Input instructions for files used by GWM-2005

Released with Version 1.5.0 of GWM-2005, September 2013

Input instructions for files used by GWM-2005 are modified as the capabilities of the code change over time. This document summarizes the current (version 1.5.0) input instructions for files that are either required or are optional for a GWM-2005 Process simulation. The input instructions are based on reports by Ahlfeld and others (2005, 2009, and 2011), Ahlfeld and Barlow (2013), and a description of Drain-Flow State Variables distributed with the code (see file “Drains.pdf” in this directory).

Contents

Modes in which GWM-2005 can be used

Before solving an optimization problem with GWM-2005, the user must already have developed a groundwater-flow model (or both parent and child models if local grid refinement (LGR) capability is being used) of the study area based on the MODFLOW-2005 GWF Process. In its current form, GWM-2005 can only be used with the Groundwater-Flow (GWF) and Observation (OBS) Processes of MODFLOW-2005. GWM-2005 can be used in one of four modes:

- MODFLOW-2005 GWF Process without GWM or LGR capabilities,
- MODFLOW-2005 GWF Process with LGR capability but without GWM,
- GWM with the MODFLOW-2005 GWF Process but without LGR, and
- GWM with the MODFLOW-2005 GWF Process with the LGR capability.

Selection of one of the four modes is made on the basis of the type and contents of the file that is entered by the user in response to an onscreen prompt when GWM-2005 is first executed. Two types of files can be entered—a MODFLOW **NAME** file or an LGR **CONTROL** file, each of which is described in detail below. Implementation of each of the four modes is done as follows:

- A MODFLOW-2005 GWF-Process Simulation without GWM or LGR capabilities. GWM-2005 can be used to run the MODFLOW-2005 GWF Process without either the GWM Process or LGR capability active. In this case, the user enters the name of a MODFLOW **NAME** file. The **NAME** file contains the names of input and output files that are used by the MODFLOW GWF and OBS Processes only. The GWM-2005 Process is not activated.

- A MODFLOW GWF-Process Simulation with LGR capability but without GWM. In this case, a MODFLOW-2005 GWF Process simulation is done with multiple models but without groundwater management. At the GWM-2005 prompt, the user enters the name of an LGR **CONTROL** file. The LGR **CONTROL** file is distinguished from a **NAME** file by the presence of the keyword ***LGR*** as the first noncommented input. LGR reads its input data for the parent and child models from this control file. **NAME** files are specified for each of the parent and child models as part of the LGR **CONTROL** file. The GWM-2005 Process is not activated.

- A GWM Simulation without LGR. In this case, GWM-2005 is used to simulate a groundwater-management problem that consists of a single model covering the model domain. At the GWM-2005 prompt, the user enters the name of a MODFLOW **NAME** file. The **NAME** file contains the names of input and output files that are used by the MODFLOW GWF and OBS Processes as well as the name of the **GWM** file (see input instructions below).

- A GWM Simulation with the LGR capability. In this case, GWM-2005 is used to simulate a groundwater-management problem that consists of multiple models—a parent model and at least one child model. At the GWM-2005 prompt, the user enters the name of an LGR **CONTROL** file. **NAME** files are specified for each of the parent and child models as part of the LGR **CONTROL** file; the name of the **GWM** file is specified in the **NAME** file for the parent model and for any child model that includes components of the management problem.

MODFLOW NAME File

The MODFLOW **NAME** file contains the names of most input and output files used by MODFLOW and determines which MODFLOW program options are activated. The **NAME** file is read on unit 99. When LGR is used, a **NAME** file must be specified for each model in the LGR **CONTROL** file. Each **NAME** file contains one record (line) of information for each input and output file. Each record consists of as many as four variables, which are read in free format; the length of each record must be 299 characters or less. Comment records can be placed anywhere in the **NAME** file and are indicated by the # character in column one. Any text characters can follow the # character. Comment records have no effect on the run; their purpose is to allow users to provide documentation about a particular run.

Each record has the following format:

Ftype   Nunit   Fname   [Fstatus]

The variables are defined as follows:

Ftype—is the file type. Ftype may be entered in all uppercase, all lowercase, or any combination thereof.

Nunit—is the Fortran unit to be used when reading from or writing to the file. Any legal unit number on the computer being used can be specified except 99. Also, the unit number for the file must be unique; that is, it cannot be equal to any of the unit numbers used for other files specified in the **NAME** file.

Fname—is the name of the file, which is a character value. Pathnames may be specified as part of Fname.

Fstatus—is the optional file status, which applies only to file types **DATA** and **DATA(BINARY)** (see Harbaugh, 2005). Two values are allowed: OLD and REPLACE. "OLD" indicates that the file should already exist. "REPLACE" indicates that if the file already exists, then it should be deleted before a new file is opened. The default actions are to open the existing file if the file exists or create a new file if the file does not exist.

The listing file (Ftype = ***LIST***) must be present in each **NAME** file and must be the first file listed in the **NAME** file. If a management model is to be solved, the **NAME** file must include a record that specifies Ftype = ***GWM*** to indicate that the GWM-2005 Process is active. The **GWM** file identified in this record contains information needed for the GWM Process. For a GWM-2005 run that solves a management model and uses LGR, a **GWM** file record must appear in the **NAME** file for the parent model; the **GWM** file record may optionally be included in the **NAME** file for the child models. When present, the ***GWM*** record must appear after the ***LIST*** record. Example input records for the **LIST** and **GWM** file types are

LIST	10	list.gwm
GWM	55	input.gwm

MODFLOW DISCRETIZATION File

When the LGR capability of GWM-2005 is used, the number of stress periods in a simulation (variable NPER), the length of each stress period (variable PERLEN), the number of time steps in each stress period (variable NSTP), and the time-step multiplier for each stress period (variable TSMULT) are the same for all models. These variables must be specified in the **DISCRETIZATION** (**DIS**) input file for each model, but GWM-2005 uses only the values specified in the **DIS** file of the parent model and ignores the values specified in the child model(s).

LGR CONTROL File

At the beginning of a GWM-2005 run, the program prompts the user for either a MODFLOW **NAME** file or an LGR **CONTROL** file. If a MODFLOW **NAME** file is specified, GWM-2005 will simulate a single model; if an LGR **CONTROL** file is specified, GWM-2005 will simulate multiple models, each with its own MODFLOW **NAME** file. The LGR **CONTROL** file is distinguished from a MODFLOW **NAME** file by the presence of the keyword ***LGR*** as the first noncommented input variable. LGR reads its input data from this **CONTROL** file. The LGR capability of MODFLOW is described by Mehl and Hill (2005, 2007, and 2010) and the report by Ahlfeld and others (2009) provides a discussion of considerations for preparing MODFLOW-2005 files for use with LGR. [If the option to route streamflow through locally refined grids with the Streamflow-Routing (SFR2) Package is being used, the user should review the modified input instructions for LGR provided by Mehl and Hill, 2010 (p. 9-12).]

Input for LGR is defined by 15 items, each of which is read in free format:

FOR EACH SIMULATION

0. [#Text]

Item 0 is optional—# must appear in column 1. Item 0 can be repeated multiple times.

1. ***LGR***
2. NGRIDS

FOR THE PARENT MODEL (input items for the parent model must be listed before those for the child model or models)

3. NAME FILE
4. GRIDSTATUS
5. IUPBHSV IUPBFSV

FOR EACH CHILD MODEL (items 6 through 15 are repeated for each child model; the total number of repetitions is NGRIDS – 1)

6. NAME FILE
7. GRIDSTATUS
8. ISHFLG IBFLG IUCBHSV IUCBFSV
9. MXLGRITER IOUTLGR
10. RELAXH RELAXF
11. HCLOSELGR FCLOSELGR
12. NPLBEG NPRBEG NPCBEG
13. NPLEND NPREND NPCEND

14. NCPP
15. NCPPL (repeat NCPPL a total of NPLEND + 1 – NPLBEG times)

The variables are defined as follows:

LGR—is a keyword that must be entered in the first item to indicate to GWM-2005 that the LGR capability is used.

NGRIDS—is the total number of grids used in this simulation, including the parent grid and all of the child grids.

NAME FILE—is the name of the **NAME** file for either the parent or child model. The name can include the file path and is limited to 200 characters.

GRIDSTATUS—is a character variable and indicates whether the file listed in NAME FILE corresponds to a parent or child model. Two options are allowed:

PARENTONLY—the **NAME** file for the parent model; and

CHILDONLY—the **NAME** file for a child model.

IUPBHSV—is an integer value greater than zero that corresponds to the unit number for the file to which boundary heads will be saved for later use by the Boundary Flow and Head (BFH) Package for independent simulations. A file with this unit number needs to be opened in the **NAME** file of the parent model. A value of zero indicates that the file is not written. For the parent model, these are the complementary boundary conditions (see Appendix 2 of Mehl and Hill, 2005).

IUPBFSV— is an integer value greater than zero that corresponds to the unit number for the file to which boundary fluxes will be saved for later use by the BFH Package for independent simulations. A file with this unit number needs to be opened in the **NAME** file of the parent model. A value of zero indicates that the file is not written. For the parent model, these are the coupling boundary conditions (see Appendix 2 of Mehl and Hill, 2005).

ISHFLG—is a flag indicating whether heads from the parent-model simulation should be used as the starting heads for the child-model simulation(s). These heads apply to the interior of the child grid(s), not the boundary heads; for interior child-grid cells that overlap the parent grid, the heads of the corresponding parent cell are used. Two options are allowed:

ISHFLG = 1. Use heads from the parent-model simulation as the starting heads for the child model. No interpolation is applied. For steady-state simulations, this procedure can provide a good initial estimate of the head distribution, and thus reduce computational time. For transient simulations, the procedure overwrites the initial condition of the child model defined in STRT of the Basic Package input file (Harbaugh, 2005) and therefore is not recommended.

ISHFLG = 0. Use the heads defined in STRT of the Basic Package for the child model.

IBFLG—is a positive integer used to define the interface of the child grid with the parent grid. Use this value around the perimeter of the child model IBOUND array. Do not use IBFLG or –IBFLG anywhere else in the parent or child IBOUND arrays. Use a unique value for each child grid.

IUCBHSV—is an integer value greater than zero that corresponds to the unit number for the file to which boundary heads will be saved for later use by the BFH Package for independent simulations. A file with this unit number needs to be opened in the **NAME** file of the child model. A value of zero indicates that the file is not written. For the child model, these are the coupling boundary conditions (see Appendix 2 of Mehl and Hill, 2005).

