period is used as the stage for the entire stress period. When the stage is at or above 99.0 ft, msl, all of the cells are simulated with the River Package. Once the stage falls below 99.0, the cells that have a lakebed top elevation of 99.0 are then simulated with the Drain Package. This allows flow only from the groundwater to the lake at these shore cells, consistent with the assumptions of the Lake Package. The flow from Fish Lake to Lower Rolling Stream is calculated externally for each stress period and incorporated into the Stream Package input file.

A third simulation also uses the River and Drain Packages, but uses 144 stress periods, each with one time step, to simulate the same 3 year period. This allows the stage to vary more smoothly as with the first (Lake Package) simulation. The stage for each stress period is calculated externally using linear interpolation. The Drain Package is used for shore cells when the stage falls below 99.0 ft, msl. The stream outflow from Fish Lake is also externally computed for each stress period and incorporated into the Stream Package input file.

The first and third simulations produce nearly identical results for aquifer head, while the second simulation produces a less smooth solution. Plots of stage vs. time and head vs. time at three observation cells are shown in Figure 30. The solution of the second simulation is less smooth because the input stage vs. time function is less smooth. The agreement of the first and third simulations indicate again that the Lake Package boundary condition formulation is consistent with that of other MODFLOW packages, that the interpolated-stage calculation is functional, and that the Stream Package-Lake Package link is functional.

5.2.3 Steady-State Stressed Model, Unknown Final Equilibrium Stage

Additional simulations were made with the pumping well on and the model set to steady-state mode. In this mode, the final equilibrium conditions, or maximum impact conditions, can be determined. The initial conditions were again taken from the unstressed steady-state simulation using the Lake Package, discussed above.

In the first simulation, the Lake Package was set to run in steady-state mode, with the final stage of Fish Lake undetermined (dependent variable). All lake stresses were included in the Lake Package input file for proper volumetric accounting. The maximum number of stage-solver iterations was set to 50 and the stage convergence criterion was set to 1.0E-5 ft. This simulation resulted in a predicted equilibrium stage of 98.45 ft, msl.

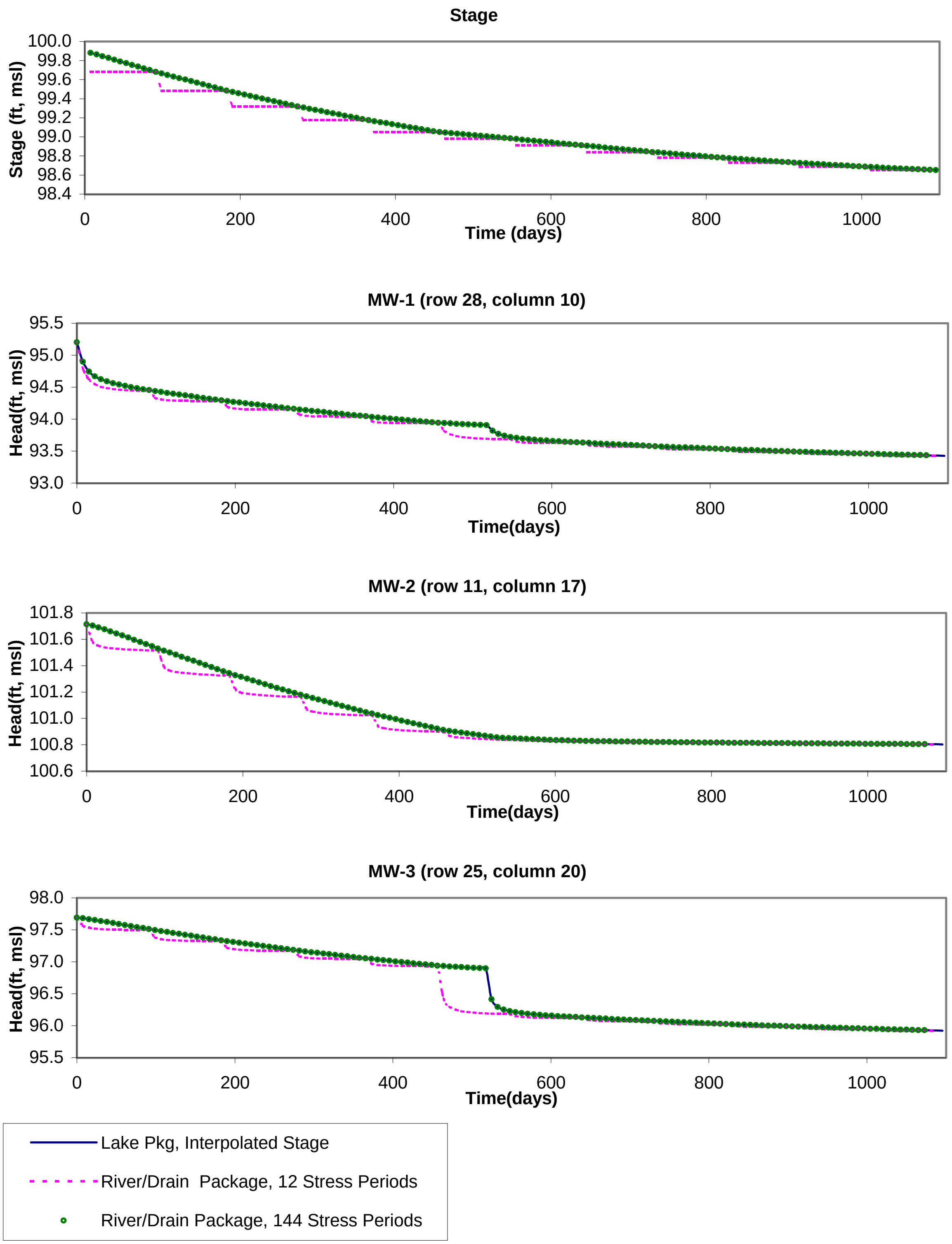


Figure 30. Stage of Fish Lake and Monitoring Well Heads for Interpolated Stage Runs

Three simulations were then made using the River and Drain Packages. In each simulation, the Drain Package was used only for the cells that have a lakebed top elevation of 99.0 ft, msl. The three simulations had slightly different values for the stage of Fish Lake: 98.45 ft, msl (same as that computed by the Lake Package), 98.55 ft, msl (0.1 ft higher stage), and 98.35 ft, msl (0.1 ft lower stage).

All simulations had acceptable aquifer mass balance errors (less than 0.02%). However, the Lake Package simulation required more MODFLOW iterations to achieve a solution. This is due to the fact that the steady-state stage solution mode adds nonlinearity to the aquifer system. Table 19 shows the volumetric balance for Fish Lake in each simulation, as reported in the output file (Lake Package simulation), or computed externally (River/Drain Package simulations). This indicates that the Lake Package steady-state stage solver did converge to the correct solution for stage, which could have also been obtained using trial-and-error with the River Package. Also, Figure 31 shows that the heads and drawdowns computed in the Lake Package simulation and the (correct stage) River/Drain Package simulation are basically identical.

5.2.4 Transient Stressed Model, Unknown Stage

Finally, a simulation was made in transient mode with the Lake Package, using *ISIMMODE* = 3 to simulate the transient change in stage due to pumping. The model was run for 30 years with 48 uniform time steps per year. The lake's stage and budget components were saved to an output file at each time step.

The resulting values of stage vs. time are shown in Figure 32. The stage declines smoothly to the final value of 98.45 ft, msl (within 0.015 of the stage predicted in the steady-state stressed model above). A small discontinuity in the plot occurs when the stage first drops below 99.0 ft, msl because the evaporation flux from the lake is abruptly reduced as the lake area is reduced, leading to a calculation of net inflow (and consequently a stage rise) for the next time step. As groundwater heads continue to decline, the lake stage also declines below 99.0 ft, eventually reaching an equilibrium state.

A spreadsheet was used to verify the calculation of stage change due to lake fluxes at each of the 1440 time steps. As shown in Table 20 (for the first 30 time steps), the modeled stage change agrees with the stage change calculated externally. This demonstrates that the transient stage solver is functioning correctly.

Table 19. Volumetric Budget Comparison for the Steady State Stressed Model Simulations.

	Steady State Stage Solution	Fixed Stage, River/Drain Package		
Stage	98.4489	98.3489	98.4489	98.5489
Precipitation	51520	51520	51520	51520
Runoff	86248	86248	86248	86248
Stream Inflow	319520	319422	319522	319614
Groundwater Seepage to Lake	242913	242323	242908	243612
Evaporation	-17920	-17920	-17920	-17920
Stream Outflow	-7000	-5854	-7000	-8271
Direct Withdrawal	-190000	-190000	-190000	-190000
Groundwater Seepage from Lake	-485281	-482290	-485307	-488427
Inflow	700201	699513	700197	700994
Outflow	-700201	-696064	-700227	-704618
Net Flow	0	3449	-30	-3624
Lake Mass Balance Error	0.00%	0.49%	0.00%	-0.52%
Aquifer Mass Balance Error	0.00%	0.00%	0.00%	0.00%

Table 20. Verification of the Stage Change Calculation for the Lake Package Transient Stage Solver

Time Step	Ending Time (d)	Starting Stage (ft, msl)	Wetted Area (ft^2)	Modeled Net Flux to Lake (ft^3/d)	Modeled Change in Stage (ft)	Calculated Change in Volume (ft^3)	Calculated Change in Stage (ft)
1	7.6	99.9000	25760000	-49349	-0.0146	-375515	-0.0146
2	15.2	99.8854	25760000	-59582	-0.0176	-453382	-0.0176
3	22.8	99.8678	25760000	-62907	-0.0186	-478683	-0.0186
4	30.4	99.8492	25760000	-64169	-0.0189	-488286	-0.0190
5	38.0	99.8303	25760000	-64540	-0.0191	-491109	-0.0191
6	45.7	99.8112	25760000	-64428	-0.0190	-490257	-0.0190
7	53.3	99.7922	25760000	-64023	-0.0189	-487175	-0.0189
8	60.9	99.7733	25760000	-63441	-0.0188	-482746	-0.0187
9	68.5	99.7545	25760000	-62742	-0.0185	-477427	-0.0185
10	76.1	99.7360	25760000	-61972	-0.0183	-471568	-0.0183
11	83.7	99.7177	25760000	-61166	-0.0181	-465435	-0.0181
12	91.3	99.6996	25760000	-60335	-0.0178	-459112	-0.0178
13	98.9	99.6818	25760000	-59502	-0.0176	-452773	-0.0176
14	106.5	99.6642	25760000	-58669	-0.0173	-446434	-0.0173
15	114.1	99.6469	25760000	-57847	-0.0171	-440180	-0.0171
16	121.8	99.6298	25760000	-57035	-0.0168	-434001	-0.0168
17	129.4	99.6130	25760000	-56243	-0.0167	-427974	-0.0166
18	137.0	99.5963	25760000	-55464	-0.0163	-422046	-0.0164
19	144.6	99.5800	25760000	-54703	-0.0162	-416256	-0.0162
20	152.2	99.5638	25760000	-53957	-0.0159	-410579	-0.0159
21	159.8	99.5479	25760000	-53231	-0.0158	-405055	-0.0157
22	167.4	99.5321	25760000	-52519	-0.0155	-399637	-0.0155
23	175.0	99.5166	25760000	-51824	-0.0153	-394348	-0.0153
24	182.6	99.5013	25760000	-51142	-0.0151	-389159	-0.0151
25	190.2	99.4862	25760000	-50477	-0.0149	-384098	-0.0149
26	197.8	99.4713	25760000	-49824	-0.0147	-379130	-0.0147
27	205.5	99.4566	25760000	-49184	-0.0145	-374260	-0.0145
28	213.1	99.4421	25760000	-48561	-0.0144	-369519	-0.0143
29	220.7	99.4277	25760000	-47947	-0.0142	-364847	-0.0142
30	228.3	99.4135	25760000	-47344	-0.0139	-360258	-0.0140
31	235.9	99.3996	25760000	-46754	-0.0138	-355769	-0.0138
32	243.5	99.3858	25760000	-46174	-0.0137	-351355	-0.0136
33	251.1	99.3721	25760000	-45602	-0.0135	-347003	-0.0135
34	258.7	99.3586	25760000	-45047	-0.0133	-342780	-0.0133
35	266.3	99.3453	25760000	-44499	-0.0131	-338610	-0.0131
36	273.9	99.3322	25760000	-43957	-0.0130	-334485	-0.0130
37	281.5	99.3192	25760000	-43431	-0.0128	-330483	-0.0128
38	289.2	99.3064	25760000	-42909	-0.0127	-326511	-0.0127
39	296.8	99.2937	25760000	-42399	-0.0125	-322630	-0.0125
40	304.4	99.2812	25760000	-41893	-0.0124	-318780	-0.0124

