CHAPTER 4
BASIC PACKAGE

The Basic (BAS) Package of the GWF Process handles a number of administrative tasks for MODFLOW. It reads the Name File to open files and determine options that will be active. The BAS Package declares and allocates memory for variables, reads the IBOUND variable, reads initial head and tracks head throughout time, reads data defining the discretization of space and time, calculates an overall water budget, and controls model output according to user specifications. The BAS Package also reads zone and multiplier arrays that can be used by other packages to define parameters and reads the optional Parameter Value File. Much of the data declared by the GWF BAS Package is used by other GWF Process packages and other processes.

Opening Files and Activating Options Using the Name File

The BAS Package reads the Name File. Each line in the Name File specifies a file name, type, and unit number. The file unit number is used by Fortran to refer to the file. The file type specifies the purpose of a file. Many of the file types correspond to major options; a major program option is a major segment of the program that is utilized only at the user's request. Specifying a file having the corresponding file type causes the option to be activated. Likewise, an option is inactive if the Name File does not include a file having the option's file type. Most packages are themselves major options, so each package generally has at least one input file that is specified in the Name File. For example, if areal recharge is simulated, a file having a file type of RCH will be required. The Input Instructions section defines all of the file types that can be used. The Name File must include two files required by the Basic Package, the BAS Package file and the Discretization File. In addition, the Name File can include files for several options included in the Basic Package: Time-Variant Constant Head, Output Control, Zone arrays, Multiplier arrays. These options are described throughout this chapter.

File types DATA and DATA(BINARY) define input or output files whose file unit numbers are referenced by other files. The DATA and DATA(BINARY) file types can be specified for many files. An example is the input of variables containing multiple values that are accessed by grid indices. The data can be read from the appropriate package file, or the package file can indicate that the data should be read from a different file unit number. If a different file unit number is indicated, the file connected to that file unit number must be specified in the Name File. For example, the RCH Package reads the recharge rate for every cell in a model layer when recharge is being simulated. These recharge values can be defined directly in the RCH file (that is, the file whose file type is RCH), or the RCH file can indicate that the recharge values should be read from a different file. Another example is the output of computed head, which can be written to a file unit number. The file connected to that file unit number must be specified in the Name File.

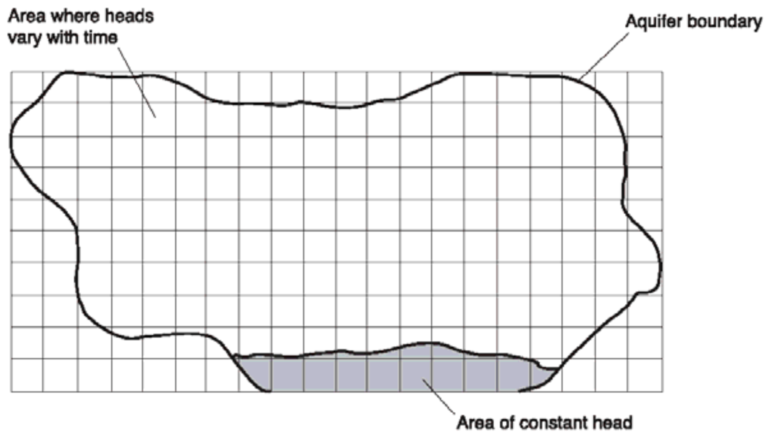
The user enters the name of the Name File either through a command line argument or an interactive prompt. Once the name of the Name File is specified, the program runs to completion without any further user intervention. If all steps in the simulation work successfully, the required input files will be opened, the input data will be read from the input files, and results will be written to output files.

Global Data

The BAS Package declares variables, called global data, that can be shared throughout the GWF Process and other processes and allocates the memory required for the data. A user need not know the names of most of the program variables, but names of variables are essential information for those wanting to understand or modify the program. Chapter 9, Programmer Documentation, defines all of the global data.