IUCBFSV— is an integer value greater than zero that corresponds to the unit number for the file to which boundary fluxes will be saved for later use by the BFH Package for independent simulations. A file with this unit number needs to be opened in the **NAME** file of the child model. A value of zero indicates that the file is not written. For the child model, these are the complementary boundary conditions (see Appendix 2 of Mehl and Hill, 2005).

MXLGRITER—is the maximum number of LGR iterations. Twenty iterations are sufficient for most problems. See “Closure Criteria for LGR Iterations” section of the report by Mehl and Hill (2005). Set MXLGRITER to 1 for a one-way coupling (see Mehl and Hill, 2005).

IOUTLGR—is an integer value that is a flag that controls printing of the maximum head and flux change for each LGR iteration. For the maximum head change, the head value and corresponding layer, row, and column of the child model is listed. For the maximum flux change, the flux value and corresponding layer, row, and column of the parent model is listed. If IOUTLGR < 0, output is written to the screen; if IOUTLGR > 0, output is written to the child listing file; and if IOUTLGR = 0, no results are written.

RELAXH, RELAXF—are real numbers equal to the relaxation factors for heads and fluxes, respectively. Values of RELAXH and RELAXF less than 1.0 and greater than zero are needed for convergence of the LGR iterations. Typically, values of about 0.5 produce convergent solutions. Values less than 0.5 may be needed when the LGR iterations have difficulty converging. In cases in which the LGR iterations converge, values greater than 0.5 may reduce the number of iterations needed for convergence. Convergence problems can be diagnosed by printing the maximum head and flux changes (IOUTLGR ≠ 0) to determine if the head and flux changes are decreasing (converging) or increasing (diverging) as the LGR iterations proceed.

HCLOSELGR—is a real number equal to the head closure criterion for the LGR iterations. The closure criterion is based on heads for the child-interface nodes. This closure criterion is satisfied when the maximum absolute head change between successive LGR iterations is less than HCLOSELGR (see equation 8b in Mehl and Hill, 2005).

FCLOSELGR—is a real number equal to the flux closure criterion for the LGR iterations. The closure criterion is based on fluxes into the parent-interface nodes. This closure criterion is satisfied when the maximum absolute relative flux change between successive LGR iterations is less than FCLOSELGR (see equation 8a in Mehl and Hill, 2005).

NPLBEG—is the layer number of the parent grid in which refinement (that is, the child model) begins. Refinement must begin in the top layer of the model, so NPLBEG must equal 1.

NPRBEG—is the row number of the parent grid in which refinement begins (cannot equal 1).

NPCBEG—is the column number of the parent grid in which refinement begins (cannot equal 1).

NPLEND—is the layer number of the parent grid in which refinement ends. NPLEND must be greater than or equal to NPLBEG.

NPREND—is the row number of the parent grid in which refinement ends. NPREND must be greater than NPRBEG, and NPREND cannot equal the number of rows in the parent grid.

NPCEND—is the column number of the parent grid in which refinement ends. NPCEND must be greater than NPCBEG, and NPCEND cannot equal the number of columns in the parent grid.

NCPP—is the number of child cells that span the width of a single parent cell along rows and columns. This must be an odd integer greater than 1, and is applied to both rows and columns.

NCPPL—is the number of child cells that span the depth of each parent layer. Each value must be an odd integer greater than or equal to 1. One value is read for each refined parent layer. The number of values needs to equal (NPLEND + 1 – NPLBEG). Values can be 1 (which results in no vertical refinement for the layer) only in layers above the bottom of the child model, unless refinement extends all the way to the bottom of the parent model. For refinement that does not extend to the bottom of the parent model, the refinement terminates at the shared node. For example, for the simulation condition shown in figure 2b, the values 5 3 would be entered for NCPPL.

GWM Process Files

Input files for the GWM Process consist of the **GWM** file and several supporting files. The GWM Process is activated by placing **GWM** file type in the MODFLOW **NAME** file along with the name of the **GWM** file. The **GWM** file, in turn, contains keywords and filenames for the components of the management problem. When the management problem is solved for multiple models, an LGR **CONTROL** file is required. It will include the **NAME** files for each model. The **NAME** files may each contain **GWM** files, with each **GWM** file containing references to the files that contain components of the management problem for the corresponding model.

Four types of information about the management problem are specified in the input files referenced in the **GWM** file: the decision and state variables, objective function, and constraints of the management problem and the solution and output-control parameters. GWM-2005 reserves Fortran unit 99 for input purposes. In contrast to the MODFLOW **NAME** file, unit numbers are not required for the files named in the **GWM** file. Instead, the program automatically determines available unit numbers and uses them for input and output.

For the most part, the general structure of the input formats for the GWM Process files are consistent with those for other MODFLOW processes; users of GWM should review the input instructions for MODFLOW given in Harbaugh (2005). Input for each GWM Process file is grouped by numbered items, and each item consists of input variables. The first item in each of the input files is Item 0, which can be used for comment lines but is optional. Some items consist of several variables, and the item can be repeated multiple times. The input data for each item must start on a new record. Each record is limited to a length of 199 characters. An input variable may include a single value or multiple values. Variables are defined after all the items are listed.

Each input variable has a data type, which can be Real, Integer, or Character. Integers are whole numbers and must not include a decimal point or exponent. Real numbers can include a decimal point and an exponent; if no decimal point is included in the entered value, then the decimal point is assumed to be at the right side of the value. Any printable character is allowed for character variables. Unlike the GWF Process, variables used by GWM that start with the letters I-N are not necessarily integers and those that start with the letters A-H and O-Z are not necessarily real numbers. Data types are specified for each input variable.

Free formatting is used for GWM input. With free formatting, values are not required to occupy a fixed number of columns in a record. Each value can occupy one or more columns as required to represent it; however, the values must still be included in the prescribed order. One or more spaces or a single comma optionally combined with spaces must separate adjacent values. Also, a numeric value of zero must be explicitly represented by 0 and not by one or more spaces.

Units of values used in the GWM Process should be consistent with the units used in the other MODFLOW data-input files.

GWM File

The **GWM** file is formatted in a manner similar to the MODFLOW **NAME** file. A series of records are read that have the following format:

Ftype Fname

Ftype is one of several keywords, and Fname is a path name of the relevant computer file. Except for keyword ***OUT***, each of the keywords triggers the reading of a file that will be referred to with the same name as the keyword. Unit numbers are determined internally by GWM and do not need to be specified in the **GWM** file. The entire record including the Fname entry is limited to 199 characters in length. Comment lines may appear anywhere in the **GWM** file and are indicated by the # character in the first column of the record.

Keywords can be specified in either uppercase or lowercase letters. Keywords may appear in any order except (1) the ***OUT*** keyword must be the first keyword in the file if it is used, and (2) the ***STAVAR*** keyword must follow the ***DECVAR*** keyword if state variables are defined for the problem. The keywords that are suitable for inclusion in a **GWM** file depend on the type of problem. If the problem is a single model (that is, a simulation without local grid refinement), then only a single **GWM** file is provided. If the problem is multimodel (with local grid refinement), then a **GWM** file is required for the parent model and may be provided for child models. The following keywords are available in GWM-2005:

OUT—a filename for all output from the GWM Process may be assigned here. If the ***OUT*** keyword is not specified, a default name of “GWM.OUT” will be used, and the output file will be written to the directory in which program execution occurs. The ***OUT*** keyword is not allowed if the **GWM** file is for a child model.

DECVAR—the Fname associated with this keyword identifies the **DECVAR** file that provides information about the decision variables. For single-model problems, the ***DECVAR*** keyword is required. For multimodel problems, a **DECVAR** file is provided for every model that includes decision variables. The **GWM** file for at least one model, although not necessarily the parent model, must contain a ***DECVAR*** keyword.

STAVAR—the Fname associated with this keyword identifies the **STAVAR** file that provides information about the state variables. A **STAVAR** file is optional, but if it is listed, it must follow the **DECVAR** file record. For multimodel problems, a **STAVAR** file is provided for every model that includes state variables.

OBJFNC—the **OBJFNC** file provides information about the objective function. The ***OBJFNC*** keyword must appear in the **GWM** file for single-model problems and in the parent model **GWM** file for multimodel problems. The ***OBJFNC*** keyword is not allowed in the **GWM** files of child models of multimodel problems.

VARCON—the **VARCON** file provides information on the lower and upper bounds specified for the decision variables defined in the **DECVAR** file. If the ***DECVAR*** keyword appears in a **GWM** file, then the ***VARCON*** keyword must also appear.

SUMCON, HEDCON, and ***STRMCON***—the **GWM** file may include up to three additional files that provide information about summation constraints, head constraints, and streamflow constraints that are allowed in GWM. None of these keywords are required in a **GWM** file. For multimodel problems, the ***SUMCON*** keyword can appear only in the parent model, whereas the ***HEDCON*** and ***STRMCON*** keywords may appear in the **GWM** files for parent or child models.

SOLN—the **SOLN** file provides information about the solution and output-control parameters. The ***SOLN*** keyword must appear in the **GWM** file for single-model problems and in the parent model **GWM** file for multimodel problems. The ***SOLN*** keyword is ignored in the **GWM** files of child models of multimodel problems.

The requirements for GWM files and keywords are summarized in table 1 and as follows. When the LGR capability is not active, each GWM run requires specification of a single GWM file. The GWM file must contain the **DECVAR**, **OBJFNC**, **VARCON**, and **SOLN** file types; the **OUT**, **STAVAR**, **SUMCON**, **HEDCON**, and **STRMCON** file types are optional. When the LGR capability is active, a **GWM** file must be in the **NAME** file for the parent model. The parent **GWM** file specifies files that are universal to solving the GWM problem, namely, the **OBJFNC** and **SOLN** files. The parent-model **GWM** file also may optionally include **OUT**, **DECVAR**, **STAVAR**, **VARCON**,

Table 1. GWM file requirements for simulations with and without Local Grid Refinement (LGR).

File Type	Simulation without LGR	Simulation with LGR	
		Parent Model	Child model(s)
GWM	Required	Required	Optional
DECVAR	Required	Optional[1]	Optional[1]
STAVAR	Optional	Optional	Optional
OBJFNC	Required	Required	None
VARCON	Required	Optional[2]	Optional[2]
SUMCON	Optional	Optional[3]	None
HEDCON	Optional	Optional	Optional
STRMCON	Optional	Optional	Optional
SOLN	Required	Required	None
OUT	Optional	Optional	None

[1]At least one **DECVAR** file and associated **VARCON** file must be specified, in either the parent or child models or both.
[2]A **VARCON** file must be specified if a **DECVAR** file is specified for the model.
[3]Constraints specified in a **SUMCON** file in the parent model may reference decision variables defined on the parent grid or any of the child grids.