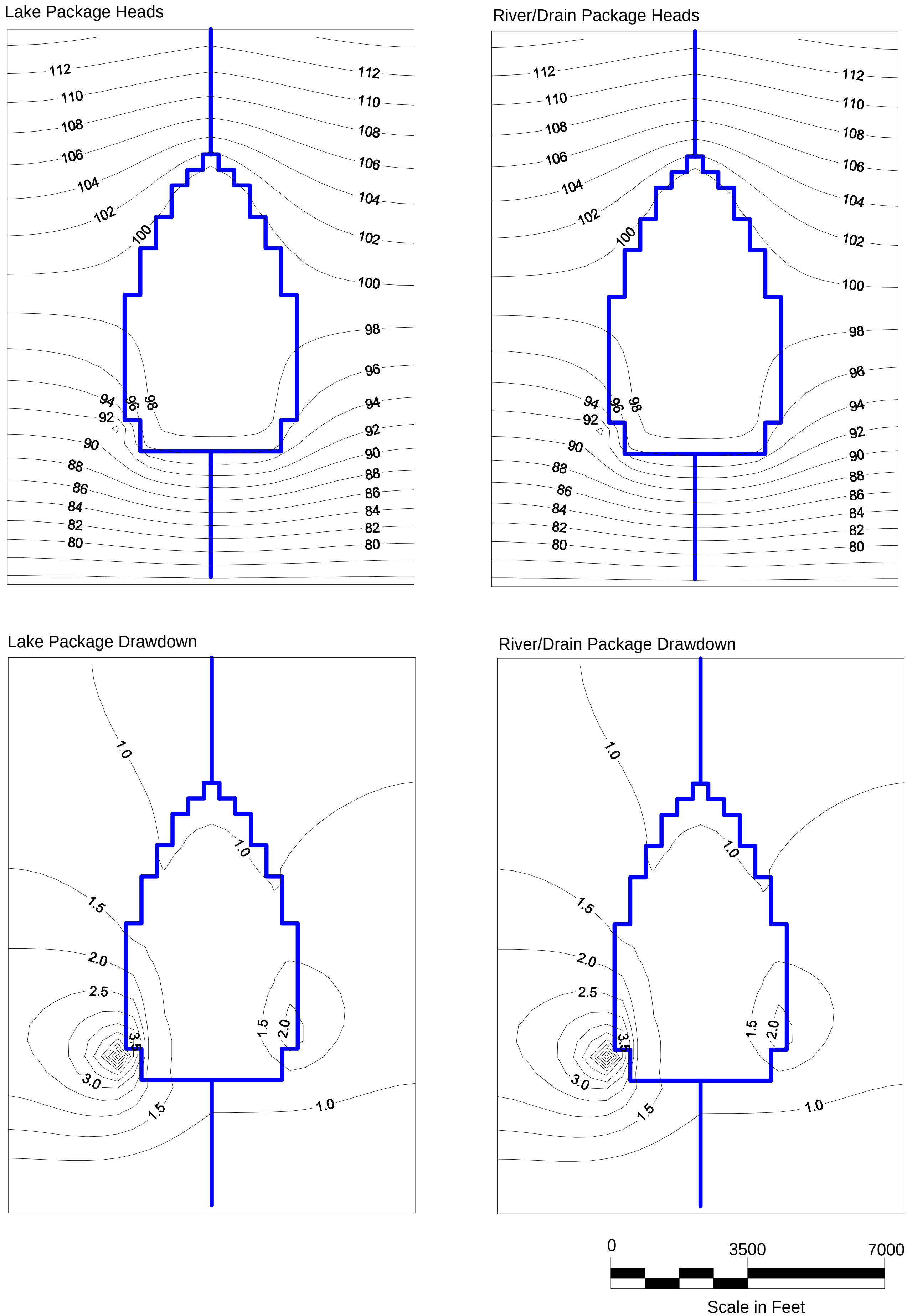


Figure 31. Comparison of Modeled Aquifer Head and Drawdowns from the Lake Package and River/Drain Package

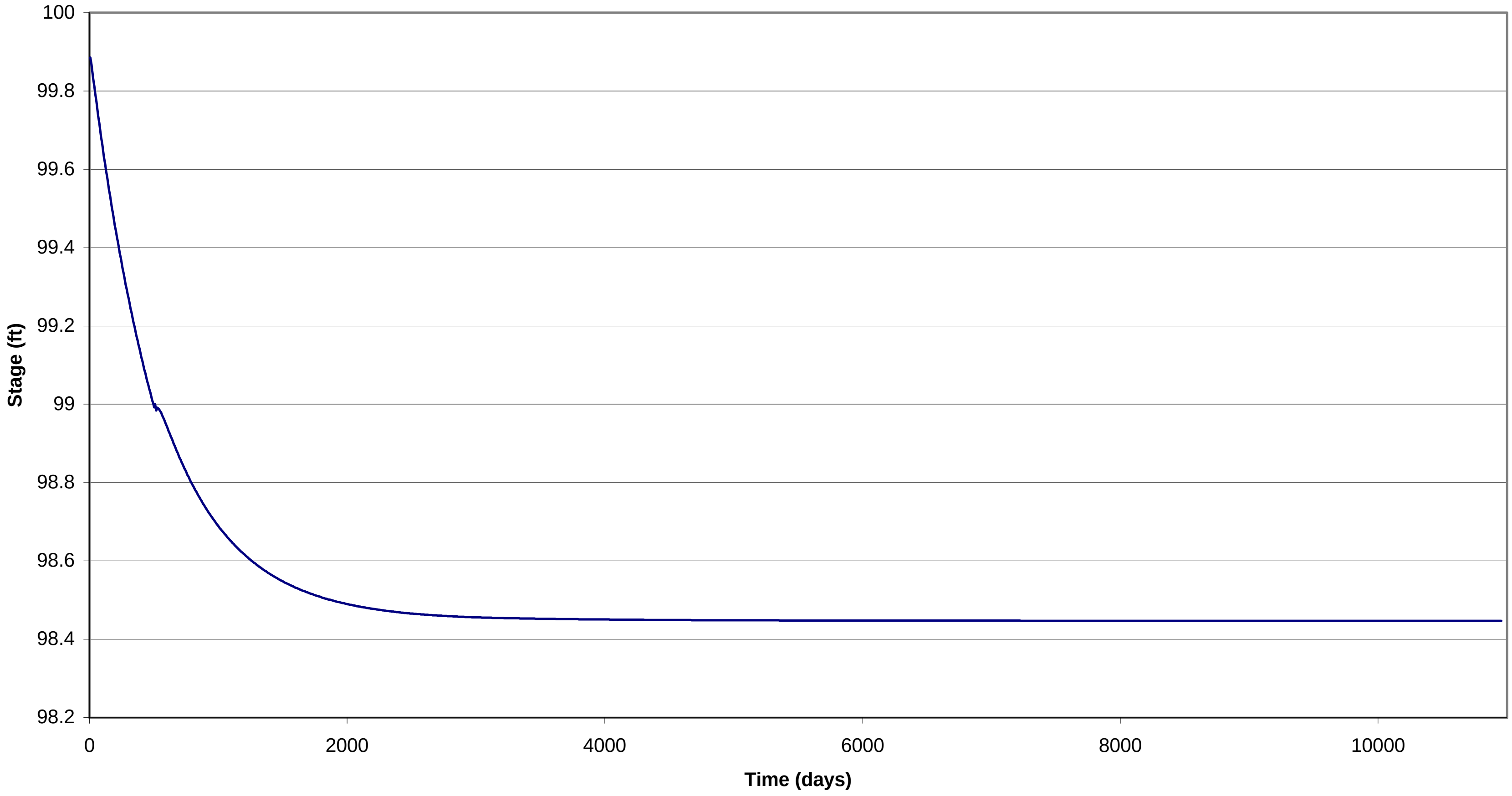


Figure 32. Stage of Fish lake as predicted by the lake package over a thirty year interval.

References

Cheng, X. and M.P. Anderson, 1993. Numerical simulation of ground-water interaction with lakes allowing for fluctuating lake levels. Ground Water. v. 31:6, pp. 929-933.

Cheng, X., 1994. Numerical Analysis of Groundwater and Lake Systems with Application to the Trout River Basin, Vilas County, Wisconsin. Ph.D. Thesis, University of Wisconsin-Madison.

Cooper, H..H., J.D. Bredehoeft, and I.S. Papadopulos, 1967. Response of a finite-diameter well to an instantaneous charge of water, Water Resources Research, Vol. 3, No. 1, pp. 263-269.

Fenske, J.P., S.A. Leake, and D.E. Prudic, 1996. Documentation of a computer program (RES1) to simulate leakage from reservoirs using the modular finite-difference ground-water flow model (MODFLOW), US Geological Survey Open-File Report 96-364.

Harbaugh, A.W., and M.G. McDonald, 1996. User's documentation for MODFLOW-96, an update to the U.S. Geological Survey modular finite-difference ground-water flow model, US geological Survey Open-file report 96-485.

McDonald, M.G. and A.W. Harbaugh, 1988. A Modular Three-Dimensional Finite-Difference Ground-Water Flow Model. Book 6, Ch. A1 of Techniques of Water-Resources Investigations of the U.S. Geological Survey.

Nair, S.K., and M.M. Wilsnack, 1998. A Comparison of Two Approaches to Simulating Lake-Ground Water Interactions with MODFLOW, Proceedings for MODFLOW '98, Colorado School of Mines, October 4-8, 1998, Vol. 2, pp. 871-878.

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

Yan, J., and K.R. Smith, 1994. Simulation of integrated surface water and ground water systems — model formulation, *Water Resources Bulletin*, Vol. 30, No. 5, pp. 879-890.