The IBOUND Variable

Recall that in Chapter 2 the finite-difference equation (eq. 2–26) is written for each variable-head cell in the grid. The IBOUND variable contains a code for each cell that indicates whether (1) the head varies with time (variable-head cell), (2) the head is constant (constant-head cell), or (3) no flow takes place within the cell (no-flow or inactive cell). Values of IBOUND are read from the BAS Package file. IBOUND can be modified by other packages if the state of a cell changes. Distribution of the IBOUND code entries for a typical model layer is shown in figure 4–1.



0	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0

IBOUND Codes

<0 Constant head

$=0$ No flow

>0 Variable head

Figure 4–1. Example of the IBOUND variable for a single layer. (Modified from McDonald and Harbaugh, 1988.)

Constant-head cells also can be designated using the Time-Variant Specified-Head (CHD) Option of the BAS Package. This capability was originally developed by Leake and Prudic (1991) and called a package; however, the capability is viewed here as an option within the BAS Package because of its function to set values of IBOUND. When the CHD Option is used by including a file with file type CHD in the Name File, constant-head cells are read from the CHD file as a list of cells. For each cell in the list, the CHD Option sets IBOUND equal to negative one. The CHD Option is invoked after IBOUND is read from the BAS Package file, so CHD can be used to designate constant-head cells without specifying a negative value for IBOUND in the BAS Package file.

In addition to designating cells as constant head, the CHD Option specifies the head at the designated constant-head cells. As explained in the following section, the head at cells that are designated as constant head when IBOUND is read is set equal to the initial head. For cells designated as constant head using the CHD Option, the head is specified in the CHD file. If a cell that is designated as constant head in the CHD Option was already designated as constant head when IBOUND was read, the value for head specified by the CHD Option is used rather than the initial head. CHD reads data every stress period, which makes changing the head at constant-head cells throughout a simulation possible.

Initial Head and Tracking Head Throughout Time

Because equation 2–26 is in backward-difference form, a known head distribution is required at the beginning of each time step of a transient simulation in order to calculate the head distribution at the end of that time step. For each time step after the first, the head distribution at the beginning of one time step is set equal to the head distribution at the end of the previous time step. Heads at the beginning of the first time step are called initial heads, and initial heads are read from the BAS Package input file.

Ideally, the initial heads are defined from a previous steady-state simulation. This insures that the initial heads are accurate at all locations. For example, a model might be developed to simulate the effect of a well field on a previously steady-state system. If the steady-state system without the well is simulated first, then the simulated steady-state heads can be used as initial heads for the transient simulation. Of course, errors in the steady-state heads can occur as a result of errors or misrepresentations in model data, but the steady-state heads will be exact for the data being used. Without simulated steady-state heads, initial heads have to be estimated for all model cells based upon measured heads at observation points. Undoubtedly some error in these interpolated heads may exist, which would impact the calculated heads in the transient model. Note, however, that the errors in simulated head caused by errors in initial heads will decrease with time depending on the storage and transmissive factors of the porous medium being simulated. Thus, the transient model possibly can be started far enough in advance of the time for which results are required so that any errors in head resulting from errors in initial heads will have dissipated.

Even for steady-state simulations, the initial heads must be specified in the BAS Package input file because initial heads become the initial estimate of heads for the interative solvers. In most situations, the initial heads do not affect the computed steady-state heads but may affect how many iterations are required. That is, the closer the initial head is to the final head, the fewer iterations will be needed; however, the impact on the number of iterations is not large unless the initial head is far from the final head. Thus, making a detailed estimate of the initial head for steady-state simulations is generally not warranted.

A water-table cell that is not given the option to convert from no flow to variable head is an example of a situation in which initial head can affect the computed steady-state head. (The option to allow cells to convert from no flow to variable head is discussed in Chapter 5.) If the initial head is below the bottom of the aquifer at such a cell, then the cell will convert to no flow at the start of the simulation and must stay no flow throughout the simulation; yet if the initial head is above the bottom of the aquifer, that cell will begin as a variable-head cell and may remain saturated depending on conditions in the simulation. See Chapter 5 for a further discussion of the effect of initial head in water-table cells.