SUMCON, **HEDCON**, and **STRMCON** file types (a **VARCON** file must be specified if a **DECVAR** file is specified). The **GWM** file should be specified for a child model only if the child model contains decision or state variables, head constraints, or streamflow constraints. The **GWM** file for each child model may include **DECVAR**, **STAVAR**, **VARCON**, **HEDCON**, and **STRMCON** file types (a **VARCON** file must be specified if a **DECVAR** file is specified); file types **OUT**, **OBJFNC**, **SUMCON**, and **SOLN** may not be specified in a child **GWM** file. At least one **GWM** file, in either the parent or child model, must contain a **DECVAR** file and associated **VARCON** file. If summation constraints are included in the problem, the **SUMCON** file must be listed in the **GWM** file for the parent model, although constraints may reference decision variables defined on any grid.

Decision Variable (DECVAR) File

This file is used to define the three types of decision variables that may be defined in a management model. The primary decision variables are the flow rates (either withdrawal or injection) at each managed well site. Flow-rate decision variables may be of WEL-type (single node) or MNW-type (multi-node) wells. Flow-rate decision variables must be defined in the **DECVAR** file that is associated with the grid in which the flow is simulated. A single well site may have more than one flow- rate decision variable associated with it. Moreover, a single flow-rate decision variable can extend over one or more model cells and can be active during one or more stress periods. The cells that make up each flow-rate decision variable, however, must be completely within either the parent grid or one of the child grids; the cells that contain flow-rate decision variables cannot cross grid interfaces. Also, if the **DECVAR** file includes any MNW-type decision variables, an MNW2 input file must be listed in the **NAME** file.

The second type of decision variable is an external variable. External variables do not have a direct effect on the system state variables and are not assigned to a specific location in the model; for this reason, they can be defined in the **DECVAR** file(s) for any of the parent or child models. The third type of decision variable is a binary variable used to define the status of each flow-rate or external decision variable as active (for example, the site is constructed) or inactive (the site is not constructed). Binary variables have a value of 0 (inactive) or 1 (active) in the solution of the management formulation. One or more flow-rate and external decision variables are associated with each binary variable. These associated variables may come from any of the flow-rate or external decision variables defined in the parent and child **DECVAR** files.

The **DECVAR** file includes five input items:

0. [#Text]

Item 0 is optional—# must appear in column 1. Item 0 can be repeated multiple times.

1. IPRN GWMWFILE
2. NFVAR NEVAR NBVAR

3a. The following records are read for each of the NFVAR flow-rate decision variables:

FVNAME NC LAY ROW COL FTYPE FSTAT WSP

<u>Options for specifying NC, LAY, ROW, and COL</u>: Input variable NC is an integer variable that describes the type and number of wells that are associated with a flow-rate decision variable. In previous versions of GWM, NC always referred to WEL-type wells. With the addition of the MNW functionality, NC can refer to either a WEL-type or MNW-type well: if NC is specified as a positive integer, then the flow-rate decision variable consists of one or more WEL-type wells; if NC is specified as a negative integer, then the flow-rate decision variable consists of one or more MNW-type wells. A flow-rate

decision variable cannot consist of a mix of WEL-type and MNW-type wells; it must be either all WEL-type or all MNW-type wells. Also, NC cannot be specified as zero (if it is, GWM will write an error message and stop execution).

Four options are provided for specifying NC. The first two options are used for WEL-type and the second two options for MNW-type flow-rate decision variables. Depending on the value specified for NC, the user must optionally complete items 3b to 3e, which are described below.

- If NC = 1, then the flow-rate decision variable consists of a single WEL-type withdrawal or injection well at a single model cell defined by LAY, ROW, COL. The user must complete item 3a, skip items 3b through 3e, and then proceed to the next flow-rate decision variable (that is, to item 3a); or, if this is the last variable to be defined, proceed to items 4 and 5.
- If NC ≥ 2, then the flow-rate decision variable consists of a total of NC WEL-type withdrawal or injection wells at model cells that must be defined in item 3b-1 below. The flow rate calculated for this decision variable is distributed over the NC cells in proportion to the values specified by RATIO for each cell in item 3b-1. Item 3b-1 is read NC times. Values must be specified for variables LAY, ROW, and COL in item 3a, but the specified values are ignored by GWM (for example, the user could specify values of zero for each of the three variables). The user must complete items 3a and 3b-1, skip items 3b-2 through 3e, and then proceed to the next flow-rate decision variable (that is, to item 3a); or, if this is the last variable to be defined, proceed to items 4 and 5.
- If NC = -1, then the flow-rate decision variable consists of a single MNW-type withdrawal or injection well. Values must be specified for variables LAY, ROW, and COL in item 3a, but the specified values are ignored by GWM (for example, the user could specify values of zero for each of the three variables). The user must complete items 3b-2 and 3c through 3e (and skip items 3b-1 and 3b-3) and then proceed to the next flow-rate decision variable (that is, to item 3a); or, if this is the last variable to be defined, proceed to items 4 and 5.
- If NC ≤ -2, then the flow-rate decision variable consists of a total of the absolute value of NC MNW-type withdrawal or injection wells. The flow rate calculated for this decision variable is distributed over the absolute value of NC multi-node wells in proportion to the values specified by RATIO for each well. Values must be specified for variables LAY, ROW, and COL in item 3a, but the specified values are ignored by GWM (for example, the user could specify values of zero for each of the three variables). The user must complete items 3b-3 through 3e (and skip items 3b-1 and 3b-2) for each of the multi-node wells associated with this decision variable, and then proceed to the next flow-rate decision variable (that is, to item 3a); or, if this is the last variable to be defined, proceed to items 4 and 5.

3b. One of the following input items must be completed if NC is not equal to 1:

3b-1. If NC ≥ 2, repeat the following item NC times: RATIO LAY ROW COL
3b-2. If NC = -1: WELLID NNODES
3b-3. If NC ≤ -2: RATIO WELLID NNODES

3c. LOSSTYPE PPFLAG

3d. If LOSSTYPE = NONE, then skip item 3d and proceed to item 3e. If LOSSTYPE ≠ NONE, then include item 3d by specifying values for one or more of the following variables depending on the type of LOSSTYPE specified in item 3c:

{Rw Rskin Kskin B C P CWC}
If LOSSTYPE = THIEM, specify Rw.

If LOSSTYPE = SKIN, specify Rw, Rskin, and Kskin.
If LOSSTYPE = GENERAL, specify Rw, B, C, and P.
If LOSSTYPE = SPECIFYcwc, then specify CWC.

Any of the variables in item 3d can be assumed to be constant for the entire length of the open interval of the well (in which case appropriate values are simply specified here in item 3d), or they can be assumed to vary along the length of the open interval of the well (in which case any negative value should be specified here in item 3d and the actual values then specified for each node or open interval in item 3e). For example, if Rw is specified as -1, then Rw is assumed to vary along the length of the well and a real value of Rw must be defined for each node (or open interval) of this well in item 3e.

3e. The user must enter location information for each multi-node well in item 3e. For each node (or open interval) of this well, the user must use input format 3e-1 or 3e-2, depending on the value of NNODES. In either case, item 3e must consist of a total number of records equal to the absolute value of NNODES. If the relevant LOSSTYPE variables were set to negative values in item 3d, then they vary in value among nodes (or open intervals, if NNODES < 0) and should be defined here in item 3e according to the definitions given under item 3d. The values specified here, if any, are those that were set to a negative value in item 3d.

For NNODES > 0, enter one line of input data for each of the NNODES model cells (nodes) for the current well:
3e-1. LAY ROW COL {Rw Rskin Kskin B C P CWC PP}

The data list of nodes defining the multi-node well must be constructed and ordered so that the first node listed represents the node closest to the wellhead, the last node listed represents the node farthest from the wellhead, and all nodes are listed in sequential order from the top to the bottom of the well. A particular node in the grid can be associated with more than one multi-node well.

For NNODES < 0, the absolute value of NNODES indicates how many open intervals are to be defined, and so must correspond exactly to the number of records in item 3e-2 for this well:
3e-2. Ztop Zbotm ROW COL {Rw Rskin Kskin B C P CWC PP}

The data list of intervals defining the multi-node well must be constructed and ordered so that the first interval listed represents the shallowest one, the last interval listed represents the deepest one, and all intervals are listed in sequential order from the top to the bottom of the well. If an interval partially or fully intersects a model layer, then a node will be defined in that cell. If more than one open interval intersects a particular layer, then a length-weighted average of the cell-to-well conductances will be used to define the well-node characteristics; for purposes of calculating the effects of partial penetration, the cumulative length of the well screens will be assumed to be centered vertically within the thickness of the cell. If the well is a single-node well by definition of LOSSTYPE = NONE and the defined open interval straddles more than one model layer, then the well will be associated with the cell that contains the center of the open interval. If the specified elevations indicate multiple well screens or open intervals within a single model layer, then the model will assign a single composite well screen for the node based on the ratio of the screen length to layer thickness (see discussion in Konikow and others, 2009). However, if CWC values are specified by the user, then these values are assumed to be already appropriate for the actual length of the screen and are not adjusted by this ratio.

4. The following record is read for each of NEVAR external decision variables:
   EVNAME ETYPE ESP
5. The following record is read for each of NBVAR binary decision variables:
   BVNAME NDV BVLIST

The variables are defined as follows:

Text—is a character variable up to 199 characters long that starts in column 2. Any characters can be included in Text. Lines beginning with # are restricted to the first lines of the file. Text is printed when the file is read.

IPRN—is an integer variable that describes the amount of output that is written to the **GWM OUT** file. IPRN must be specified as either 0 or 1. When IPRN equals 0, a minimum amount of information about the decision variables is written to the GWM output file; when IPRN equals 1, detailed information about the decision variables is written to the GWM output file.

GWMWFILE—is an integer variable equal to the unit number of the file to which the values of the flow-rate decision variables at the optimal solution will be written. If GWMWFILE is set to zero, no output will be written. The flow rates are written in a format similar to the format used in the MODFLOW well (**WEL**) file. The unit number should be associated with a ***DATA*** file type defined in the MODFLOW **NAME** file. (See section "Writing of Flow-Rate Decision Variables in the Optimal Solution to a MODFLOW Well-Type File" for more details.)