Appendix A. LAK2 Source Code

```
C*********************************************************************
      SUBROUTINE LAK2AL(ISUM,LENX,LCILAKE,LCRLAKE,LCRLNODE,LCISTRIN,
     &                  LCISTROUT,LCRATEQ,NLAKES,ILKCBC,ILKOUT,IECHO,
     &                  NSUBSTEPS,MXNODE,MXSTRIN,MXSTROUT,MXRATEQ,
     &                  IN,IOUT)

C       ----------------------------------------------------------------
C       SUBROUTINE TO READ INPUT FILE & SET UP X SUBARRAYS
C       ----------------------------------------------------------------

        IMPLICIT NONE
        INTEGER ISUM,LENX,LCILAKE,LCRLAKE,LCRLNODE,LCISTRIN,
     &          LCISTROUT,LCRATEQ,NLAKES,ILKCBC,ILKOUT,IECHO,
     &          NSUBSTEPS,MXNODE,MXSTRIN,MXSTROUT,MXRATEQ,
     &          IN,IOUT, K, NODES, NSTRIN, NSTROUT, NRATEQ, K1, K2,
     &          ISP, ISPT
        CHARACTER*8 LAKID
        LOGICAL ISOPEN

C       IDENTIFY PACKAGE
        WRITE (IOUT,1) IN
    1   FORMAT(' LAK2 -- LAKE PACKAGE VERSION 2.2 5/5/96',
     &         ' INPUT READ FROM UNIT',I5)

C       READ HEADER & FIRST LINE & PRINT
        READ(IN,'(A8)') LAKID
        IF(LAKID.NE.'/*LAK2.2') THEN
          PRINT *,' ERROR:  UNIT',IN,' IS NOT A LAK2.2 INPUT FILE'
          STOP
        ENDIF
        READ(IN,2) NLAKES, ILKCBC, ILKOUT, IECHO, NSUBSTEPS
    2   FORMAT(5I10)

        WRITE(IOUT,3) NLAKES
        IF(NLAKES.LE.0) THEN
          WRITE(IOUT,11)
          IN=0
          RETURN
        ENDIF

        IF(ILKCBC.GT.0) THEN
          WRITE(IOUT,4) ILKCBC
        ELSE
          WRITE(IOUT,5)
        ENDIF

        IF(ILKOUT.GT.0) THEN
          WRITE(IOUT,6) ILKOUT
          INQUIRE(ILKOUT, OPENED=ISOPEN)
          IF (.NOT. ISOPEN) THEN
            WRITE(IOUT,26) ILKOUT
            STOP
          ENDIF
          WRITE(ILKOUT,25)
        ENDIF

        IF(IECHO.LT.0) THEN
          WRITE(IOUT,7)
        ELSE IF(IECHO.EQ.0) THEN
          WRITE(IOUT,8)
        ELSE
```

```
          WRITE(IOUT,9)
        ENDIF
        IF(NSUBSTEPS.LE.1) THEN
          NSUBSTEPS=1
        ELSE
          WRITE(IOUT,10) NSUBSTEPS
        ENDIF
    3   FORMAT(5X,I6,' LAKES IN THIS SIMULATION')
   11   FORMAT(5X,'NO LAKES -- LAK2 IS NOT ACTIVE',/)
    4   FORMAT(5X,'CELL-BY-CELL FLOWS WILL BE SAVED ON UNIT',I5)
    5   FORMAT(5X,'CELL-BY-CELL FLOWS WILL NOT BE SAVED')
    6   FORMAT(5X,'STAGE/RATE RECORDS WILL BE WRITTEN TO UNIT',I5)
    7   FORMAT(5X,'NO ECHOING OF LAKE INPUTS')
    8   FORMAT(5X,'SUMMARY ECHOING OF LAKE INPUTS')
    9   FORMAT(5X,'FULL ECHOING OF LAKE INPUTS')
   10   FORMAT(5X,'NUMBER OF SUB-TIME-STEPS FOR TRANSIENT STAGE',
     &         'UPDATING:',I5)
   25   FORMAT('LAKE_NAME        TIME          STAGE         PRECIP',
     &  '           EVAP         RUNOFF      STREAM_IN     STREAM_OUT',
     &  '        SEEPAGE           AREA         VOLUME      NEW_STAGE')
   26   FORMAT(' NO FILE CONNECTED TO UNIT',I5,' -- STOPPING.')
C       INITIALIZE TOTALS
        MXNODE=0
        MXSTRIN=0
        MXSTROUT=0
        MXRATEQ=0
C       SKIP REST OF SIMULATION DATA SECTION
        DO 100 K=1,NLAKES
  100     READ(IN,*)

C       SCAN THE PHYSICAL DATA SECTION, ACCUMULATING TOTAL NODES, ETC.
        DO 600 K=1,NLAKES

C         ACCUMULATE NODES, NSTRIN, NSTROUT
          READ(IN,12) NODES, NSTRIN, NSTROUT
   12     FORMAT(3I10)
          MXNODE=MXNODE+ NODES
          MXSTRIN=MXSTRIN+ NSTRIN
          MXSTROUT=MXSTROUT+ NSTROUT

C         READ INFO FOR LAKE, ACCUMULATE NRATEQ
          IF(NSTRIN.GT.0) THEN
            DO 200 K1=1, NSTRIN
  200         READ(IN,*)
          ENDIF
          IF(NSTROUT.GT.0) THEN
            DO 400 K1=1, NSTROUT
              READ(IN,13) NRATEQ
              MXRATEQ=MXRATEQ+ NRATEQ
              DO 300 K2=1, NRATEQ
  300           READ(IN,*)
  400       CONTINUE
          ENDIF
          DO 500 K1=1,NODES
  500       READ(IN,*)
  600   CONTINUE
   13   FORMAT(10X,I10)

C       REWIND & REPOSITION
        REWIND(IN)
        READ(IN,*)
        READ(IN,*)

C       SET ADDRESSES
        LCISTRIN=ISUM
```

```
        ISP= MXSTRIN
        ISPT=ISPT+ ISP
        ISUM=ISUM+ ISP

        LCISTROUT=ISUM
        ISP= MXSTROUT*3
        ISPT=ISPT+ ISP
        ISUM=ISUM+ ISP

        LCRATEQ=ISUM
        ISP= MXRATEQ*4
        ISPT=ISPT+ ISP
        ISUM=ISUM+ ISP

        LCILAKE=ISUM
        ISP= NLAKES*9
        ISPT=ISP
        ISUM=ISUM+ ISP

        LCRLAKE=ISUM
        ISP= NLAKES*9*2
        ISPT=ISPT+ ISP
        ISUM=ISUM+ ISP

        LCRLNODE=ISUM
        ISP= MXNODE*5
        ISPT=ISPT+ ISP
        ISUM=ISUM+ ISP

C       AVOID DIMENSION-ZERO VARIABLES IN SUBROUTINES
        IF(MXSTRIN.EQ.0) MXSTRIN = 1
        IF(MXSTROUT.EQ.0) MXSTROUT = 1
        IF(MXRATEQ.EQ.0) MXRATEQ = 1

C       PRINT SPACE USED & CHECK BOUNDS
        WRITE(IOUT,14) ISPT
   14   FORMAT(I10,' ELEMENTS OF X ARRAY ARE USED FOR LAKES')
        ISPT=ISUM-1
        WRITE(IOUT,15) ISPT, LENX
   15   FORMAT(I10,' ELEMENTS OF X ARRAY USED OUT OF', I10)
        IF(ISPT.GT.LENX) WRITE(IOUT,16)
   16   FORMAT('   ***** X ARRAY MUST BE DIMENSIONED LARGER *****')
        WRITE(IOUT,*)

        RETURN
      END
C     END OF SUBROUTINE LAK2AL
C
C*********************************************************************
      SUBROUTINE LAK2RP1(ILAKE,RLAKE,RLNODE,ISTRIN,ISTROUT,RATEQ,
     &                   LAKENAME,NLAKES,MXNODE,MXSTRIN,MXSTROUT,
     &                   MXRATEQ,IECHO,IN,IOUT,NLAY,NROW,NCOL)

C       ------------------------------------------------------------
C       SUBROUTINE TO READ SIMULATION & PHYSICAL LAKE DATA
C       ------------------------------------------------------------

        IMPLICIT NONE
        INTEGER NLAKES,MXNODE,MXSTRIN,MXSTROUT,MXRATEQ,IECHO,IN,IOUT,
     &          ILAKE(9,NLAKES),ISTRIN(MXSTRIN),ISTROUT(3,MXSTROUT),
     &          NLAY,NROW,NCOL
        DOUBLE PRECISION RLAKE(9,NLAKES), STAGE, CONVCRIT, STAGEMX,
     &          STAGEMN, TOTAREA
        REAL RLNODE(5,MXNODE),RATEQ(4,MXRATEQ)
        CHARACTER*10 LAKENAME(NLAKES), LNAME
```

```
      INTEGER K, K1, K2, ITERLAKE, NODES, NSTRIN, NSTROUT, ICONDOP,
     &          ISEG, NRATEQ, ILAY, IROW, ICOL
      REAL CUTOFF, CONST, ELEV,
     &       EXPNT, TOP, BOT, AREA, COND, TOTCOND
      INTEGER LRLNODE, LISTRIN, LISTROUT, LRATEQ, ISIMMODE

C       INITAILAIZE ARRAY LOCATORS
      LRLNODE=1
      LISTRIN=1
      LISTROUT=1
      LRATEQ=1

C       READ SIMULATION DATA FOR EACH LAKE
      IF(IECHO.GE.0) WRITE(IOUT,2)
      DO 100 K=1,NLAKES
        READ(IN,1) LNAME, ISIMMODE, STAGE, ITERLAKE, CONVCRIT
        IF(LNAME.EQ.' ') WRITE(LNAME,'(I10)') K
        LAKENAME(K)=LNAME
        ILAKE(7,K)=ISIMMODE
        ILAKE(8,K)=ITERLAKE
        RLAKE(1,K)=STAGE
        RLAKE(8,K)=CONVCRIT

        IF(IECHO.LT.0) GOTO 100
        IF(ISIMMODE.EQ.1) THEN
          WRITE(IOUT,3) LNAME,STAGE
        ELSE IF(ISIMMODE.EQ.2) THEN
          WRITE(IOUT,4) LNAME,STAGE,ITERLAKE,CONVCRIT
        ELSE IF(ISIMMODE.EQ.3) THEN
          WRITE(IOUT,5) LNAME,STAGE,ITERLAKE,CONVCRIT
        ELSE
          WRITE(IOUT,6) LNAME,STAGE
        ENDIF
  100 CONTINUE
    1 FORMAT(A10,I10,F10.0,I10,F10.0)
    2 FORMAT(/42X,'----STAGE SOLVER----',/,' LAKE NAME   SIM MODE',7X,
     &        'START STAGE   MAX ITER   TOLERANCE',/,1X,65('-'))
    3 FORMAT(1X,A10,'  INTERP STAGE',1P,G15.6,7X,'N/A   N/A')
    4 FORMAT(1X,A10,'  STEADY STATE',1P,G15.6,I10,G15.6)
    5 FORMAT(1X,A10,'  TRANSIENT   ',1P,G15.6,I10,G15.6)
    6 FORMAT(1X,A10,'  SET STAGE   ',1P,G15.6,7X,'N/A   N/A')

C       READ PHYSICAL DATA FOR EACH LAKE
      DO 600 K=1,NLAKES

C         READ LAKE HEADER & SET ARRAY LOCATORS
        READ(IN,7) NODES,NSTRIN,NSTROUT,STAGEMX,ICONDOP
        ILAKE(1,K)=NODES
        ILAKE(2,K)=NSTRIN
        ILAKE(3,K)=NSTROUT
        ILAKE(4,K)=LRLNODE
        ILAKE(5,K)=LISTRIN
        ILAKE(6,K)=LISTROUT
        RLAKE(2,K)=STAGEMX
    7   FORMAT(3I10,F10.0,I10)

C         READ STREAM INFLOW SEGMENT(S)
        IF(NSTRIN.GT.0) THEN
          IF(IECHO.GT.0) WRITE(IOUT,9) LAKENAME(K)
          DO 200 K1=1,NSTRIN
            READ(IN,8) ISEG
            IF(IECHO.GT.0) WRITE(IOUT,10) K1,ISEG
            ISTRIN(LISTRIN)=ISEG
            IF(LISTRIN.LT.MXSTRIN) LISTRIN=LISTRIN+1
  200     CONTINUE
```

```
      ENDIF
    8 FORMAT(I10)
    9 FORMAT(/,' LAKE INFLOW STREAMS FOR ',A10,':')
   10 FORMAT('   INFLOW NO.',I4,' = STREAM SEGMENT',I5)

C     READ STREAM OUTFLOW SEGMENT(S) & RATING EQUATIONS
      IF(NSTROUT.GT.0) THEN
        IF(IECHO.GT.0) WRITE(IOUT,12) LAKENAME(K)
        DO 400 K1=1,NSTROUT
          READ(IN,11) ISEG, NRATEQ
          IF(IECHO.GT.0) WRITE(IOUT,13) K1, ISEG, NRATEQ
          ISTROUT(1,LISTROUT)=ISEG
          ISTROUT(2,LISTROUT)=NRATEQ
          ISTROUT(3,LISTROUT)=LRATEQ
          IF(LISTROUT.LT.MXSTROUT) LISTROUT=LISTROUT+1

          DO 300 K2=1,NRATEQ
            READ(IN,14) CUTOFF, CONST, ELEV, EXPNT
            IF(IECHO.GT.0) WRITE(IOUT,30) CONST, ELEV, EXPNT, CUTOFF
            RATEQ(1,LRATEQ)= CUTOFF
            RATEQ(2,LRATEQ)= CONST
            RATEQ(3,LRATEQ)= ELEV
            RATEQ(4,LRATEQ)= EXPNT
            IF(LRATEQ.LT.MXRATEQ) LRATEQ=LRATEQ+1
  300     CONTINUE
  400   CONTINUE
      ENDIF
   11 FORMAT(2I10)
   12 FORMAT(/,' LAKE OUTFLOW STREAMS FOR ',A10,':')
   13 FORMAT('   OUTFLOW NO.',I4,' = STREAM SEGMENT',I5,
     &       ' WITH A',I4,'-PART RATING EQUATION:')
   14 FORMAT(4F10.0)
   30 FORMAT(5X,'OUTFLOW = ',1P,G13.6,'*(STAGE-',G13.6,')**',G13.6,
     &       '  (ABOVE ',G13.6,')')

C     READ NODE INFORMATION
      IF(IECHO.GT.0) WRITE(IOUT,15) LAKENAME(K)
      STAGEMN=1.0E30
      TOTAREA=0.0
      TOTCOND=0.0
      DO 500 K1=1,NODES
        READ(IN,16) ILAY,IROW,ICOL,TOP,BOT,AREA,COND
        IF(ICONDOP.NE.1) COND=COND*AREA/(TOP-BOT)
        IF(IECHO.GT.0) WRITE(IOUT,17) K1,ILAY,IROW,ICOL,TOP,BOT,
     &                 AREA,COND
        CALL SETPOS(RLNODE(1,LRLNODE),ILAY,IROW,ICOL,NLAY,NROW,NCOL)
        RLNODE(2,LRLNODE)= TOP
        RLNODE(3,LRLNODE)= BOT
        RLNODE(4,LRLNODE)= AREA
        RLNODE(5,LRLNODE)= COND
        LRLNODE=LRLNODE+1
        IF(TOP.LT.STAGEMN) STAGEMN=TOP
        TOTAREA=TOTAREA+AREA
        TOTCOND=TOTCOND+COND
  500 CONTINUE
   15 FORMAT(/' LAKE NODES FOR ',A10,':',/,'  NO.  LAY  ROW  COL',
     &       8X,'TOP',12X,'BOT',11X,'AREA',6X,'CONDUCTANCE',/,1X,
     &       79('-'))
   16 FORMAT(3I10,4F10.0)
   17 FORMAT(4I5,1P,4G15.6)

C     SET STAGEMN & TOTAREA, INITIALIZE STRESS VARIABLES
      RLAKE(3,K)= STAGEMN
      RLAKE(4,K)=0.0
      RLAKE(5,K)=0.0
```

```
          RLAKE(6,K)=0.0
          RLAKE(7,K)=0.0
          RLAKE(9,K)= TOTAREA

C         PRINT SUMMARY INFO
          IF(IECHO.GE.0) THEN
            WRITE(IOUT,18) LAKENAME(K)
            WRITE(IOUT,19) NODES
            WRITE(IOUT,20) NSTRIN
            WRITE(IOUT,21) NSTROUT
            WRITE(IOUT,22) STAGEMX
            WRITE(IOUT,23) STAGEMN
            WRITE(IOUT,24) TOTAREA
            WRITE(IOUT,25) TOTCOND
          ENDIF
   18     FORMAT(/,' PHYSICAL DATA SUMMARY FOR ',A10,':')
   19     FORMAT(3X,I7,' NODES')
   20     FORMAT(3X,I7,' INFLOW STREAMS')
   21     FORMAT(3X,I7,' OUTFLOW STREAMS')
   22     FORMAT('       MAXIMUM STAGE=',1P,G13.6)
   23     FORMAT('       MINIMUM STAGE=',1P,G13.6)
   24     FORMAT('          TOTAL AREA=',1P,G13.6)
   25     FORMAT('   TOTAL CONDUCTANCE=',1P,G13.6)

  600   CONTINUE
C       END OF LAKE LOOP (PHYSICAL DATA)

        RETURN
      END
C     END OF SUBROUTINE LAK2RP1
C
C*********************************************************************
      SUBROUTINE LAK2RP2(ILAKE,RLAKE,LAKENAME,NLAKES,IN,IOUT,PERLEN)

C       ----------------------------------------------------------------
C       SUBROUTINE TO READ & PREPARE LAKE STRESS PERIOD DATA
C       ----------------------------------------------------------------

        IMPLICIT NONE
        INTEGER NLAKES, IN, IOUT, ILAKE(9,NLAKES), ITMP, K, IOUTOP
        DOUBLE PRECISION RLAKE(9,NLAKES), DSDT,
     &          PRECIP, EVAP, RUNOFF, DRYRCH, STAGE
        REAL PERLEN
        CHARACTER*10 LAKENAME(NLAKES)

C       READ FLAG & PRINT
        READ(IN,1) ITMP
    1   FORMAT(I10)

C       CHECK FOR REUSE OF LAST STRESS INFO, SET DS/DT = 0
        IF(ITMP.LT.0) THEN
          WRITE(IOUT,2)
          DO 100 K=1,NLAKES
  100       IF(ILAKE(7,K).EQ.1) RLAKE(8,K)=0.0
          RETURN
        ENDIF
    2   FORMAT(/' REUSING LAKE STRESS INFORMATION FROM LAST STRESS',
     &         ' PERIOD')

C       READ STRESSES FOR EACH LAKE, PRINT,  & STORE
        WRITE(IOUT,4)
        DO 200  K=1,NLAKES
          READ(IN,3) PRECIP, EVAP, RUNOFF, DRYRCH, IOUTOP, STAGE
          RLAKE(4,K)=PRECIP
          RLAKE(5,K)=EVAP
```

```
          RLAKE(6,K)=RUNOFF
          RLAKE(7,K)=DRYRCH
          ILAKE(9,K)=IOUTOP
          ITMP=ILAKE(7,K)
          IF(ITMP.EQ.1) THEN
            WRITE(IOUT,5) LAKENAME(K),PRECIP,EVAP,RUNOFF,DRYRCH,STAGE
            DSDT=(STAGE-RLAKE(1,K))/PERLEN
            RLAKE(8,K)=DSDT
          ELSE IF(ITMP.EQ.2.OR.ITMP.EQ.3) THEN
            WRITE(IOUT,6) LAKENAME(K),PRECIP, EVAP, RUNOFF, DRYRCH
          ELSE
            WRITE(IOUT,7) LAKENAME(K),PRECIP,EVAP,RUNOFF,DRYRCH,STAGE
            RLAKE(1,K)=STAGE
          ENDIF
  200   CONTINUE
        WRITE(IOUT,*)
    3   FORMAT(4F10.0,I10,F10.0)
    4   FORMAT(/,' LAKE STRESS INFORMATION:',//' LAKE NAME',6X,'PRECIP',
     &     11X,'EVAP',9X,'RUNOFF',7X,'DRY RECH',10X,'STAGE',/1X,95('-'))
    5   FORMAT(1X,A10,1P,5G15.6,' (FINAL)')
    6   FORMAT(1X,A10,1P,4G15.6,8X,'N/A')
    7   FORMAT(1X,A10,1P,5G15.6)