At cells that are specified as constant head when IBOUND is read from the BAS Package file (IBOUND<0), the initial head becomes the constant head. The head at these cells remains the same throughout the simulation unless the head is changed using the Time-Variant Specified-Head (CHD) Option.

Discretization of Space

As described in Chapter 2, space is discretized by a grid of cells. The finite-difference grid in MODFLOW is assumed to be rectangular horizontally, while the grid can be distorted vertically (fig. 4–2). The Basic Package reads spatial discretization information from a file called the Discretization File, which is required for all model simulations.

The number of rows and columns are specified in variables NROW and NCOL, respectively. The horizontal size of cells is defined by variables DELR and DELC, which represent Δr and Δc, respectively, in the equations of Chapter 2. DELR has one value for every column in the grid, and the value of $DELR_J$ is the width of column J (fig. 4–2A). This width of a specified column is the same for all rows. Similarly $DELC_I$ is the width of row I. The width of a row is the same for all columns.

Some users of previous versions of MODFLOW have reported confusion over DELR and DELC. They interpret DELR to represent distance “between” rows rather than distance “along” rows. DELR is the cell size in the X direction if the grid for a layer is imposed on a Cartesian coordinate system. Likewise, DELC is the cell size in the Y direction.

The number of layers in the grid is specified in variable NLAY. Layers are numbered starting from the top layer and going down (fig. 4–2B). Cell elevations are specified in variable BOTM. Elevation of the top of layer 1 is specified in addition to the bottom elevation of every layer. Below each layer except the bottom layer, a confining bed also can be included through which only vertical flow is simulated. Simulating confining beds by this method often is called the Quasi-Three-Dimensional (Quasi–3D) approach (McDonald and Harbaugh, 1988, p. 5–18). For these confining beds, the elevation of the bottom of the bed is specified also. Use of the Quasi–3D approach is not required; that is, any confining bed can be simulated using one or more distinct model layers as desired. Chapter 5 explains how the Quasi–3D approach is implemented.

The elevation information is used to calculate the thickness of all cells and confining beds below cells. As described, the user must specify the top elevation of layer 1 and the bottom elevations of all model layers and confining beds. The top elevation of other model layers and confining beds need not be separately specified because the top elevation is the same as the bottom elevation of the layer or confining bed that is directly above. Use of distorted layers, as shown in figure 4–2B, is not required. A flat layer is indicated by simply specifying the same value for the bottom elevation of all the cells in the layer.