NFVAR—is an integer variable equal to the total number of flow-rate decision variables simulated by either the WEL- or MNW-type functionalities. NFVAR must be greater than 0. Only one flow-rate decision variable can be defined for a particular combination of well locations and stress periods associated with the decision variable, with the exceptions that both a withdrawal variable (FTYPE = W) and an injection variable (FTYPE = I) can be defined simultaneously for the same combination of well locations and stress periods.

NEVAR—is an integer variable equal to the number of external decision variables. NEVAR must be greater than or equal to 0.

NBVAR—is an integer variable equal to the number of binary variables. NBVAR must be greater than or equal to 0. If NBVAR is 0, binary variables are not included in the management formulation.

FVNAME—is a character variable up to 10 characters long that is a unique name designated for the flow-rate decision variable. Each name must be unique (that is, the same name cannot be used for more than one variable or in more than one model when the local-grid refinement capability of GWM is used). No spaces are allowed in the name. The end of the name is designated by a blank space.

NC—is an integer variable that describes the type and number of wells that are associated with a flow-rate decision variable. See the discussion above to determine the correct value to specify for NC.

LAY, ROW, and COL—are integer variables equal to the layer, row, and column number of the model cell or multi-node well to which the flow for decision-variable FVNAME will be assigned. See the discussion above to determine the correct values to specify for LAY, ROW, and COL.

FTYPE—is a character variable that indicates whether the decision variable is a withdrawal or injection site. If FTYPE is W, the site is used for withdrawal; if FYPTE is I, the site is used for injection. If either withdrawal or injection is allowed at the site, two decision variables must be defined for the site, one for withdrawal (that is, with FTYPE=W) and one for injection (FTYPE=I).

FSTAT—is a character variable that indicates whether the decision variable will be considered in the management problem. If FSTAT is Y, the decision variable is available; if FSTAT is N, the decision variable is unavailable. If the decision variable is unavailable, then no withdrawal or injection will be calculated at the decision-variable location. For linear-optimization problems, FSTAT can be used to remove a well from the candidate set of decision variables without having to recalculate the response matrix (in this case, IRM=0; see instructions in the section "Solution and Output-Control Parameters (SOLN) File" below).

WSP—is a character string up to 120 characters long that indicates the stress periods associated with decision variable FVNAME. A single flow rate will be determined by GWM for all the stress periods included in WSP. The string must not contain any blank spaces. Multiple stress periods are indicated by colons (:), hyphens (-), or combinations of colons and hyphens. For example,

1 indicates that stress period 1 is the only stress period associated with the decision variable,

1:3 indicates that the flow rate is the same for stress periods 1 and 3 (but not 2), and

1-12 indicates that the flow rate is the same for stress periods 1 through 12, and

2:5-7:12 indicates that the flow rate is the same for stress periods 2, 5, 6, 7, and 12.

RATIO—is a real variable. RATIO is the fraction of the total flow rate for decision variable FVNAME that is distributed to each individual cell or multi-node well associated with decision variable FVNAME. The sum of the RATIO values must equal 1.0 for all of the NC cells or multi-node wells specified for FVNAME; if the sum does not equal 1.0, GWM calculates the fraction for each cell or multi-node well by dividing the RATIO value specified for each cell or multi-node well by the sum of the RATIO values specified for all cells or multi-node wells associated with FVNAME.

WELLID—is a character variable that is a unique alphanumeric identification label for each multi-node well. The text string is limited to 20 alphanumeric characters. If the name of the well includes spaces, then enclose the name in quotation marks.

NNODES—is an integer variable equal to the number of cells (nodes) associated with this multi-node well. NNODES normally is specified to be > 0, but for the case of a vertical borehole, setting NNODES < 0 will allow the user to specify the elevations of the tops and bottoms of well screens or open intervals (rather than grid layer numbers). The absolute value of NNODES equals the number of open intervals (or well screens) to be specified in dataset 3e. If this option is used, then the model will compute the layers that contain the open intervals, the lengths of the open intervals, and the relative vertical positions of the open intervals within each model layer (for example, see figure 14 and related discussion in Konikow and others, 2009).

LOSSTYPE—is a character variable used to determine the user-specified model for well loss (equation 1). Available options are

NONE—there are no well corrections and the head in the well is assumed to equal the head in the cell. This option ($h_{well} = h_n$) is valid only for a single-node well (NNODES = 1).

THIEM—this option allows for only the cell-to-well correction based on the Thiem (1906) equation; the head in the well is determined from equation 1 in Ahlfeld and Barlow (2013) as $h_{well} = h_n + AQ_n$, and the model computes A on the basis of the user-specified well radius (Rw, defined below). Coefficients B and C in equation 1 are automatically set to zero.

SKIN—this option allows for formation damage or skin corrections at the well: $h_{well} = h_n + AQ_n + BQ_n$, where A is determined by the model from the value specified for the variable Rw (defined below), and B is determined by the model from the values specified for variables Rskin and Kskin (defined below).

GENERAL—head loss is defined with coefficients A, B, and C and power exponent P ($h_{well} = h_n + AQ_n + BQ_n + CQ_n^P$). Coefficient A is determined by the model from the value specified for the variable Rw (defined below). User must also specify variables B, C, and P, which are defined below. A value of P = 2.0 is suggested if no other data are available (the model allows $1.0 \leq P \leq 3.5$). Entering a value of $C = 0$ will result in a linear model in which the value of B is entered directly (rather than entering properties of the skin, as with the SKIN option).

SPECIFYcwc—the user specifies an effective conductance value (equivalent to the combined effects of the A, B, and C well-loss coefficients expressed in equation 15 in Konikow and others, 2009) between the well and the cell representing the aquifer through variable CWC (defined below). If there are multiple screens within the grid cell or if partial-penetration corrections are to be made, then the effective value of CWC for the node may be further adjusted automatically by MNW2.

PPFLAG—is an integer variable that determines whether the calculated head in the well will be corrected for the effect of partial penetration of the well screen in the cell. If PPFLAG = 0, then the head in the well will not be adjusted for the effects of partial penetration. If PPFLAG > 0, then the head in the well will be adjusted for the effects of partial penetration if the section of well containing the well screen is vertical (as indicated by identical row-column locations in the grid). If NNODES < 0 (that is, the open intervals of the well are defined by top and bottom elevations), then the model will automatically calculate the fraction of penetration for each node and the relative vertical position of the well screen. If NNODES > 0, then the fraction of penetration for each node must be defined in dataset 3e, and the well screen will be assumed to be centered vertically within the thickness of the cell (except if the well is in the uppermost unconfined model layer, in which case the bottom of the well screen will be assumed to be aligned with the bottom boundary of the cell, and the assumed length of well screen will be based on the initial head in that cell).

Rw—is a real variable equal to the radius of the well (L).

Rskin—is a real variable equal to the radius to the outer limit of the skin (L).

Kskin—is a real variable equal to the hydraulic conductivity of the skin (L/T).

B—is a real variable equal to the linear well-loss coefficient in equation 1 of Ahlfeld and Barlow (2013) (T/L^2).

C—is a real variable equal to the nonlinear well-loss coefficient in equation 1 of Ahlfeld and Barlow (2013) ($T^P/L^{(3P-1)}$).

P—is a real variable equal to the power (exponent) of the nonlinear discharge component of well loss in equation 1 of Ahlfeld and Barlow (2013) (dimensionless).

CWC—is a real variable equal to the cell-to-well hydraulic conductance in equation 15 in Konikow and others (2009) (L^2/T).

PP—is a real variable equal to the fraction of partial penetration for this cell (see PPFLAG). Only specify if PPFLAG > 0 and NNODES > 0.

Ztop, Zbotm—are real variables equal to the top and bottom elevations of the open intervals (or screened intervals) of a vertical well.

EVNAME—is a character variable up to 10 characters long that is a unique name designated for the external decision variable. Each name must be unique (that is, the same name cannot be used for more than one variable, or in more than one model). No spaces are allowed in the name. The end of the name is designated by a blank space.

ETYPE—is a character variable that indicates the external variable type. ETYPE can be assigned one of seven values: IM defines the external variables as a source (import) or water; EX defines the variable as a sink (export) or water; HD defines the variable as a head type; SF defines the variable as a streamflow type; ST defines the variable as a storage type; DR defines the variable as a drain-flow type; and GN defines the external variable as a general type. Any combination of external-variable types can be used in a management problem. The value of ETYPE is used by GWM in the output for the external variable. Regardless of the variable type, all external variables are treated as positive-valued variables.

ESP—is a character string up to 120 characters long that indicates the stress periods associated with external variable EVNAME. A single value of the external variable will be determined by GWM for all of the stress periods included in ESP. The total time during which the external variable is active is determined by summing the durations of all the stress periods in ESP. This total time is applied to the objective function only if FNTYP is specified as WSDV in the **OBJFNC** file (eq. 2a in Ahlfeld and others, 2011). The string must not contain any blank spaces. Multiple stress periods are indicated by colons (:) or hyphens (-). For example,

1 indicates that stress period 1 is the only stress period associated with the decision variable,

1:3 indicates that the flow rate for the external variable is the same for stress periods 1 and 3 (but not 2), and

1-12 indicates that the flow rate for the external variable is the same for stress periods 1 through 12.

BVNAME—is a character variable up to 10 characters long that is a unique name designated for the binary decision variable. Each name must be unique over the parent model and all child models. The use of BVNAME, NDV, and BVLIST allows the user to associate one or more FVNAME or EVNAME decision variables with a single binary-variable identifier. For example, the user may want to define 12 decision variables as the monthly withdrawal rates at a single well site. If any one of the 12 decision variables is selected in the optimal solution, then an installation cost associated with the binary variable for the well site must be incurred.

NDV—is an integer variable equal to the number of flow-rate or external decision variables associated with BVNAME.

BVLIST—is a list of the flow-rate and external decision variables associated with binary variable BVNAME. The list is drawn from the character names of these variables, FVNAME and EVNAME, defined in records 3a and 4 in any parent or child **DECVAR** file. Each character variable in the list must be separated by a space, and there must be a total of NDV variables listed. The list can extend over multiple lines; a space followed by the character "&" at the end of an input line instructs GWM to read the following line as a continuation of the list. The list can include any combination of decision variables, irrespective of well-site locations, stress period, or grid on which the decision variable has been defined.