        RETURN
      END
C     END OF SUBROUTINE LAK2RP2
C
C***********************************************************************
      SUBROUTINE LAK2AD(ILAKE,RLAKE,ISTROUT,RATEQ,NLAKES,MXSTROUT,
     &                  MXRATEQ,ISTRM,STRM,NSTREM,DELT)

C       ---------------------------------------------------------------
C       SUBROUTINE TO SET INTERPOLATED STAGE & SET FLOW TO LAKE OUTFLOW
C         SEGMENTS
C       ---------------------------------------------------------------

        IMPLICIT NONE
        INTEGER NLAKES, MXSTROUT, MXRATEQ, NSTREM, ILAKE(9,NLAKES),
     &          ISTROUT(3,MXSTROUT), ISTRM, K, K1, NRATEQ, LISTROUT,
     &          LRATEQ, NSTROUT, ISEGOUT
        DOUBLE PRECISION RLAKE(9,NLAKES), STAGE
        REAL RATEQ(4,MXRATEQ), STRM, DELT, QSTROUT
        REAL DSCHRGE


        DO 200 K=1,NLAKES

C         UPDATE STAGE FOR ISMMODE=1
          IF(ILAKE(7,K).EQ.1) RLAKE(1,K)=RLAKE(1,K)+ RLAKE(8,K)*DELT
C         SET OUTFLOWS
          NSTROUT=ILAKE(3,K)

          IF(NSTROUT.GT.0) THEN
            LISTROUT=ILAKE(6,K)
            STAGE=RLAKE(1,K)

            DO 100 K1= 1, NSTROUT
              ISEGOUT= ISTROUT(1,LISTROUT)
              NRATEQ= ISTROUT(2,LISTROUT)
              LRATEQ= ISTROUT(3,LISTROUT)
              LISTROUT=LISTROUT+1
              QSTROUT= DSCHRGE(STAGE,RATEQ(1,LRATEQ),NRATEQ)
              CALL SETSTROUT(ISEGOUT,QSTROUT,ISTRM,STRM,NSTREM)
  100       CONTINUE
```

```
          ENDIF

  200   CONTINUE

        RETURN
      END
C     END OF SUBROUTINE LAK2AD
C
C*********************************************************************
      SUBROUTINE LAK2FM(ILAKE,RLAKE,RLNODE,ISTRIN,ISTROUT,RATEQ,NLAKES,
     &                  MXNODE,MXSTRIN,MXSTROUT,MXRATEQ,HEAD,IBOUND,
     &                  HCOF,RHS,NLAY,NROW,NCOL,ARTRIB,NSS,STRM,ISTRM,
     &                  NSTREM)

C       -----------------------------------------------------------------
C       SUBROUTINE TO FORMULATE THE LAKE BC PART OF THE GW EQUATIONS
C         ALSO CALLS SSSTAGE TO SOLVE FOR STEADY STATE STAGE
C       -----------------------------------------------------------------

        IMPLICIT NONE
        INTEGER NLAKES, MXNODE, MXSTRIN, MXSTROUT, MXRATEQ, NLAY, NROW,
     &          NCOL, NSS, NSTREM, ILAKE(9,NLAKES), ISTRIN(MXSTRIN),
     &          ISTROUT(3,MXSTROUT), IBOUND, ISTRM
        REAL RLNODE(5,MXNODE), RATEQ(4,MXRATEQ),
     &       HCOF, RHS, ARTRIB, STRM
        DOUBLE PRECISION HEAD, RLAKE(9, NLAKES), STAGE, DRYRCH
        INTEGER K, LRLNODE, NSTRIN, NSTROUT, LISTROUT, NODES, ISIMMODE,
     &          ITERLAKE, LISTRIN
        REAL DUM

C       STEP THROUGH LAKES, GET LAKE-WIDE VARIABLES
        DO 400 K=1,NLAKES
          NODES= ILAKE(1,K)
          LRLNODE= ILAKE(4,K)
          ISIMMODE= ILAKE(7,K)

C         CALL STEADY-STATE STAGE SOLVER FOR MODE=2
          IF(ISIMMODE.EQ.2) THEN
            NSTRIN= ILAKE(2,K)
            NSTROUT= ILAKE(3,K)
            LISTRIN= ILAKE(5,K)
            LISTROUT= ILAKE(6,K)
            ITERLAKE= ILAKE(8,K)
            STAGE = RLAKE(1,K)
            CALL SSSTAGE(RLAKE(1,K),RLNODE(1,LRLNODE),NODES,
     &                   ISTRIN(LISTRIN),NSTRIN,ISTROUT(1,LISTROUT),
     &                   NSTROUT,RATEQ,MXRATEQ,ITERLAKE,
     &                   IBOUND,HEAD,NLAY,NROW,NCOL,ARTRIB,NSS,
     &                   ISTRM,STRM,NSTREM)

C           DON'T UPDATE STAGE IF DIDN'T CHANGE MUCH
            IF(ABS(RLAKE(1,K)-STAGE).LT.RLAKE(8,K)) THEN
              RLAKE(1,K)=STAGE
c           ELSE
c             PRINT 77, k,stage,rlake(1,k)
c  77         FORMAT(' Lake',i3,' stage updated from',F10.4,' to',F10.4)
            ENDIF
          ENDIF

C         FORMULATE EQUATIONS
          STAGE=RLAKE(1,K)
          DRYRCH=RLAKE(7,K)
          CALL LAKECELL(STAGE,RLNODE(1,LRLNODE),NODES,'FM',NLAY,NROW,
     &                  NCOL,HEAD,IBOUND,RHS,HCOF,DUM,DUM,DUM,
     &                  DUM,DUM,0,0,DRYRCH)
```

```
  400   CONTINUE

        RETURN
      END
C     END OF SUBROUTINE LAK2FM
C
C***********************************************************************
      SUBROUTINE LAK2BD(ILAKE,RLAKE,RLNODE,ISTRIN,ISTROUT,RATEQ,NLAKES,
     &                 MXNODE,MXSTRIN,MXSTROUT,MXRATEQ,HEAD,IBOUND,BUFF,
     &                 NLAY,NROW,NCOL,ARTRIB,NSS,
     &                 IOUT,ILKCBC,ILKOUT,NSUBSTEPS,DELT,VBVL,VBNM,
     &                 MSUM,KSTP,KPER,ICBCFL,TOTIM,LAKENAME,ISTRM,STRM,
     &                 NSTREM)

C       ------------------------------------------------------------------
C       SUBROUTINE TO COMPUTE & WRITE THE BUDGET TERMS FOR LAKES,
C         ALSO COMPUTES TRANSIENT STAGE WITH TRSTAGE
C       ------------------------------------------------------------------

        IMPLICIT NONE
        INTEGER NLAKES, MXNODE, MXSTRIN, MXSTROUT, MXRATEQ, NLAY, NROW,
     &          NCOL, NSS, ILAKE(9,NLAKES), ISTRIN(MXSTRIN),
     &          ISTROUT(3,MXSTROUT), IBOUND, IOUT, ILKCBC,
     &          ILKOUT, NSUBSTEPS, MSUM, KSTP, KPER, ICBCFL,
     &          NSTREM, ISTRM
        REAL RLNODE(5,MXNODE), RATEQ(4,MXRATEQ),
     &       ARTRIB, BUFF(NCOL,NROW,NLAY), DELT, TOTIM,
     &       VBVL(4,20), STRM
        DOUBLE PRECISION HEAD, RLAKE(9,NLAKES), STAGE, STAGENW
        CHARACTER*10 LAKENAME(NLAKES)
        CHARACTER*4 VBNM(4,20), TEXT(4)
        INTEGER K, LRLNODE, NSTRIN, LISTRIN, NSTROUT, LISTROUT,
     &          NODES, ISIMMODE, I, J, IOUTOP, ITERLAKE
        REAL BUDGET(2,5), QNET, WETAR, VOL, DUM, QPRECIP, QEVAP,
     &          QRUNOFF, QSTRIN, QSTROUT, QSEEPAGE, DT, TOTIN,
     &          TOTOUT, PCTERROR
        LOGICAL CBCSAVE
        DATA TEXT/'    ','LAKE',' LEA','KAGE'/

C       SEE IF CBC OUTPUT IS NEEDED, INITIALIZE
        IF(ICBCFL.EQ.0.OR.ILKCBC.LE.0) THEN
          CBCSAVE=.FALSE.
        ELSE
          CBCSAVE=.TRUE.
          DO 90 K=1,NLAY
          DO 90 I=1,NROW
          DO 90 J=1,NCOL
   90       BUFF(J,I,K)=0.0
        ENDIF
        TOTIN=0.0
        TOTOUT=0.0

C       STEP THROUGH LAKES, SET VARIABLES
        DO 500 K=1,NLAKES
          NODES= ILAKE(1,K)
          NSTRIN= ILAKE(2,K)
          NSTROUT= ILAKE(3,K)
          LRLNODE= ILAKE(4,K)
          LISTRIN= ILAKE(5,K)
          LISTROUT= ILAKE(6,K)
          ISIMMODE= ILAKE(7,K)
          ITERLAKE= ILAKE(8,K)
          IOUTOP= ILAKE(9,K)
          IF(CBCSAVE) IOUTOP=IOUTOP+16
```

```
C         PRINT HEADER IF CELL-BY-CELLS WILL BE PRINTED
          IF(MOD(IOUTOP,2).EQ.1) WRITE(IOUT,1) LAKENAME(K)
    1     FORMAT(/,' CELL-BY-CELL LAKE SEEPAGE (+ TO GW) FOR ',A10)

C         SOLVE FOR STEADY-STATE STAGE
          IF(ISIMMODE.EQ.2) CALL SSSTAGE(RLAKE(1,K),RLNODE(1,LRLNODE),
     &          NODES,ISTRIN(LISTRIN),NSTRIN,ISTROUT(1,LISTROUT),
     &          NSTROUT,RATEQ,MXRATEQ,ITERLAKE,
     &          IBOUND,HEAD,NLAY,NROW,NCOL,ARTRIB,NSS,
     &          ISTRM,STRM,NSTREM)

          STAGE=RLAKE(1,K)

C         CALL BUDGET ROUTINE
          CALL LAKEBUD(RLAKE(1,K),RLNODE(1,LRLNODE),NODES,
     &                 ISTRIN(LISTRIN),NSTRIN,ISTROUT(1,LISTROUT),
     &                 NSTROUT,RATEQ,MXRATEQ,BUDGET,QNET,WETAR,VOL,
     &                 DUM,'BD',IBOUND,HEAD,BUFF,NLAY,NROW,NCOL,
     &                 IOUTOP,IOUT,ARTRIB,NSS)

C         SOLVE FOR TRANSIENT STAGE
          IF(ISIMMODE.EQ.3) THEN
            DT=DELT/NSUBSTEPS
            DO 200 I=1,NSUBSTEPS
              CALL TRSTAGE(RLAKE(1,K),RLNODE(1,LRLNODE),NODES,
     &                   ISTRIN(LISTRIN),NSTRIN,ISTROUT(1,LISTROUT),
     &                   NSTROUT,RATEQ,MXRATEQ,ITERLAKE,
     &                   IBOUND,HEAD,NLAY,NROW,NCOL,ARTRIB,NSS,DT)
  200       CONTINUE
          ENDIF

C         WRITE STAGE/BUDGET INFO
          STAGENW=RLAKE(1,K)
          QPRECIP=BUDGET(1,1)
          QEVAP=BUDGET(2,1)
          QRUNOFF=BUDGET(1,2)+BUDGET(2,2)
          QSTRIN=BUDGET(1,3)
          QSTROUT=BUDGET(2,3)
          QSEEPAGE=BUDGET(1,4)+BUDGET(2,4)

          IF(ILKOUT.GT.0)  THEN
            IF(MOD(IOUTOP,16).GT.7) THEN
              WRITE(ILKOUT,2) LAKENAME(K), TOTIM, STAGE, QPRECIP, QEVAP,
     &          QRUNOFF, QSTRIN, QSTROUT, QSEEPAGE, WETAR, VOL, STAGENW
            ELSE IF(MOD(IOUTOP,8).GT.3) THEN
              WRITE(ILKOUT,3) LAKENAME(K), TOTIM, STAGE
            ENDIF
          ENDIF
    2     FORMAT(A10,1P,11G15.6)
    3     FORMAT(A10,1P,2G15.6)

          IF(MOD(IOUTOP,4).GT.1) THEN
            WRITE(IOUT,4) LAKENAME(K), KSTP, KPER, TOTIM
            WRITE(IOUT,5) STAGE, WETAR, VOL
            WRITE(IOUT,6) ((BUDGET(J,I),J=1,2),I=1,5)
            WRITE(IOUT,7) QNET
            IF(ISIMMODE.EQ.2) THEN
              PCTERROR=QNET/((BUDGET(1,5)-BUDGET(2,5))/2.0)*100.0
              WRITE(IOUT,8) PCTERROR
            ELSE IF(ISIMMODE.EQ.3) THEN
              WRITE(IOUT,9) STAGENW
            ENDIF
          ENDIF
```

```
    4     FORMAT(/,' LAKE VOLUMETRIC BUDGET FOR ',A10,':',
     &          /,' (TIME STEP',I5,', STRESS PERIOD',I5,
     &          ', ELAPSED TIME',1P,G13.6,')')
    5     FORMAT(/,1P,'               STAGE =',8X,G18.9,
     &              /,'         WETTED AREA =',8X,G15.6,
     &              /,'              VOLUME =',8X,G15.6)
    6     FORMAT(/,26X,'INFLOW',8X,'OUTFLOW',1P,
     &              /,'          PRECIP/EVAP:',2G15.6,
     &              /,'     RUNOFF/WITHDRAWL:',2G15.6,
     &              /,'           STREAMFLOW:',2G15.6,
     &              /,'              SEEPAGE:',2G15.6,
     &              /,'                TOTAL:',2G15.6)
    7     FORMAT(/,1P,'            NET FLOW =',8X,SP,G15.6)
    8     FORMAT(/,   '  STEADY-STATE ERROR =',8X,SP,F11.2,'%')
    9     FORMAT(/,1P,'           NEW STAGE =',8X,G15.6)

C         ACCUMULATE AQUIFER FLUX
          TOTIN=TOTIN-BUDGET(2,4)
          TOTOUT=TOTOUT+BUDGET(1,4)

 500    CONTINUE
C       END OF LAKE LOOP

C       SAVE CBC FLUX
        IF(CBCSAVE) CALL UBUDSV(KSTP,KPER,TEXT,ILKCBC,BUFF,NCOL,NROW,
     &          NLAY,IOUT)

C       STORE VOLUMETRIC BUDGET TERMS
        VBVL(1,MSUM)= VBVL(1,MSUM) + TOTIN*DELT
        VBVL(2,MSUM)= VBVL(2,MSUM) + TOTOUT*DELT
        VBVL(3,MSUM)= TOTIN
        VBVL(4,MSUM)= TOTOUT
        VBNM(1,MSUM)= TEXT(1)
        VBNM(2,MSUM)= TEXT(2)
        VBNM(3,MSUM)= TEXT(3)
        VBNM(4,MSUM)= TEXT(4)
        MSUM=MSUM+1

        WRITE(IOUT,*)
        RETURN
      END
C     END OF SUBROUTINE LAK2BD
C
C*********************************************************************
      SUBROUTINE LAKEBUD(RLAKE,RLNODE,NODES,ISTRIN,NSTRIN,ISTROUT,
     &                  NSTROUT,RATEQ,MXRATEQ,BUDGET,QNET,WETAR,VOL,
     &                  DQDS,FCT,IBOUND,HEAD,BUFF,NLAY,NROW,NCOL,
     &                  IFLAG,IOUT,ARTRIB,NSS)

C       ------------------------------------------------------------
C       SUBROUTINE TO CALCULATE ALL FLOW COMPONENTS FOR A LAKE
C       ------------------------------------------------------------

        IMPLICIT NONE
        INTEGER NODES, NSTRIN, NSTROUT, MXRATEQ, NSS, ISTRIN(NSTRIN),
     &          ISTROUT(3,NSTROUT), NLAY, NROW, NCOL, IBOUND, IFLAG,
     &          IOUT
        DOUBLE PRECISION RLAKE(9), STAGE, PRECIP, EVAP, RUNOFF, DRYRCH,
     &          TOTAREA
        REAL RLNODE, RATEQ(4,MXRATEQ), BUDGET(2,5), QNET,
     &       WETAR, VOL, DQDS, BUFF, ARTRIB
        DOUBLE PRECISION HEAD
        CHARACTER*2 FCT
        INTEGER K, LRATEQ, NRATEQ
        REAL DSCHRGE, DDSCHRGE, GETSTRIN
```

```
C       GET LAKE VARIABLES
        STAGE= RLAKE(1)
        PRECIP= RLAKE(4)
        EVAP= RLAKE(5)
        RUNOFF= RLAKE(6)
        DRYRCH= RLAKE(7)
        TOTAREA= RLAKE(9)

C       INITIALIZE BUDGET
        DO 20 K=1,5
          BUDGET(1,K)= 0.0
          BUDGET(2,K)= 0.0
   20   CONTINUE

C       CALL LAKE CELL LOOP, COMPUTES SEEPAGE
        CALL LAKECELL(STAGE,RLNODE,NODES,FCT,NLAY,NROW,NCOL,
     &                HEAD,IBOUND,BUFF,BUFF,BUDGET(1,4),BUDGET(2,4),
     &                DQDS,WETAR,VOL,IFLAG,IOUT,DRYRCH)

C       COMPUTE PRECIP/EVAP & RUNOFF/WITHDRAWL
        IF(PRECIP.GT.0.0) THEN
          BUDGET(1,1) = PRECIP*TOTAREA
        ELSE
          BUDGET(2,1) = PRECIP*TOTAREA
        ENDIF

        IF(EVAP.GT.0.0) THEN
          BUDGET(1,1) = EVAP*WETAR
        ELSE
          BUDGET(2,1) = EVAP*WETAR
        ENDIF

        IF(RUNOFF.GT.0.0) THEN
          BUDGET(1,2) = RUNOFF
        ELSE
          BUDGET(2,2) = RUNOFF
        ENDIF

C       COMPUTE STREAM INFLOW & OUTFLOW, UPDATE DQDS
        IF (NSTRIN.GT.0) THEN
          DO 30 K=1,NSTRIN
   30       BUDGET(1,3)=BUDGET(1,3)+ GETSTRIN(ISTRIN(K),ARTRIB,NSS)
        ENDIF

        IF(NSTROUT.GT.0) THEN
          DO 40 K=1,NSTROUT
            NRATEQ=ISTROUT(2,K)
            LRATEQ=ISTROUT(3,K)
            BUDGET(2,3)=BUDGET(2,3) -
     &                  DSCHRGE(STAGE,RATEQ(1,LRATEQ),NRATEQ)
            DQDS=DQDS - DDSCHRGE(STAGE,RATEQ(1,LRATEQ),NRATEQ)
   40     CONTINUE
        ENDIF

C       TOTAL BUDGET
        BUDGET(1,5)=BUDGET(1,1)+BUDGET(1,2)+BUDGET(1,3)+BUDGET(1,4)
        BUDGET(2,5)=BUDGET(2,1)+BUDGET(2,2)+BUDGET(2,3)+BUDGET(2,4)
        QNET=BUDGET(1,5) + BUDGET(2,5)

        RETURN
      END
C     END OF SUBROUTINE LAKEBUD
C
C*********************************************************************
```

```
      SUBROUTINE LAKECELL(STAGE,RLNODE,NODES,FCT,NLAY,NROW,NCOL,
     &                    HEAD,IBOUND,RHS,HCOF,SEEPIN,SEEPOUT,DQDS,
     &                    WETAR,VOL,IFLAG,IOUT,DRYRCH)

C       ---------------------------------------------------------------
C       SUBROUTINE TO LOOP THROUGH LAKE CELLS & PERFORM A VARIETY OF