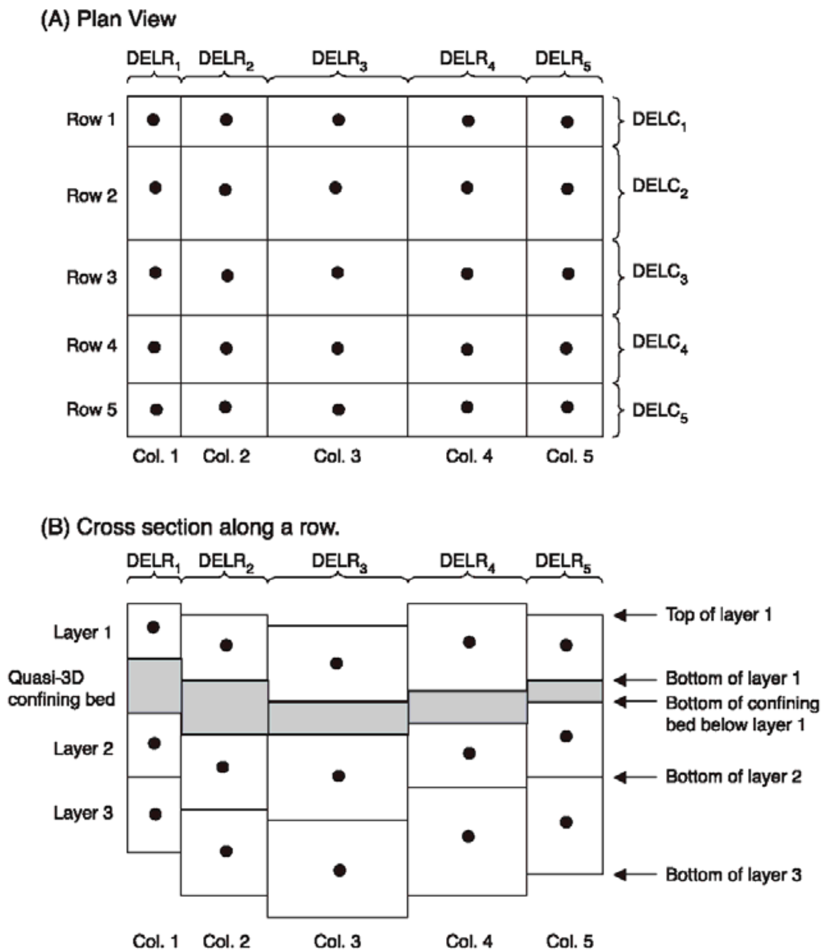


Figure 4–2. Finite-difference grid in (A) plan view and (B) cross-section view.

Discretization of Time

Simulation time is divided into stress periods—time intervals during which the input data for all external stresses are constant—which are, in turn, divided into time steps (fig. 4–3). Note that time steps are fundamental to the finite-difference method whereas stress periods have been incorporated in MODFLOW as a convenience for user input. Discretization information for time is read from the Discretization File.

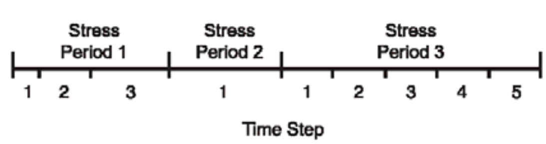


Figure 4–3. Division of simulation time into stress periods and time steps.

Within each stress period, the time steps form a geometric progression. The user specifies the length of the stress period (PERLEN), the number of time steps into which the stress period is to be divided (NSTP), and the time step multiplier (TSMULT). The time step multiplier is the ratio of the length of each time step to that of the preceding time step. Using these values, the program calculates the length of the first time step (Δt^1) in the stress period as:

$$\Delta t^1 = \mathrm{PERLEN}\left(\frac{\mathrm{TSMULT} - 1}{\mathrm{TSMULT}^{\mathrm{NSTP}} - 1}\right) \text{ when TSMULT} \neq 1\text{, and}$$

$$\Delta t^1 = \frac{\mathrm{PERLEN}}{\mathrm{NSTP}} \text{ when TSMULT is one.}$$

The length of each successive time step is computed as:

$$\Delta t^{m+1} = \Delta t^{m}\,\mathrm{TSMULT}\ .$$

As stated previously, stress periods are implemented only as a convenience. Packages that define time-dependent stresses read input data every stress period. Stress periods facilitate the frequent need to have constant input data for stresses for multiple time steps. Situations are not unusual, however, in which a need arises to change stress data for every time step. In this situation, each stress period must consist of a single time step.

As an example of how stress periods are used, consider a simulation that uses the River and Well Packages. The simulation is for a period of 90 days, and 90 one-day time steps are used. The well and river cells are the same throughout the simulation, but the pumping rates and river stage vary. The number of stress periods depends on how frequently the river stage and pumping rates vary because a new stress period must start whenever stage or pumping for any cell changes. For example, if river stage or pumping only change every 30 days, then three 30-day stress periods can be used. Likewise, if pumping or river stage varies every 3 days, then 30 three-day stress periods would be used. When a new stress period begins, all stress data must be redefined; however, most stress packages have an option to reuse the data from the previous stress period. In this example, reuse of river data would be useful if a new stress period is started because the pumping rates change while the river stage stays the same.