State Variables (STAVAR) File

This optional file is used to define the state variables of the management problem. State variables represent system state and can be used in the objective function or in linear-summation constraints. Head, streamflow, change-in-aquifer-storage (storage), and drain-flow variables are the types of state variables currently supported in GWM-2005. Head and streamflow state variables are associated with a specified location and are evaluated at the end of a specified stress period. Drain-flow variables are associated with one or more drain cells that have been specified in a Drain Package input file during a stress period or periods when either a drain-flow rate or volume will be determined. Head, streamflow, and drain-flow state variables must be defined in the **STAVAR** file associated with the parent or child grid within which the state variable is located.

GWM allows heads at managed MNW-type wells to be defined as head-type state variables by use of a **STAVAR** input file. The MNW wells must be defined as flow-rate decision variables in the **DECVAR** file for the management formulation. Head-type state variables defined at MNW-type wells are included with other head-type state variables under the NHVAR input variable in item 2 below. There are two options for item 3: head-type state variables specified at model cells are defined in item 3a, whereas head-type state variables specified for MNW wells are defined in item 3b. The variable MFLG in item 3b is a flag that must be set to zero and indicates to GWM that the line of input data is formatted as item 3b (head-type state variable at an MNW well). The variable WELLID is the name of an MNW well that has been defined in the **DECVAR** file for the management formulation; GWM uses this name to find the location in the flow-process model where the constraint will be defined.

Storage state variables record the change in aquifer storage within a specified region of the model domain over a specified time period. The specified time period is defined by starting and ending stress periods. The time period extends from the beginning of the starting stress period, defined by the SPSTRT input variable, to the end of the ending stress period, defined by the SPEND input variable. The specified region can be any set of model cells selected by the user. It may include the entire model domain or be limited to a portion of the domain, as specified in the CZONE input variable. When the storage state variable is associated with only a portion of the domain, the portion is defined cell-by-cell using the NSVZL, LNUM, and SVZONE input variables. When a multigrid model is used, the **STAVAR** file can be provided for each grid. Storage state variables may be defined over the entirety or just a portion of each individual grid. Storage state variables also can be defined that include change in storage in multiple grids. This is accomplished by assigning the same storage state variable name, SVNAME, in each of the **STAVAR** files that include the state variable. For example, if the storage state variable is intended to cover the entire model domain, an **STAVAR** file would be provided for each grid and would contain an entry specifying the same SVNAME.

The **STAVAR** file consists of six input items:

0. [#Text]
   Item 0 is optional—# must appear in column 1. Item 0 can be repeated multiple times.
1. IPRN
2. NHVAR NRVAR NSVAR NDVAR

3a. The following record is read for each of the NHVAR head state variables:
   SVNAME LAY ROW COL SVSP

3b. The following record is read for each of the NHVAR head state variables:
   SVNAME MFLG WELLID SVSP

4. The following record is read for each of the NRVAR streamflow state variables:
   SVNAME SEG REACH SVSP [FL_OR_LK]

5a. The following record is read for each of the NSVAR storage state variables:
   SVNAME SPSTRT SPEND CZONE

5b. The following record is read if CZONE is assigned a value of “ZONE”:
   NSVZL

5c-d. The following two records are read NSVZL times:
   LNUM
   SVZONE (Read using the MODFLOW U2DINT utility subroutine; see Harbaugh, 2005, p. 8-57 through 8-59)

6a. The following record is read for each of the NDVAR drain state variables:

SVNAME NMDCEL DRNTYP SVSP SPSTRT SPEND

6b. The following record is read NMDCEL times:

LAY ROW COL [Aux-value]

The variables are defined as follows:

Text—is a character variable up to 199 characters long that starts in column 2. Any characters can be included in Text. Lines beginning with # are restricted to the first lines of the file. Text is printed when the file is read.

IPRN—is an integer variable that describes the amount of output written to the **GWM OUT** file. IPRN must be specified as either 0 or 1. When IPRN equals 0, a minimum amount of information about the decision variables is written to the GWM output file; when IPRN equals 1, detailed information about the decision variables is written to the GWM output file.

NHVAR—is an integer variable equal to the total number of head-type state variables specified at cells or MNW-type managed wells. NHVAR must be greater than or equal to 0.

NRVAR—is an integer variable equal to the number of streamflow-type state variables. NRVAR must be greater than or equal to 0.

NSVAR—is an integer variable equal to the number of storage-type state variables. NSVAR must be greater than or equal to 0.

NDVAR—is an integer variable equal to the number of drain state variables. NDVAR must be greater than or equal to 0.

SVNAME—is a character variable up to 10 characters long that is a unique name designated for the state variable. The names for the head, streamflow, and drain state variables must be unique (that is, the same name cannot be used for more than one variable, or in more than one model when the local-grid refinement capability of GWM is used). For storage state variables applied to multimodel problems (that is, those using local grid refinement), the same name may appear in more than one **STAVAR** file to define a storage state variable that extends over multiple grids. However, in a given **STAVAR** file on one grid of a multimodel problem, the state variable name must be unique. No spaces are allowed in the name. The end of the name is designated by a blank space.

LAY, ROW, and COL—are integer variables equal to the layer, row, and column number of the model cell in which the head-type or drain-type state variable is located. For drain state variables, the LAY, ROW, and COL numbers must correspond to a valid drain location as specified in the MODFLOW DRN input file.

SVSP—is an integer variable that indicates the stress period during which the head, streamflow, or drain flow rate state variable is to be evaluated. To evaluate a head, streamflow, or drain flow rate state variable for multiple stress periods, define multiple state variables.

MFLG—is an integer variable that must be specified as 0. The variable is a flag that indicates to GWM that a head-type state variable is being defined for an MNW well.

WELLID—is a character variable that is the unique alphanumeric identification of an MNW-type well defined in the **DECVAR** file. The text string is limited to 20 alphanumeric characters. If the name of the well includes spaces, then enclose the name in quotation marks.

SEG and REACH—are integer variables equal to the segment and reach numbers of the model cell in which the streamflow-type state variable is located. The SEG and REACH numbers must correspond to a valid segment and reach as specified in either the STR or SFR input files.

FL_OR_LK – an integer variable indicating the type of the streamflow state variable. A value of zero indicates that the state variable is a flow type that will hold the value of streamflow leaving the reach. Any other value indicates that the state variable is a leakage type that will hold the value of leakage between the aquifer and stream in the reach. Both the flow and leakage types have units of cubic length per time. This variable is optional. If no value is listed, the variable will be assumed to be flow type (that is, FL_OR_LK = 0).

SPSTRT and SPEND—are integer variables equal to the stress periods at which the evaluation of the storage or drain volume state variables will start and end. For storage state variables, the change in storage associated with the state variable will be computed by subtracting the volume of water in a specified portion of the model domain at the beginning of stress period SPSTRT from the volume of water in the same portion at the end of stress period SPEND. For drain volume state variables, the volume of discharge to the drain state variable will be computed for the time extending from the beginning of the specified stress period SPSTRT to the end of the specified stress period SPEND.

CZONE—is a character variable that describes the portion of the aquifer domain to be included in the storage state variable. Two options are allowed:

ALL—the storage state variable will record the change in water stored in the entire model domain.

ZONE—the storage state variable will record the change in water stored in a portion of the model domain that is defined in subsequent records in the file.

NSVZL—is an integer variable equal to the number of model layers included in the storage state variable zone. The zone array will be read one layer at a time.

LNUM—is an integer variable equal to the layer number for the storage state variable zone array.

SVZONE—is a two-dimensional (one layer) zone array that is read using the U2DINT utility subroutine of MODFLOW (see Harbaugh, 2005, p. 8-57 through 8-59). A value will be read for each cell in the model layer. A value of zero indicates the cell is not included in the storage state variable; a value greater than zero indicates the cell will be included in the definition of the storage state variable.

NMDCEL—is an integer variable equal to the number of managed drain cells associated with the state variable.

DRNTYP—is an integer variable that takes a value of 1 or 2:

If DRNTYP = 1, the state variable represents a drain flow rate at the end of stress period SVSP (the values of SPSTRT and SPEND are ignored).

If DRNTYP = 2, the state variable represents a volume of drain flow between a starting time (SPSTRT) and ending time (SPEND). The value of SVSP is ignored (but a value must be specified).

AUX-value—is an optional positive-integer variable that serves as an index to identify the drain on the cell. When multiple drains are present on a single cell, GWM-2005 determines the correct drain by matching the auxiliary-variable values defined by AUX-value with the auxiliary variables defined by auxiliary-variable GWM-DR in the Drain Package input file.

Objective Function (OBJFNC) File

This file is used to define the objective function that is to be maximized or minimized and the coefficients for each decision variable in the objective function. When the LGR capability is used, the **OBJFNC** file must be defined in the parent-model **GWM** file. The GWM run will terminate if an **OBJFNC** file is referenced in a child-model **GWM** file. The **OBJFNC** file may reference decision variables defined for any model. Note that it is not necessary to include all decision variables in the objective function.

The **OBJFNC** file includes seven input items:

0. [#Text]

Item 0 is optional—# must appear in column 1. Item 0 can be repeated multiple times.

1. IPRN
2. OBJTYPE FNTYP
3. NFVOBJ NEVOBJ NBVOBJ [NSVOBJ]
4. The following record is repeated for each of the NFVOBJ flow-rate decision variables:
   FVNAME FVOBJC
5. The following record is repeated for each of the NEVOBJ external decision variables:
   EVNAME EVOBJC
6. The following record is repeated for each of the NBVOBJ binary decision variables:
   BVNAME BVOBJC
7. The following record is repeated for each of the NSVOBJ state variables:
   SVNAME SVOBJC

The variables are defined as follows:

Text—is a character variable up to 199 characters long that starts in column 2. Any characters can be included in Text. Lines beginning with # are restricted to these first lines of the file. Text is printed when the file is read.

IPRN— is an integer variable that describes the amount of output that is written to the **GWM OUT** file. IPRN must be specified as either 0 or 1. When IPRN equals 0, a minimum amount of information about the objective function is written to the GWM output file; when IPRN equals 1, detailed information about the objective function is written to the GWM output file.

OBJTYP—is a character variable used to define whether the objective is to maximize or minimize the objective function. OBJTYP must be defined as either MIN (for minimize) or MAX (for maximize).