C         FUNCTIONS INCL. SUMMING SEEPAGE, WETTED AREA, & FORMULATING
C         THE GW EQUATIONS.
C       ---------------------------------------------------------------

        IMPLICIT NONE
        INTEGER NODES, NLAY, NROW, NCOL, IBOUND(NCOL,NROW,NLAY), IOUT
        DOUBLE PRECISION STAGE, DRYRCH
        REAL RLNODE(5,NODES), RHS(NCOL,NROW,NLAY),
     &       HCOF(NCOL,NROW,NLAY), SEEPIN, SEEPOUT, DQDS, WETAR, VOL,
     &       POS
        DOUBLE PRECISION HEAD(NCOL,NROW,NLAY)
        CHARACTER*2 FCT
        INTEGER K, IL, IR, IC, IBCELL, IFLAG
        REAL TOP, BOT, AREA, COND, H, CSEEP, RCH


C       INITIALIZE
        SEEPIN=0.0
        SEEPOUT=0.0
        DQDS=0.0
        WETAR=0.0
        VOL=0.0
C       BEGIN LOOP
        DO 100 K=1, NODES

C         SET NODE VARIABLES
          CSEEP=0.0
          RCH=0.0
          POS=RLNODE(1,K)
          TOP=RLNODE(2,K)
          BOT=RLNODE(3,K)
          AREA=RLNODE(4,K)
          COND=RLNODE(5,K)

C         CONVERT POS TO IL, IR, IC
          CALL GETPOS(POS,IL,IR,IC,NLAY,NROW,NCOL,IBOUND,IBCELL)

C         ACCUMULATE WETTED AREA AND VOLUME, DQDS
          IF(STAGE.GE.TOP) THEN
            WETAR=WETAR+AREA
            VOL=VOL+AREA*(STAGE-TOP)
            IF(IBCELL.GT.0) DQDS=DQDS-COND
          ENDIF

C         DETERMINE CELL SEEPAGE TO CONNECTED CELLS AND FORMULATE
          IF(IBCELL.GT.0) THEN

C           GET AQUIFER HEAD
            H = HEAD(IC, IR, IL)

C           CHECK FOR UPWARD SEEPAGE TO NON-WETTED CELL, APPLY DRYRCH
            IF(STAGE.LT.TOP) THEN
              IF(H.GT.TOP) THEN
                CSEEP = COND * (TOP-H)
                IF(FCT.EQ.'FM') THEN
                  RHS(IC,IR,IL) = RHS(IC, IR, IL) - COND*TOP
                  HCOF(IC,IR,IL) = HCOF(IC,IR,IL) - COND
                ENDIF
              ELSE
```

```
              RCH=AREA*DRYRCH
              IF(FCT.EQ.'FM') RHS(IC,IR,IL)= RHS(IC,IR,IL)-RCH
            ENDIF

C           CHECK FOR WATER TABLE CONNECTED TO LAKE
            ELSE IF(H.GT.BOT) THEN
              CSEEP = COND * (STAGE-H)
              IF(FCT.EQ.'FM') THEN
                RHS(IC,IR,IL) = RHS(IC, IR, IL) - COND*STAGE
                HCOF(IC,IR,IL) = HCOF(IC,IR,IL) - COND
              ENDIF

C           UNSATURATED BETWEEN LAKE & WATER TABLE
            ELSE
              CSEEP = COND * (STAGE-BOT)
              IF(FCT.EQ.'FM') RHS(IC,IR,IL)=RHS(IC,IR,IL)-CSEEP

            ENDIF
          ENDIF

C         ACCUMULATE TOTAL SEEPAGE
          IF(CSEEP.GT.0.0) THEN
            SEEPOUT= SEEPOUT - CSEEP
          ELSE
            SEEPIN = SEEPIN - CSEEP
          ENDIF

C         PRINT OR STORE CELL-BY-CELL SEEPAGE
          IF(FCT.EQ.'BD') THEN
            IF(RCH.GT.0.0) CSEEP=RCH
            IF(MOD(IFLAG,2).EQ.1)WRITE(IOUT,1) K, IL, IR, IC, CSEEP
            IF(MOD(IFLAG,32).GT.15) RHS(IC,IR,IL)= RHS(IC,IR,IL) + CSEEP
          ENDIF

  100   CONTINUE
    1   FORMAT(4I5,1P,G15.6)

        RETURN
      END
C     END OF SUBROUTINE LAKECELL
C
C***********************************************************************
      REAL FUNCTION DSCHRGE (STAGE, OUTFALL, NRAT)

C       ---------------------------------------------------------------
C       GWC 04/30/96
C       COMPUTES STREAM OUTFLOW FOR A GIVEN LAKE STAGE
C       ---------------------------------------------------------------

        IMPLICIT NONE
        INTEGER NRAT, K
        DOUBLE PRECISION STAGE
        REAL OUTFALL(4,NRAT)
C       LOOP THROUGH RATING EQNS (SORTED BY CUTOFF, DESCENDING)
        DO 100 K=1, NRAT

C         CHECK CUTOFF
          IF (STAGE.GE.OUTFALL(1,K)) THEN

C           APPLY RATING EQN AND RETURN
            DSCHRGE = OUTFALL(2,K)* (STAGE- OUTFALL(3,K))** OUTFALL(4,K)
            RETURN

          ENDIF
```

```
  100   CONTINUE
C       END OF RATING EQUATION LOOP

C       STAGE BELOW ALL CUTOFFS, OUTFLOW = 0.0
        DSCHRGE = 0.0
        RETURN

      END
C     END OF FUNCTION DSCHRGE



      REAL FUNCTION DDSCHRGE (STAGE, OUTFALL, NRAT)

C       ----------------------------------------------------------------
C       GWC 04/30/96
C       COMPUTES DERIVATIVE OF STREAM OUTFLOW W.R.T. LAKE STAGE
C       ----------------------------------------------------------------

        IMPLICIT NONE
        INTEGER NRAT, K
        DOUBLE PRECISION STAGE
        REAL OUTFALL(4,NRAT)

C       LOOP THROUGH RATING EQNS (SORTED BY CUTOFF, DESCENDING)
        DO 100 K=1, NRAT

C         CHECK CUTOFF
          IF (STAGE.GE.OUTFALL(1,K)) THEN

C           APPLY RATING EQN AND RETURN
            DDSCHRGE = OUTFALL(4,K)* OUTFALL(2,K)*
     &                 (STAGE- OUTFALL(3,K))** (OUTFALL(4,K)-1.0)
            RETURN

          ENDIF

  100   CONTINUE
C       END OF RATING EQUATION LOOP

C       STAGE BELOW ALL CUTOFFS, DERIV. OF OUTFLOW = 0.0
        DDSCHRGE = 0.0
        RETURN

      END
C     END OF FUNCTION DDSCHRGE
C
C***********************************************************************
      REAL FUNCTION GETSTRIN(ISEG,ARTRIB,NSS)

C       ----------------------------------------------------------------
C       GWC 04/30/96
C       SET STREAM INFLOW BASED ON VARIABLE QSTROUT
C       ----------------------------------------------------------------

        IMPLICIT NONE
        INTEGER ISEG, NSS
        REAL ARTRIB(NSS)

C       RETURN IF SEGMENT IS OUT OF RANGE
        IF(ISEG.LT.1 .OR. ISEG.GT.NSS) RETURN

C       GET OUTFLOW FROM THIS SEGMENT = ARTRIB(INSEG)
        GETSTRIN= ARTRIB(ISEG)
```

```
        RETURN

      END
C     END OF FUNCTION GETSTRIN
C
C***********************************************************************
      SUBROUTINE SETSTROUT(ISEG,QSTROUT,ISTRM,STRM,NSTREM)

C       -----------------------------------------------------------------
C       GWC 04/30/96
C       SET STREAM INFLOW BASED ON VARIABLE QSTROUT
C       -----------------------------------------------------------------

        IMPLICIT NONE
        INTEGER ISEG, NSTREM, ISTRM(5,NSTREM), K
        REAL QSTROUT, STRM(11,NSTREM)

C       RETURN IF SEGMENT IS OUT OF RANGE
        IF(ISEG.LT.1) RETURN

C       LOOK FOR FIRST OCCURANCE OF ISEG IN ISTRM(4,:)
        DO 100 K=1,NSTREM
  100     IF(ISTRM(4,K).EQ.ISEG) GOTO 200

C       ERROR -- FIRST REACH OF STREAM SEGEMENT NOT FOUND
        PRINT *,' ERROR -- FIRST REACH OF STREAM SEGEMENT NOT FOUND'
       PRINT *, 'ISEG=',ISEG,'  NSTREM=',NSTREM
        STOP

C       SET INFLOW TO THIS REACH = QSTROUT
  200   STRM(1,K) = QSTROUT
        RETURN

      END
C     END OF SUBROUTINE SETSTROUT
C
C***********************************************************************
      SUBROUTINE SETPOS(IPOS,ILAY,IROW,ICOL,NLAY,NROW,NCOL)

C       -----------------------------------------------------------------
C       SUBROUTINE TO STORE 1 BYTE POSITION BASED ON LAYER,ROW,COL
C       CONNECT TO NEXT ACTIVE LAYER FOR LAYER<=0
C       -----------------------------------------------------------------

        INTEGER IPOS, ILAY, IROW, ICOL, NLAY, NROW, NCOL

        IF(ILAY.EQ.0) ILAY=-1
        IPOS = (ABS(ILAY)-1)*NROW*NCOL + (IROW-1)*NCOL + ICOL
        IF(ILAY.LT.0) IPOS= -IPOS

      END
C     END OF SUBROUTINE SETPOS
C
C***********************************************************************
      SUBROUTINE GETPOS(IPOS,ILAY,IROW,ICOL,NLAY,NROW,NCOL,IBOUND,IBCEL)

C       -----------------------------------------------------------------
C       SUBROUTINE TO RETRIEVE LAYER,ROW,COL FROM 1 BYTE POSITION
C       CONNECT TO NEXT ACTIVE LAYER FOR POSITION < 0
C       -----------------------------------------------------------------

        INTEGER IPOS,ILAY,IROW,ICOL,NLAY,NROW,NCOL,NRC,IBCEL,ITMP
        INTEGER IBOUND(NCOL,NROW,NLAY)

        NRC=NROW* NCOL
```

```
        ITMP = ABS(IPOS) - 1
        ILAY = ITMP/NRC + 1
        IROW = MOD(ITMP,NRC)/NCOL + 1
        ICOL = MOD(ITMP,NCOL) + 1

        IF(IPOS.GT.0) THEN
          IBCEL = IBOUND(ICOL,IROW,ILAY)
          RETURN
        ENDIF

        ITMP=ILAY
        DO 100 ILAY=ITMP,NLAY
          IBCEL = IBOUND(ICOL,IROW,ILAY)
          IF(IBCEL.NE.0) RETURN
  100   CONTINUE
        ILAY=ITMP
        RETURN

      END
C     END SUBROUTINE GETPOS
C
C***********************************************************************
      SUBROUTINE SSSTAGE(RLAKE,RLNODE,NODES,ISTRIN,NSTRIN,ISTROUT,
     &                   NSTROUT,RATEQ,MXRATEQ,ITERLAKE,
     &                   IBOUND,HEAD,NLAY,NROW,NCOL,ARTRIB,NSS,
     &                   ISTRM,STRM,NSTREM)

C       ----------------------------------------------------------------
C       SUBROUTINE TO CALCULATE STEADY-STATE STAGE
C       ----------------------------------------------------------------

        IMPLICIT NONE
        INTEGER NODES, NSTRIN, NSTROUT, MXRATEQ, NSS, ISTRIN,
     &          ISTROUT(3,NSTROUT), NLAY, NROW, NCOL, IBOUND, ISTRM,
     &          NSTREM, ITERLAKE
        DOUBLE PRECISION RLAKE(9), STAGEMX, STAGEMN, CONVCRIT, STAGEOLD,
     &          STAGE
        REAL RLNODE, RATEQ(4,MXRATEQ), BUDGET(2,5), QNET,
     &       WETAR, VOL, DQDS, ARTRIB, STRM
        DOUBLE PRECISION HEAD
        INTEGER K, ISEGOUT, NRATEQ, LRATEQ
        LOGICAL NEWTON
        REAL DSCHRGE

C       INITIALIZE
        STAGE=RLAKE(1)
        STAGEMX=RLAKE(2)
        STAGEMN=RLAKE(3)
        CONVCRIT=RLAKE(8)/10.0
        NEWTON=.TRUE.

C       BEGIN IETERATION LOOP
        DO 100 K=1,ITERLAKE
          STAGEOLD=STAGE

C         CALL LAKEBUD TO GET NET FLOW & DQDS
          CALL  LAKEBUD(RLAKE,RLNODE,NODES,ISTRIN,NSTRIN,ISTROUT,
     &                  NSTROUT,RATEQ,MXRATEQ,BUDGET,QNET,WETAR,VOL,
     &                  DQDS,'SS',IBOUND,HEAD,IBOUND,NLAY,NROW,NCOL,
     &                  0,0,ARTRIB,NSS)

C         RESET BOUNDS
          IF(QNET.GT.0.0) THEN
            STAGEMN= STAGE
          ELSE
```

```
            STAGEMX= STAGE
          ENDIF

C         NEWTON'S METHOD UPDATE/CHECK
          IF(NEWTON) THEN

C           CHECK FOR DISCONNECTED LAKE (NO SEEPAGE OR STREAM OUTFLOW)
            IF(DQDS.GE.0.0) THEN
              NEWTON=.FALSE.
              STAGE=(STAGEMN+STAGEMX) /2.0
              RLAKE(1) = STAGE
              GO TO 100
            ENDIF

C           UPDATE STAGE
            STAGE= STAGE- QNET/DQDS

C           CHECK FOR NON-CONVERGENCE WITH NEWTON'S METHOD
            IF(STAGE.GE.STAGEMX .OR. STAGE.LE.STAGEMN) THEN
              NEWTON= .FALSE.
              STAGE=(STAGEMN+STAGEMX) /2.0
              RLAKE(1) = STAGE
              GO TO 100
            ENDIF

C         RANGE SPLIT UPDATE
          ELSE
            STAGE= (STAGEMN + STAGEMX) /2.0
          ENDIF

C         SET STAGE AND CHECK FOR CONVERGENCE
          RLAKE(1)= STAGE
          IF(ABS(STAGE-STAGEOLD).LE.CONVCRIT) GO TO 200

  100   CONTINUE

C       FAILED TO CONVERGE, KEEP GOING ANYWAY

C       SET FLOW TO LAKE OUTFLOW SEGMENTS FOR NEXT FORMULATION
  200   IF(NSTROUT.GT.0) THEN
          DO 40 K=1,NSTROUT
            ISEGOUT=ISTROUT(1,K)
            NRATEQ=ISTROUT(2,K)
            LRATEQ=ISTROUT(3,K)
            QNET= DSCHRGE(STAGE,RATEQ(1,LRATEQ),NRATEQ)
            CALL SETSTROUT(ISEGOUT,QNET,ISTRM,STRM,NSTREM)
   40     CONTINUE
        ENDIF

        RETURN
      END
C     END OF SUBROUTINE SSSTAGE
C
C***********************************************************************
      SUBROUTINE TRSTAGE(RLAKE,RLNODE,NODES,ISTRIN,NSTRIN,ISTROUT,
     &                   NSTROUT,RATEQ,MXRATEQ,ITERLAKE,
     &                   IBOUND,HEAD,NLAY,NROW,NCOL,ARTRIB,NSS,DT)

C       ----------------------------------------------------------------
C       SUBROUTINE TO CALCULATE NEW STAGE IN TRANSIENT SIMULATION
C       ----------------------------------------------------------------

        IMPLICIT NONE
        INTEGER NODES, NSTRIN, NSTROUT, MXRATEQ, NSS, ISTRIN,
     &          ISTROUT, NLAY, NROW, NCOL, IBOUND, ITERLAKE
```

```
      DOUBLE PRECISION RLAKE(8), STAGEMX, STAGEMN, CONVCRIT
      REAL RLNODE, RATEQ, BUDGET(2,5), QNET,
     &     WETAR, VOL, DQDS, ARTRIB, DT
      DOUBLE PRECISION HEAD
      INTEGER K
      REAL DVOL, DSTAGE, TARGVOL

C     GET LAKE VARIABLES
      STAGEMX= RLAKE(2)
      STAGEMN= RLAKE(3)
      CONVCRIT= RLAKE(8)

C     GET BUDGET, SET TARGET VOLUME
      CALL LAKEBUD(RLAKE,RLNODE,NODES,ISTRIN,NSTRIN,ISTROUT,
     &             NSTROUT,RATEQ,MXRATEQ,BUDGET,QNET,WETAR,VOL,
     &             DQDS,'TR',IBOUND,HEAD,IBOUND,NLAY,NROW,NCOL,
     &             0,0,ARTRIB,NSS)

      DVOL = QNET*DT
      TARGVOL = VOL+DVOL

C     CHECK FOR EMPTYING LAKE, SET STAGE TO MINIMUM
      IF(TARGVOL.LE.0.0) THEN
        RLAKE(1) = STAGEMN
        RETURN
      ENDIF

C     CHECK FOR ALREADY DRY LAKE, START STAGE IN MIDDLE
      IF (WETAR.LE.0.0) THEN
        RLAKE(1)= 0.5* (STAGEMX + STAGEMN)
        CALL LAKEBUD(RLAKE,RLNODE,NODES,ISTRIN,NSTRIN,ISTROUT,
     &               NSTROUT,RATEQ,MXRATEQ,BUDGET,QNET,WETAR,VOL,
     &               DQDS,'TR',IBOUND,HEAD,IBOUND,NLAY,NROW,NCOL,
     &               0,0,ARTRIB,NSS)
        DVOL = TARGVOL - VOL
      ENDIF

C     ITERATE TO TARGVOL
      DO 100 K=1, ITERLAKE
        DSTAGE = DVOL/WETAR
        RLAKE(1) = RLAKE(1) + DSTAGE
        IF(DSTAGE.LE.CONVCRIT) GOTO 110
        CALL LAKEBUD(RLAKE,RLNODE,NODES,ISTRIN,NSTRIN,ISTROUT,
     &               NSTROUT,RATEQ,MXRATEQ,BUDGET,QNET,WETAR,VOL,
     &               DQDS,'TR',IBOUND,HEAD,IBOUND,NLAY,NROW,NCOL,
     &               0,0,ARTRIB,NSS)

        DVOL = TARGVOL-VOL
 100  CONTINUE

 110  IF(RLAKE(1).GT.STAGEMX) RLAKE(1)=STAGEMX

      RETURN
     END
C    END OF SUBROUTINE TRSTAGE
```