Simulation of steady state requires a single stress period with one time step. The length of a steady-state stress period does not have an impact on the computed head because the storage term in the flow equation is set to zero by the internal flow package. A length of one is suggested for steady-state simulations, unless a specific stress period length is needed.

Budget Calculations in the Basic Package

The calculation of the overall volumetric budget is carried out in two parts—the calculation of individual budget entries and the summation of all entries. As explained in Chapter 3, the entries, which correspond to individual components of flow, are calculated in the hydrologic packages and stored in the variable VBVL. VBVL is passed to the Basic Package, which sums and writes the budget entries.

Figure 4–4 is an example of the overall volumetric budget for the end of a stress period. When sufficient space is available, the budget values are written with aligned decimal points and without exponents rather than using Fortran's exponential notation. The use of aligned decimal points aids in observing the relative sizes of the values; however, this can result in an inappropriate number of significant digits being written. The number of valid significant digits depends on a variety of factors such as the accuracy of the computed heads and the precision used in the computer to represent numbers. On computers in which single precision real numbers are represented with 32 bits, the maximum possible number of significant digits is usually seven, and the actual number of significant digits is likely to be less.

```
 VOLUMETRIC BUDGET FOR ENTIRE MODEL AT END OF TIME STEP  1 IN STRESS PERIOD   1
 ------------------------------------------------------------------------------

    CUMULATIVE VOLUMES      L**3       RATES FOR THIS TIME STEP      L**3/T
    ------------------                 ------------------------

          IN:                                   IN:
          ---                                   ---
            STORAGE =           0.0000            STORAGE =           0.0000
```

```
       CONSTANT HEAD =             0.0000         CONSTANT HEAD =             0.0000
               WELLS =             0.0000                 WELLS =             0.0000
       RIVER LEAKAGE =             0.0000         RIVER LEAKAGE =             0.0000
            RECHARGE =      13608000.0000              RECHARGE =           157.5000

            TOTAL IN =      13608000.0000              TOTAL IN =           157.5000

          OUT:                                   OUT:
          ----                                   ----
             STORAGE =             0.0000               STORAGE =             0.0000
       CONSTANT HEAD =       4326522.0000         CONSTANT HEAD =            50.0755
               WELLS =       6480000.0000                 WELLS =            75.0000
       RIVER LEAKAGE =       2801081.2500         RIVER LEAKAGE =            32.4199
            RECHARGE =             0.0000              RECHARGE =             0.0000

           TOTAL OUT =      13607603.0000             TOTAL OUT =           157.4954

            IN - OUT =           397.0000              IN - OUT =         4.5929E-03

 PERCENT DISCREPANCY =               0.00   PERCENT DISCREPANCY =               0.00
```

Figure 4–4. Sample of an overall volumetric water budget. (Modified from McDonald and Harbaugh, 1988.)

Output

The primary output of the GWF Process is head distribution. The user may control the frequency at which heads are written to the listing or to separate files through the "Output Control" option of the Basic Package. Other output includes drawdown and budget information; the Output Control option also provides for writing these to listing or separate files. Output Control has its own input file, which is specified in the Name File by file type OC. If Output Control is not utilized, a default output option is invoked—the head distribution and the overall volumetric budget are written to the Listing File at the end of each stress period, and drawdowns are also written to the Listing File.

Zone and Multiplier Arrays for Parameters

As described in Chapter 8, special variables called zone arrays and multiplier arrays can be used to define array parameters. Each multiplier and zone array is given a name, which is used when parameters are defined. The BAS Package reads the zone and multiplier arrays from files specified in the Name File using file types ZONE and MULT, respectively.

Parameter Value File

As described in Chapter 8, some input data for various packages can be defined using parameters. Each parameter has a value that is defined in the input file in which the parameter is defined; however, flexibility has been incorporated to allow all or some parameter values to be specified in a single file, which is called the Parameter Value File. If the Parameter Value File exists, it is read by the Basic Package, and the values in the file replace the values specified when the parameter is defined.