FNTYP—is a character variable used to define the type of objective function. Three options are allowed:

> WSDV—the objective function takes the form of a weighted sum of decision variables (eq. 2a in Ahlfeld and others, 2011). Weighting is automatically applied by multiplying flow-rate, external, and state variables by the length of time the variable is active. The length of time is determined by summing the length of all stress periods during which the decision variable is active, as defined in the associated definition of the variable. This form of the objective function is commonly used to convert variables that have units of flow rate to variables with units of volume. Additional weighting is applied to the variables by the user-specified values of the cost/benefit coefficients (FVOBJC, EVOBJC, BVOBJC, and SVOBJC).

USDV—the objective function takes the form of an unweighted sum of decision variables (eq. 2b in Ahlfeld and others, 2011). The variables are not weighted by time. Additional weighting is applied to the variables by the user-specified values of the cost/benefit coefficients (FVOBJC, EVOBJC, BVOBJC, and SVOBJC).

MSDV—the objective function takes the form of a sum of decision variables with mixed weighting (eq. 2c in Ahlfeld and others, 2011). Flow-rate decision variables are weighted by the duration of their activity, but external and state variables are not weighted by their duration of activity. Additional weighting is applied to the variables by the user-specified values of the cost/benefit coefficients (FVOBJC, EVOBJC, BVOBJC, and SVOBJC).

NFVOBJ—is an integer variable equal to the number of flow-rate decision variables in the objective function and must have a value less than or equal to NFVAR specified in the decision-variables file.

NEVOBJ—is an integer variable equal to the number of external decision variables in the objective function and must have a value less than or equal to NEVAR specified in the decision-variables file.

NBVOBJ—is an integer variable equal to the number of binary decision variables in the objective function and must have a value less than or equal to NBVAR specified in the decision-variables file.

NSVOBJ—is an optional integer variable equal to the number of state variables in the objective function. NSVOBJ must have a value that is less than or equal to the sum of NHVAR, NRVAR, NSVAR, and NDVAR specified in the **STAVAR** file.

FVNAME—is a character variable up to 10 characters long that is one of the flow-rate decision-variable names defined for any parent or child model. Each of the FVNAME variables listed must be defined in a parent or child **DECVAR** file. A flow-rate decision-variable name can only be listed once in the **OBJFNC** file.

FVOBJC—is a real variable that is a coefficient associated with each flow-rate decision variable FVNAME. For example, FVOBJC could represent the cost per unit volume of water withdrawn or injected at the management site.

EVNAME— is a character variable up to 10 characters long that is one of the external decision-variable names defined for any parent or child model. Each of the EVNAME variables listed must be defined in a parent or child **DECVAR** file. An external decision-variable name can be listed only once in the OBJFNC file.

EVOBJC—is a real variable that is a coefficient associated with each external decision variable EVNAME. For example, EVOBJC could represent the cost per unit volume of water associated with the external variable.

BVNAME— is a character variable up to 10 characters long that is one of the binary decision-variable names defined for any parent or child model. Each of the BVNAME variables listed must be defined in a parent or child **DECVAR** file. A binary-variable name can be listed only once in the **OBJFNC** file.

BVOBJC—is a real variable that is a coefficient associated with each binary variable BVNAME. For example, BVOBJC could represent the cost for installation of the management site. Most often, the coefficients will be positive when OBJTYP is MIN and negative when OBJTYP is MAX. This will ensure that the binary variables are active only when their associated flow-rate and external decision variables are active.

SVNAME— is a character variable up to 10 characters long that is one of the state-variable names defined in the **STAVAR** file. A sttate-variable name can be listed only once in the **OBJFNC** file.

SVOBJC—is a real variable that is a coefficient associated with each state variable SVNAME.

Decision-Variable Constraints (VARCON) File

The decision-variable constraints file is used to define lower and upper bounds for the flow-rate and external decision variables and the reference flow rates to be used in the first groundwater-flow run by GWM. Records must be specified for all NFVAR and NEVAR decision variables defined in the corresponding **DECVAR** file for a model. Reference flow rates (input variable FVREF) must be specified for each flow-rate decision variable when either drawdown constraints or streamflow-depletion constraints are used in a GWM run (see discussions on pages 15 and 31 in Ahlfeld and others, 2005); reference flow rates also may be used to calculate base conditions for the calculation of the response matrix (see discussion of variable IBASE in the **SOLN** file, and on pages 31-32 in Ahlfeld and others, 2005).

The **VARCON** file includes three input items:

0. [#Text]

Item 0 is optional—# must appear in column 1. Item 0 can be repeated multiple times.

1. IPRN
2. The following record is read for each of the NFVAR decision variables:
   FVNAME FVMIN FVMAX FVREF
3. The following record is read for each of the NEVAR decision variables:
   EVNAME EVMIN EVMAX

The variables are defined as follows:

Text—is a character variable up to 199 characters long that starts in column 2. Any characters can be included in Text. Lines beginning with # are restricted to these first lines of the file. Text is printed when the file is read.

IPRN— is an integer variable that describes the amount of output that is written to the **GWM OUT** file. IPRN must be specified as either 0 or 1. When IPRN equals 0, a minimum amount of information about the decision-variable constraints is written to the GWM output file; when IPRN equals 1, detailed information about the decision-variable constraints is written to the GWM output file.

FVNAME—is a character variable up to 10 characters long that is one of the flow-rate decision-variable names defined for any parent or child model. Each of the FVNAME variables listed must be defined ina parent or child **DECVAR** file. A flow-rate decision-variable name can only be listed once in the **VARCON** file.

FVMIN and FVMAX—are real variables that are equal to the minimum (FVMIN) and maximum (FVMAX) flow rates allowed for the decision variable. Values greater than or equal to 0 must be specified for FVMIN and FVMAX; the specification of FTYPE given in the **DECVAR** file indicates whether the pumping rates are withdrawal or injection rates. FVMIN must be less than or equal to FVMAX. Note that a nonzero value of FVMIN implies that the decision variable has been associated with a binary variable in the **DECVAR** file. If the decision variable is not associated with a binary variable, then the nonzero value of FVMIN is ignored by GWM and FVMIN is set to zero. The user can specify a nonzero lower bound for a flow-rate decision variable not associated with a binary variable by use of a linear-summation constraint (see description of **SUMCON** file).

FVREF—is a real variable equal to the flow rate for the decision variable that is used by GWM to calculate the reference values of the state variables. These include heads at drawdown-constraint locations if drawdown constraints are used and reference streamflows at streamflow-constraint locations. FVREF also may be used to calculate base conditions for the calculation of the response matrix. If no value is entered for FVREF, it is assigned a value of 0.

EVNAME— is a character variable up to 10 characters long that is one of the external decision-variable names defined for any parent or child model. Each of the EVNAME variables listed must be defined in a parent or child **DECVAR** file. An external decision-variable name can only be listed once in the **VARCON** file.

EVMIN and EVMAX—are real variables that are equal to the minimum (EVMIN) and maximum (EVMAX) values allowed for the external decision variable. Because external variables are defined as positive-valued variables, values greater than or equal to 0 must be specified for EVMIN and EVMAX. EVMIN must be less than or equal to EVMAX. Note that a nonzero value of EVMIN implies that the decision variable has been associated with a binary variable in the **DECVAR** file. If the decision variable is not associated with a binary variable, then the nonzero value of EVMIN is ignored by GWM and EVMIN is set to zero. The user can specify a nonzero lower bound for an external decision variable not associated with a binary variable by use of a summation constraint (see description of **SUMCON** file).

Linear-Summation Constraints (SUMCON) File

The linear-summation constraints file is used to define linear relations among decision variables. If the LGR capability is active and linear-summation constraints are used, then the **SUMCO**N file must be listed in the parent-model GWM file. The GWM run will terminate if a **SUMCON** file is referenced in a child-model GWM file. The **SUMCON** file may include constraints that reference decision variables defined for any model.

The **SUMCON** file includes three input items:

0. [#Text]

Item 0 is optional—# must appear in column 1. Item 0 can be repeated multiple times.

1. IPRN
2. SMCNUM

3a. Records 3a and 3b are read for each of the SMCNUM constraints:
SMCNAME NTERMS TYPE RHS
3b. The following record is repeated once for each of NTERMS terms specified in record 3a:
GVNAME GVCOEFF

The variables are defined as follows:

Text—is a character variable up to 199 characters long that starts in column 2. Any characters can be included in Text. Lines beginning with # are restricted to these first lines of the file. Text is printed when the file is read.

IPRN— is an integer variable that describes the amount of output that is written to the **GWM OUT** file. IPRN must be specified as either 0 or 1. When IPRN equals 0, a minimum amount of information about the summation constraints is written to the GWM output file; when IPRN equals 1, detailed information about the summation constraints is written to the GWM output file.

SMCNUM—is an integer variable equal to the number of summation constraints defined in the file.

SMCNAME—is a character variable up to 10 characters long that is a unique name designated for the constraint. No spaces are allowed in the name. The end of the name is designated by a blank space.

NTERMS—is an integer variable equal to the number of terms on the left-hand side of the constraint. All of the terms are combined to form the left-hand side of the constraint.

TYPE—is a character variable used to specify the type of constraint. Three options are allowed:

LE indicates that the left-hand side of the constraint is less than or equal to the right-hand side of the constraint,

GE indicates that the left-hand side of the constraint is greater than or equal to the right-hand side of the constraint, and

EQ indicates that the left-hand and right-hand sides of the constraint are equal.

RHS—is a real variable equal to the value of the right-hand side of the constraint.

GVNAME—is a character variable up to 10 characters long that is one of the decision-variable names defined in a **DECVAR** file or one of the state-variable names defined in a **STAVAR** file. Any combination of flow-rate (FVNAME), external (EVNAME), binary (BVNAME), or state (SVNAME) variables may be defined in a constraint. The user must ensure that the variables included are logically consistent.

GVCOEFF—is a real variable equal to the value of the coefficient associated with variable GVNAME. The user must ensure that a consistent set of units is used for all GVCOEFF and RHS terms.

Head Constraints (HEDCON) File

The head-constraints file is used to define head constraints at model cells. These include upper and lower bounds on heads, drawdowns, head differences between two cells, and gradients between two cells. Upper- and lower-bound head constraints also can be defined at managed MNW-type wells, but heads in MNW wells cannot be used in drawdown, head-difference, or gradient constraints. The MNW wells must be defined as flow-rate decision variables in the **DECVAR** file for the management formulation. Head-bound constraints on heads at MNW-type wells are included with other head-bound constraints under the NHB input variable in item 2 below. Item 3 includes two options: head-bound constraints specified at model cells are defined in item 3a, whereas head-bound constraints specified at MNW wells are defined in item 3b. The variable MFLG in item 3b is a flag that must be set to zero and indicates to GWM that the line of input data is formatted as item 3b (head-bound constraint at an MNW well). The variable WELLID is the name of an MNW well that has been defined in the **DECVAR** file for the management formulation; GWM uses this name to find the location in the flow-process model where the constraint will be imposed.