Appendix B. LAK2 Input File Instructions

```
                         IDENTIFICATION LINE

/*LAK2.2
  format:  exactly as shown


                              SIMULATION DATA

NLAKES  ILKCBC  ILKOUT  IECHO  NSUBSTEPS
  format:  5I10

NAME  ISIMMODE  STSTAGE  ITERLAKE  CONVCRIT   (one line for each lake)
  format:  A10, I10, F10.0, I10, F10.0


                               PHYSICAL DATA
                          (one set for each lake)

NODES  NSTRIN  NSTROUT  STAGEMX  ICONDOP
  format:  3I10, F10.0, I10

ISEGIN                  (one line for each inflow stream)
  format:  I10

ISEGOUT  NRATEQ                                                  | one set
  format:  2I10                                                  | for each
                                                                 | outflow
CUTOFF  CONST  ELEV  EXPNT  (one line for each rating equation)  | stream
  format:  4F10.0           (sorted by CUTOFF, descending)       |

ILAY  IROW  ICOL  TOP  BOT  AREA  COND  (one line for each lake node)
  format:  3I10, 4F10.0


                             STRESS PERIOD DATA
                       (one set for each stress period)

ITMP
  format:  I10

PRECIP  EVAP  RUNOFF  DRYRCH  IOUTOP  STAGE  (one line each lake if ITMP >=0)
  format:  4F10.0, I10, F10.0

-------------------------------------------------------------------------------

                            VARIABLE DESCRIPTIONS

NLAKES:     Number of lakes
ILKCBC:     >0 Cell-by-cell unit number, <=0 Do not save cell-by-cell
ILKOUT:     >0 Stage/Budget unit number, <=0 Do not write stage/budget records
IECHO:      <0 No input echoing, 0 Summary of input, >0 Full echoing of input
NSUBSTEPS: Number of sub-time-steps for simulating lakes in transient mode

NAME:       Name (ID) of lake (10 characters)
ISIMMODE:   Simulation Mode:
            0 Fixed Stage, 1 Interpolated stage, 2 Steady-state, 3 Transient
STSTAGE:    Starting Stage (not required for ISIMMODE = 0)
ITERLAKE:   Max iterations for stage solver (ISIMMODE = 2 or 3)
CONVCRIT:   Stage solver termination criteria (change in stage in 1 iteration,
            ISIMMODE = 2 or 3)
```