If the LGR capability is being used, head constraints must be on the grid with which the **HEDCON** file is associated; head difference and gradient constraints between two cells may not cross the interface between grids.

The **HEDCON** file includes six input items:

0. [#Text]

Item 0 is optional—# must appear in column 1. Item 0 can be repeated multiple times.

1. IPRN
2. NHB NDD NDF NGD

3a. The following record is read for each head constraint calculated at a model cell:

HBNAME LAYH ROWH COLH TYPH BND NSP

3b. The following record is read for each head constraint calculated at an MNW-type well that has been defined in the **DECVAR** file:

HBNAME MFLG WELLID TYPH BND NSP

4. The following record is read for each of the NDD drawdown constraints:

DDNAME LAYD ROWD COLD TYPD BND NSP

5. The following record is read for each of the NDF head-difference constraints:

HDIFNAME LAY1 ROW1 COL1 LAY2 ROW2 COL2 HD NSP

6. The following record is read for each of the NGD gradient constraints:

GRADNAME LAY1 ROW1 COL1 LAY2 ROW2 COL2 LEN GRAD NSP

The variables are defined as follows:

Text—is a character variable up to 199 characters long that starts in column 2. Any characters can be included in Text. Lines beginning with # are restricted to these first lines of the file. Text is printed when the file is read.

IPRN— is an integer variable that describes the amount of output that is written to the **GWM OUT** file. IPRN must be specified as either 0 or 1. When IPRN equals 0, a minimum amount of information about the head constraints is written to the GWM output file; when IPRN equals 1, detailed information about the head constraints is written to the GWM output file.

NHB—is an integer variable equal to the total number of head-bound constraints defined in items 3a and 3b that need to be satisfied in the management model.

NDD—is an integer variable equal to the number of drawdown constraints that need to be satisfied in the management model.

NDF—is an integer variable equal to the number of head-difference constraints that need to be satisfied in the management model.

NGD—is an integer variable equal to the number of gradient constraints that need to be satisfied in the management model.

Head-bound constraints:

HBNAME—is a character variable up to 10 characters long that is a unique name designated for the head-bound constraint. No spaces are allowed in the name. The end of the name is designated by a blank space.

LAYH, ROWH, and COLH—are integer variables equal to the layer, row, and column number of the model cell in which the head-bound constraint is located.

TYPH—is a character variable used to specify the type of head bound. Two options are allowed:

LE indicates that head calculated by the model must be less than or equal to the value specified by BND, and

GE indicates that head calculated by the model must be greater than or equal to the value specified by BND.

BND—is a real variable equal to the specified upper or lower bound on head at the model cell or multi-node well at the end of the stress period.

NSP—is an integer variable that indicates the stress period during which the constraint is imposed. If the constraint is imposed over multiple stress periods, then a separate record must be provided for each stress period.

MFLG—is an integer variable that must be specified as 0. The variable is a flag that indicates to GWM that a head-bound constraint is being defined at an MNW well.

WELLID—is a character variable that is the unique alphanumeric identification of an MNW-type well defined in the **DECVAR** file. The text string is limited to 20 alphanumeric characters. If the name of the well includes spaces, then enclose the name in quotation marks.

Drawdown constraints:

DDNAME—is a character variable up to 10 characters long that is a unique name designated for the drawdown constraint. No spaces are allowed in the name. The end of the name is designated by a blank space.

LAYD, ROWD, COLD—are integer variables equal to the layer, row, and column number of the model cell in which the drawdown constraint is located.

TYPD—is a character variable used to specify the type of drawdown bound. Two options are allowed:

LE indicates that drawdown calculated by the model must be less than or equal to the value specified by BND, and

GE indicates that drawdown calculated by the model must be greater than or equal to the value specified by BND.

BND—is a real variable equal to the specified upper or lower bound on drawdown at the model cell at the end of the stress period (see fig. 2B in Ahlfeld and others, 2005).

NSP—was defined for record 3.

Head-difference constraints:

HDIFNAME—is a character variable up to 10 characters long that is a unique name designated for the head-difference constraint. No spaces are allowed in the name. The end of the name is designated by a blank space.

LAY1, ROW1, and COL1—are integer variables equal to the layer, row, and column number of the model cell corresponding to the first location, $(i, j, k)_1$, in the head-difference constraint (see fig. 2C in Ahlfeld and others, 2005).

LAY2, ROW2, and COL2—are integer variables equal to the layer, row, and column number of the model cell corresponding to the second location, $(i, j, k)_2$, in the head-difference constraint (see fig. 2C in Ahlfeld and others, 2005). GWM requires that the head at the second location be lower than the head at the first location by an amount of at least HD. The constraint will impose the requirement that head at the second location be lower than the head at the first location by an amount of at least HD.

HD—is a real variable equal to the specified difference in heads between the first and second model cells at the end of the stress period (see fig. 2C in Ahlfeld and others, 2005).

NSP—was defined for record 3.

Gradient constraints:

GRADNAME—is a character variable up to 10 characters long that is a unique name designated for the gradient constraint. No spaces are allowed in the name. The end of the name is designated by a blank space.

LAY1, ROW1, and COL1—are integer variables equal to the layer, row, and column number of the model cell corresponding to the first location, $(i, j, k)_1$, in the gradient constraint (see fig. 2D in Ahlfeld and others, 2005).

LAY2, ROW2, and COL2—are integer variables equal to the layer, row, and column number of the model cell corresponding to the second location, $(i, j, k)_2$, in the gradient constraint (see fig. 2D in Ahlfeld and others, 2005). GWM requires that the head at the second location be lower than the head at the first location. The constraint will impose the requirement that head at the second location be lower than the head at the first location.

LEN—is a real variable equal to the distance between the first and second model cells (Δx shown in fig. 2D in Ahlfeld and others, 2005).

GRAD—is a real variable equal to the minimum gradient between the first and the second model cells at the end of the stress period (see fig. 2D in Ahlfeld and others, 2005).

NSP—was defined for record 3.

Streamflow Constraints (STRMCON) File

The streamflow constraints (**STRMCON**) file is used to define streamflow, streamflow-depletion, and stream-leakage constraints. Either the STR (Prudic, 1989) or SFR (Prudic and others, 2004; Niswonger and Prudic, 2005) Streamflow-Routing Packages may be used to simulate streamflow, but both packages cannot be used on a grid simultaneously. The file types specified in the MODFLOW **NAME** file indicate which package is being used to simulate streamflow; if GWM detects that both STR and SFR have been made active on a grid in the **NAME** file, the code will stop, and an error message will be printed to the GWM **OUT** file.

The input structure of the **STRMCON** file is the same regardless of which streamflow-routing package is being used. When the LGR capability is used, the constraint locations must reference locations on the grid associated with the **STRMCON** file because streams cannot cross a grid interface.

The **STRMCON** file includes four input items:

0. [#Text]

Item 0 is optional—# must appear in column 1. Item 0 can be repeated multiple times.

1. IPRN
2. NSF NSD NLK
3. The following record is read for each of the NSF streamflow constraints:
   SFNAME SEG REACH TYPSF BND NSP
4. The following record is read for each of the NSD streamflow-depletion constraints:
   SDNAME SEG REACH TYPSD BND NSP
5. The following record is read for each of the NLK stream-leakage constraints:
   SLNAME SEG REACH TYPSL BND NSP

The variables are defined as follows:

Text—is a character variable up to 199 characters long that starts in column 2. Any characters can be included in Text. Lines beginning with # are restricted to these first lines of the file. Text is printed when the file is read.

IPRN— is an integer variable that describes the amount of output that is written to the **GWM OUT** file. IPRN must be specified as either 0 or 1. When IPRN equals 0, a minimum amount of information about the streamflow constraints is written to the GWM output file; when IPRN equals 1, detailed information about the streamflow constraints is written to the GWM output file.

NSF—is an integer variable equal to the number of streamflow constraints that need to be satisfied in the management model.

NSD—is an integer variable equal to the number of streamflow-depletion constraints that need to be satisfied in the management model.

NLK—is an integer variable equal to the number of stream-leakage constraints that need to be satisfied in the management model.

Streamflow constraints:

SFNAME—is a character variable up to 10 characters long that is a unique name designated for the streamflow constraint. No spaces are allowed in the name. The end of the name is designated by a blank space.

SEG and REACH—are integer variables equal to the segment and reach number, as specified in the STR or SFR Packages, of the model cell in which the streamflow constraint is located.

TYPSF—is a character variable used to specify the type of streamflow constraint. Two options are allowed:

LE indicates that streamflow at the stream site must be less than or equal to the value specified by BND, and

GE indicates that streamflow at the stream site must be greater than or equal to the value specified by BND.

BND—is a real variable equal to the specified amount of streamflow allowed at the stream site at the end of the stress period.

NSP—is an integer variable that indicates the stress period during which the constraint is imposed. If the constraint is imposed over multiple stress periods, then a separate record must be provided for each stress period.

Streamflow-depletion constraints:

SDNAME—is a character variable up to 10 characters long that is a unique name designated for the streamflow-depletion constraint. No spaces are allowed in the name. The end of the name is designated by a blank space.

SEG and REACH—are integer variables equal to the segment and reach number of the model cell in which the streamflow-depletion constraint is located.

TYPSD—is a character variable used to specify the type of streamflow-depletion constraint. Two options are allowed:

LE indicates that streamflow depletion at the stream site must be less than or equal to the value specified by BND, and

GE indicates that streamflow depletion at the stream site must be greater than or equal to the value specified by BND.

BND—is a real variable equal to the specified amount of streamflow depletion allowed at the stream site at the end of the stress period.

NSP—was defined for record 3.

Stream-leakage constraints:

SLNAME—is a character variable up to 10 characters long that is a unique name designated for the stream-leakage constraint. No spaces are allowed in the name. The end of the name is designated by a blank space.

SEG and REACH—are integer variables equal to the segment and reach number, as specified in the STR or SFR Packages, of the model cell in which the stream-leakage constraint is located.