NODES: Number of lake nodes
NSTRIN: Number of inflow streams
NSTROUT: Number of outflow streams
STAGEMX: Maximum lake stage
ICONDOP: <=0 Hydraulic conductivity input, >0 Conductance input

ISEGIN: Inflow stream segment (from Streamflow Routing Package)
ISEGOUT: Outflow stream segment (from Streamflow Routing Package)
NRATEQ: Number of equations used to define stage-discharge relationship

CUTOFF: Lower stage limit of rating equation
CONST: Rating equation constant
ELEV: Rating equation reference (outfall) elevation
EXPNT: Rating equation exponent

Outflow = CONST * (STAGE - ELEV)^ EXPNT (Above CUTOFF)

ILAY: Lake node model layer (0 for top active layer)
IROW: Lake node model row
ICOL: Lake node model column
TOP: Lakebed top elevation
BOT: Lakebed bottom elevation
AREA: Lake node area
COND: Lakebed hydraulic conductivity or conductance (see ICONDOP above)

ITMP: <0 Use information from last stress period, >=0 read new information
PRECIP: Total-area-dependent flow rate (L/T). The Lake Package budget routine multiplies PRECIP by the total area of the lake (wetted cells + shore cells) and adds the resulting flux to the lake's volumetric budget. Specify a positive number for lake inflow or a negative number for
lake outflow.
EVAP: Wetted-area-dependent flow rate (L/T). The Lake Package budget routine multiplies EVAP by the wetted area of the lake (excluding shore cells)
and adds the resulting flux to the lake's volumetric budget. Specify a positive number for lake inflow or a negative number for lake
outflow.
RUNOFF: Fixed lake inflow (L^3/T, positive = inflow to lake, e.g. runoff)
DRYRCH: Recharge rate applied to groundwater beneath dry lake cells (L/T)
IOUTOP: Output option, constructed as follows:
0 = no output
+1 = print cell-by-cell flows in main output file
+2 = print lake budget information in main output file
+4 = write stage to stage/budget output file
+8 = write flows (& stage) to stage/budget output file
(e.g. 6 to print lake budget in main output file, write stage record to stage/budget output file)
STAGE: If ISIMMODE = 0, lake stage for the stress period
If ISIMMODE = 1, final lake stage for the stress period
Ignored for ISIMMODE = 2 or 3

Appendix C. Instructions for Incorporating LAK2 into MODFLOW

I. Modify The Main Modflow Routine

The main MODFLOW routine should be changed to include the Lake Package. The exact method of implemetation will depend on the version of MODFLOW it is applied to. For example, different MODFLOW versions may differ in the way the IUNIT array is handled, the method for opening files, and the method of allocating or dimensioning arrays. Changes to the FORTRAN source code should be made by a programmer who understands the FORTRAN syntax and the flow of operations within the main MODFLOW routine. Adaptations of the instructions below are often warranted.

The Lake Package reads and sets variables used by Version 2 of the (Prudic) Streamflow Routing Package (the version is identified in the comments below the SUBROUTINE statements of the Streamflow Routing Package -- even though the subroutines are called STR1AL, STR1RP, etc., they may in fact be Version 2 subroutines).

It will be assumed that the Lake Package will be activated by activating IUNIT(15). If that slot is already taken by another package, simply replace all references to IUNIT(15) with another IUNIT location (e.g. IUNIT(20)) in the statements below.

An example of a main MODFLOW routine that has already been modified to include the Lake Package is provided electronically as part of Appendix D.

A. Declarations Section

In the declarations section (where the PARAMETER, DIMENSION, and CHARACTER statements are) of the main MODFLOW routine, the character array LAKENAME should be declared.

If your FORTRAN compiler doesn't support run-time array allocation (of if you don't know the syntax used to invoke run-time allocation), add the following lines in the declarations section:

```
C-----LAK2.2 (BEGIN)
      PARAMETER (MXLAKES=100)
      CHARACTER*10 LAKENAME
      DIMENSION LAKENAME(MXLAKES)
C-----LAK2.2 (END)
```

Note: The value chosen for MXLAKES is the maximum number of lakes allowed in a simulation.

If Fortran90-style auto-allocation is supported by your compiler, use these lines instead:

```
C-----LAK2.2 (BEGIN)
      CHARACTER*10 LAKENAME(:)
      ALLOCATABLE LAKENAME
C-----LAK2.2 (END)
```

B. CUNIT Array

In MODFLOW-96 a CUNIT character array is used to associate the IUNIT array slots with character abbreviations for the various packages. Older versions of MODFLOW do not include a CUNIT array.

If the CUNIT array is initialized, modify the statement to include the 'LAK ' identifier in the 15th position (to associate the Lake Package with IUNIT(15)). In the following example, the original CUNIT initialization statement is commented-out and replaced by one that includes the 'LAK ' identifier:

```
C-----LAK2.2 (BEGIN)
C     DATA CUNIT/'BCF ','WEL ','DRN ','RIV ','EVT ','TLK ','GHB ',
C    1           'RCH ','SIP ','DE4 ','SOR ','OC  ','PCG ','GFD ',
C    2           '    ','HFB ','RES ','STR ','IBS ','CHD ','FHB ',
C    3           '    ','    ','    ','    ','    ','    ','    ',
C    4           '    ','    ','    ','    ','    ','    ','    ',
C    5           '    ','    ','    ','    ','    '/
      DATA CUNIT/'BCF ','WEL ','DRN ','RIV ','EVT ','TLK ','GHB ',
     1           'RCH ','SIP ','DE4 ','SOR ','OC  ','PCG ','GFD ',
     2           'LAK ','HFB ','RES ','STR ','IBS ','CHD ','FHB ',
     3           '    ','    ','    ','    ','    ','    ','    ',
     4           '    ','    ','    ','    ','    ','    ','    ',
     5           '    ','    ','    ','    ','    '/
C-----LAK2.2 (END)
```

If there is no CUNIT declaration or initialization statement, then no change is required. See section 3 on opening files, below.

C. Initialize Streamflow Routing Variables

Before any CALL statements in the MODFLOW routine, the following lines should be added to initialize Streamflow Routing Package variables that are also used by the Lake Package:

```
C-----LAK2.2 (BEGIN)
      NSS=0
      NSTREM=0
      LCTRIB=1
      LCSTRM=1
      ICSTRM=1
C-----LAK2.2 (END)
```

D. LAK2AL Call

In the group of calls to allocation (AL) routines, just after the call to STR1AL, the Lake Package allocation routine should be called:

```
C-----LAK2.2 (BEGIN)
      IF(IUNIT(15).GT.0) CALL LAK2AL(ISUM,LENX,LCILAKE,LCRLAKE,LCRLNODE,
     &                  LCISTRIN,LCISTROUT,LCRATEQ,NLAKES,ILKCBC,ILKOUT,
     &                  IECHO,NSUBSTEPS,MXNODE,MXSTRIN,MXSTROUT,MXRATEQ,
     &                  IUNIT(15),IOUT)
C-----LAK2.2 (END)
```

If run-time allocation of the LAKENAME variable is not being used, a check should be made that the number of lakes is not too large with these lines just after the above CALL statement:

```
C-----LAK2.2 (BEGIN)
      IF(IUNIT(15).GT.0 .AND. NLAKES.GT.MXLAKES) THEN
        WRITE(IOUT,492)
  492   FORMAT(' MAXIMUM NUMBER OF LAKES EXCEEDED, INCREASE MXLAKES.')
        STOP
      ENDIF
C-----LAK2.2 (END)
```

If run-time allocation of the LAKENAME varable is being used, the following ALLOCATE statement should be added just after that CALL statement:

```
C-----LAK2.2 (BEGIN)
      IF(IUNIT(15).GT.0) ALLOCATE(LAKENAME(NLAKES))
C-----LAK2.2 (END)
```

E. LAK2RP1 Call

In the group of CALL statements to read-and-prepare (RP) routines BEFORE the stress period DO loop begins (i.e. before the line that reads DO 300 KPER=1,NPER), the following statement should be added:

```
C-----LAK2.2 (BEGIN)
      IF(IUNIT(15).GT.0) CALL LAK2RP1(X(LCILAKE),X(LCRLAKE),
     &          X(LCRLNODE),X(LCISTRIN),X(LCISTROUT),X(LCRATEQ),
     &          LAKENAME,NLAKES,MXNODE,MXSTRIN,MXSTROUT,
     &          MXRATEQ,IECHO,IUNIT(15),IOUT,NLAY,NROW,NCOL)
C-----LAK2.2 (END)
```

F. LAK2RP2 Call

In the group of RP calls INSIDE the stress period loop, the following lines should be added:

```
C-----LAK2.2 (BEGIN)
      IF(IUNIT(15).GT.0) THEN
C-------RECOMPUTE LENGTH OF PERIOD, PERLEN, A LOCAL VARIABLE IN
```

```
C-------SUBROUTINE BAS1AD (CODE TAKEN FROM RES1.FOR)
        PERLEN=DELT*FLOAT(NSTP)
        IF(TSMULT.NE.1.) PERLEN=DELT*(1.-TSMULT**NSTP)/(1.-TSMULT)
        CALL LAK2RP2(X(LCILAKE),X(LCRLAKE),LAKENAME,NLAKES,IUNIT(15),
     &    IOUT,PERLEN)
      ENDIF
C-----LAK2.2 (END)
```

G. LAK2AD Call

The call to the Lake Package advance (AD) routine should be made somewhere BETWEEN the call to the Basic Package AD routine (BAS1AD, BAS5AD, or similar), and the call to the BCF AD routine (e.g. BCF1AD, BCF2AD, BCF5AD).

```
C----- LAKE2.2 (BEGIN)
      IF(IUNIT(15).GT.0) CALL LAK2AD(X(LCILAKE),X(LCRLAKE),X(LCISTROUT),
     &          X(LCRATEQ),NLAKES,MXSTROUT,MXRATEQ,X(ICSTRM),
     &          X(LCSTRM),NSTREM,DELT)
C----- LAKE2.2 (END)
```

H. LAK2FM Call

The Lake Package formulate (FM) call should be inserted AFTER the call to the BCF formulate call (which may be labeled BCF1FM, BCF2FM, etc.), and BEFORE the call to STR1FM:

```
C-----LAK2.2 (BEGIN)
      IF(IUNIT(15).GT.0) CALL LAK2FM(X(LCILAKE),X(LCRLAKE),X(LCRLNODE),
     &          X(LCISTRIN),X(LCISTROUT),X(LCRATEQ),NLAKES,
     &          MXNODE,MXSTRIN,MXSTROUT,MXRATEQ,X(LCHNEW),X(LCIBOU),
     &          X(LCHCOF),X(LCRHS),NLAY,NROW,NCOL,X(LCTRIB),NSS,
     &          X(LCSTRM),X(ICSTRM),NSTREM)
C-----LAK2.2 (END)
```

I. LAK2BD Call

The call to the Lake budget (BD) routine should be made AFTER the call to STR1BD:

```
C----- LAKE2.2 (BEGIN)
      IF(IUNIT(15).GT.0) CALL LAK2BD(X(LCILAKE),X(LCRLAKE),X(LCRLNODE),
     &          X(LCISTRIN),X(LCISTROUT),X(LCRATEQ),NLAKES,
     &          MXNODE,MXSTRIN,MXSTROUT,MXRATEQ,X(LCHNEW),X(LCIBOU),
     &          X(LCBUFF),NLAY,NROW,NCOL,X(LCTRIB),NSS,
     &          IOUT,ILKCBC,ILKOUT,NSUBSTEPS,DELT,VBVL,VBNM,
     &          MSUM,KSTP,KPER,ICBCFL,TOTIM,LAKENAME,X(ICSTRM),
     &          X(LCSTRM),NSTREM)
C----- LAKE2.2 (END)
```

This completes the code modifications needed to incorporate the Lake Package into an existing version of MODFLOW.

II. Compiling MODFLOW with the Lake Package

The file Lak2.for should be added to the list of sources to be compiled and then linked to create the MODFLOW executable. This can usually be accomplished with a minor modification to an existing Makefile. The compiler's user's manual may provide more specific details on compiling source files to create an execuatable program.

When compiling Lak2.for you may receive warnings that the variable types or sizes in different references to subroutines do not conform. These warnings should be ignored. Note that similar warnings may also result for other MODFLOW packages.

III. Opening Files

In MODFLOW-96 and other versions of MODFLOW, a "name" file is used to list all input and output files for a simulation. In this case, opening the Lake Package input file is easily accomplished by including an extra line in the "name" file with the correct unit number (from IUNIT(15)) and (for MODFLOW-96) the CUNIT identifier 'LAK '. In MODFLOW-96, the numbers specified in the IUNIT array in the Basic Package are ignored -- the CUNIT identifiers are used instead.

Additionally, the Lake Package can create a separate ASCII data output file to report simulated stages and budgetary information. This file is used depending on the value set for the ILKOUT variable (see Lak2file.txt). This file can also generally be opened using the name file (specify the same unit number as ILKOUT, and, in MODFLOW-96, use the "DATA" identifier).

If your version of MODFLOW opens files using prompts to the screen, then the Lake Package input and output files should be similarly opened, when needed, by adding the appropriate code changes. The syntax of these code changes should be similar to the file opening statements used for other packages.

Some UNIX versions of MODFLOW will automatically read and write data from/to files named fort.x where x is the unit number. In this case, use the unit number specified in IUNIT(15) for the Lake Package input file, and the unit number specified in ILKOUT will be used for the Lake Package output file.

Appendix D. Electronic Files

MODFLOW-96 Code with the Lake Package

A file called "ModfLake.zip" is supplied along with this manual. The file contains the MODFLOW-96 source code, modified to incorporate the Lake Package, a Makefile for use with the Lahey FORTRAN-90 compiler, a PC/Windows executable MODFLOW program with the Lake Package, and an example problem (from Cheng and Andersen, 1993). With the enclosed MODFLOW program, the Lake Package is activated by specifying a file of type "LAK" in the Name File. The supplied file is a PKZIP archive that includes directory names. Please see the README file in the archive for more information.

Input and Output Files for the Test problems

A file called "TestProb.zip" is supplied along with this manual. The file contains input and output files for all simulations discussed in Section 5 of this manual. Binary output files are of Lahey FORTRAN-90 unformatted type. The file is a PKZIP archive that includes directory names. Please see the README file in the archive for more information.