TYPSF—is a character variable used to specify the type of stream-leakage constraint. Two options are allowed:

LE indicates that stream leakage at the stream site must be less than or equal to the value specified by BND, and

GE indicates that stream leakage at the stream site must be greater than or equal to the value specified by BND.

BND—is a real variable equal to the specified amount of leakage allowed at the stream site at the end of the stress period. Leakage rates have units of cubic length per time. BND can be specified as either positive or negative. Note that stream leakage in a reach is positive when flow is from the stream to the aquifer and negative when the flow is from the aquifer to the stream. For example, a leakage constraint specified as TYPSF = LE and BND = -10 cubic feet per second indicates that the leakage (or groundwater-discharge) rate from the aquifer to the stream in the reach must be greater than or equal to 10 cubic feet per second.

NSP—is an integer variable that indicates the stress period during which the constraint is imposed. If the constraint is imposed over multiple stress periods, then a separate record must be provided for each stress period.

Solution and Output-Control Parameters (SOLN) File

The solution and output-control parameters file is used to define several variables that control the solution algorithm for the optimization problem and the type and amount of output that is printed to the output files. Only one **SOLN** file is defined for any GWM problem and, when the LGR capability is used, the **SOLN** file must be defined in the parent-model **GWM** file. GWM will ignore a **SOLN** file that is referenced in a child-model GWM file.

The **SOLN** file includes the following input items:

0. [#Text]

Item 0 is optional—# must appear in column 1. Item 0 can be repeated multiple times.

1. SOLNTYP

If SOLNTYP is NS, then these records are read:

2a. DELTA

2b. NSIGDIG NPGNMX PGFACT CRITMFC
2c. RMNAME IRM
Skip to record 6a.

If SOLNTYP is MPS, then these records are read:
3a. DELTA
3b. NSIGDIG NPGNMX PGFACT CRITMFC
3c. MPSNAME
Skip to record 6a.

If SOLNTYP is LP, then these records are read:
4a. IRM
4b. LPITMAX BBITMAX
4c. DELTA
4d. NSIGDIG NPGNMX PGFACT CRITMFC
4e. BBITPRT RANGE
The following record is read if IRM equals 0, 1, or 3:
4f. RMNAME1
The following record is read if IRM equals 4 or 5:
4g. RMNAME1 RMNAME2
Skip to record 6a.

If SOLNTYP is SLP, then these records are read:
5a. SLPITMAX LPITMAX BBITMAX
5b. SLPVCRIT SLPZCRIT DINIT DMIN DSC
5c. NSIGDIG NPGNMX PGFACT AFACT NINFMX CRITMFC
5d. SLPITPRT BBITPRT RANGE

If SOLNTYP is FR, then proceed to record 6a.

6a. IBASE
If IBASE equals 1, the following record is read for each of the NFVAR decision variables:
6b. FVNAME FVBASE

The variables are defined as follows:
Text—is a character variable up to 199 characters long that starts in column 2. Any characters can be included in Text. Lines beginning with # are restricted to these first lines of the file. Text is printed when the file is read.

SOLNTYP—is a character variable specified as either NS, MPS, LP, SLP, or FR:

NS indicates that no solution to the management formulation will be found by GWM. GWM will calculate the response matrix and write it to the response-matrix file specified by RMNAME in record 2c; GWM will then stop.

MPS indicates that no solution to the management formulation will be found by GWM. GWM will write the management formulation in MPS (Mathematical Programming System) format to the file specified by MPSNAME in record 3c; GWM will then stop.

LP (with or without binary variables) indicates that the optimization formulation is to be solved by using linear programming, or, for problems with binary variables, linear programming and the branch and bound method. A SOLNTYP equal to LP is normally used if the flow model contains linear features only (see the section "Solution of Ground-Water Management Problems with

GWM" in Ahlfeld and others, 2005, for a discussion of linear and nonlinear features). Alternatively, the use of LP allows the user to force GWM to apply a linear solution even if nonlinear features are present. In that case, the management formulation will be solved with a single response matrix. This option should be used carefully and may lead to inaccurate results if the management problem has a significant nonlinear response.

SLP (with or without binary variables) indicates that the optimization formulation is to be solved by sequential (iterative) linear programming or, for problems with binary variables, sequential linear programming and the branch and bound method.

FR indicates that a forward run of GWM will be done with user-specified values of the flow-rate decision variables. Such a run can be useful for evaluating how small changes in the values of the flow-rate decision variables affect the value of the objective function and simulation-based constraints. When this option is selected, the user must specify a value of IBASE in record 6a. IBASE can be set to a value of 0, in which case the flow-rate decision variables are specified with variable FVREF in the VARCON file, or a value of 1, in which case they are specified with variable FVBASE in record 6b. GWM will use the specified flow rates in a forward run of the MODFLOW Groundwater Flow (GWF) Process. As with all other SOLNTYP options, if IBASE is set to a value of 1 and either drawdown or stream depletion constraints are included in the formulation, then both a reference and a base run for the GWF Process will be conducted. The value of the objective function that is reported for the run will include only the flow-rate decision variable and state-variable portions of the objective function; the external-variable and binary-variable portions of the objective function will be set to zero for the computation. GWM will also write the simulated value of each state variable and the status of the simulation-based constraints in a format that is equivalent to the output of a base-condition flow simulation. Any external variables or binary variables that appear in constraints will be assigned a value of zero. GWM will then stop, without calculating a response matrix or solving an optimization formulation. (An example use of the FR option is provide for the DEWATER sample problem.)

SOLNTYP is NS:

DELTA—is a real variable equal to the perturbation parameter δ^0 (see eq. 63 in Ahlfeld and others, 2005) used to determine the response matrix. DELTA is multiplied by FVMAX for each flow-rate decision variable to determine each perturbation value. A positive value of DELTA implies a forward-difference calculation of the response coefficient (that is, an increase in flow rate), whereas a negative value implies a backward-difference calculation (that is, a decrease in flow rate). If the flow model has a linear response to pumping, then the value assigned to DELTA has negligible significance and values between 0.1 and 1.0 are commonly used. If the flow model has nonlinear responses, then smaller values of DELTA are preferred, commonly 0.001 to 0.1. See "Solution of Ground-Water Management Problems with GWM" in Ahlfeld and others (2005) for further discussion of DELTA.

NSIGDIG—is an integer variable equal to a lower limit on the number of significant digits in response-matrix entries. For each entry in a column of the response matrix, the ratio of the difference in observed state to the HCLOSE variable is computed. If the largest ratio in the column has fewer than NSIGDIG significant digits, the perturbation is considered to have failed. Values of NSIGDIG of 1 to 3 are often adequate for small problems. Higher values should be used as problems become more complex. See "Description of Selected Conventions, Options, and Variables in GWM" in Ahlfeld and others (2005, p. 32-33) for further discussion of NSIGDIG.

NPGNMX—is an integer variable that controls the automatic reexecution of the flow process when it fails. When failure is detected, automatic resetting of flow-rate decision-variable values may produce a successful solution. Failure may occur during either base or perturbation flow-process runs. For perturbation runs, the resetting is controlled by PGFACT, whereas for base runs, the resetting is controlled by AFACT (see record 5c). The value of NPGNMX is the maximum number of attempts to achieve a successful flow-process run. When NPGNMX is set to 0, testing of the flow-process results is eliminated except to confirm that the flow process has converged. If NPGNMX is set to 0 and GWM determines that the GWF Process has failed to converge, then the GWM run will be terminated.

PGFACT—is a real variable equal to the perturbation step-length adjustment factor used during perturbation failure. PGFACT must be greater than 0 and less than 1. The smaller the value of PGFACT, the larger the adjustment to the perturbation value. A value of 0.5 is suggested. See "Description of Selected Conventions, Options, and Variables in GWM" in Ahlfeld and others (2005, p. 33) for further discussion of PGFACT.

CRITMFC—is a real variable used to adjust the criteria for accepting convergence of a GWF Process run. For a single-grid problem, the options are:

CRITMFC = 0.0: convergence of the GWF Process will be assumed only if the applicable GWF convergence criteria (HCLOSE and(or) RCLOSE) have been met at the end of each time step (this is the standard approach for the GWF Process of MODFLOW);

CRITMFC > 0.0: the results of the GWF Process will be accepted if the applicable GWF convergence criteria (HCLOSE and(or) RCLOSE) have not been met, but the percent discrepancy between the inflow and outflow rates for each time step as reported in the GWF Process volumetric budget is less than or equal to the value specified for CRITMFC. For example, if CRITMFC is specified as 0.1, then GWF Process results will be accepted if the percent discrepancy between the inflow and outflow rates of the volumetric budget at the end of each time step is less than or equal to 0.1 percent;

CRITMFC < 0.0: the results of the GWF Process will be accepted regardless of the GWF convergence criteria (HCLOSE and(or) RCLOSE) or the percent discrepancy.

For multiple grids, the options are:

CRITMFC = 0.0: convergence of the GWF Process will be assumed only if the HCLOSELGR and FCLOSELGR criteria have been met at the end of each time step (this is the standard approach for an LGR simulation);

CRITMFC > 0.0: the results of the GWF Process will be accepted if the HCLOSELGR or FCLOSELGR criteria have not been met, but the percent discrepancy between the inflow and outflow rates for each time step as reported in the GWF Process volumetric budget for all grids is less than or equal to the value specified for CRITMFC. For example, if CRITMFC is specified as 0.1, then GWF Process results will be accepted if the percent discrepancy between the inflow and outflow rates of the volumetric budget for all grids at the end of each time step is less than or equal to 0.1 percent;

CRITMFC < 0.0: the results of the GWF Process will be accepted regardless of the HCLOSELGR or FCLOSELGR criteria or the percent discrepancy.

RMNAME—is the filename (or pathname) to which the response matrix will be written.

IRM—is an integer variable. Its value specifies the format in which the response matrix will be written to file RMNAME (if no value is specified for IRM, a default value of 1 is assumed):

IRM = 1: the response matrix is saved to a non-formatted file;

IRM = 3: the response matrix is printed to a formatted file.

SOLNTYP is MPS:

DELTA—was defined for item 2a.

NSIGDIG, NPGNMX, PGFACT, CRITMFC—were defined for item 2b.

MPSNAME—is the filename (or pathname) to which the formulation will be written in MPS format.

SOLNTYP is LP:

IRM—is an integer variable. Its value specifies whether or not the response matrix will be calculated or read from an input file, and whether or not the response matrix will be saved:

IRM = 0: An existing nonformatted file containing the response matrix will be read from the file specified by RMNAME1 in item 4f, and the LP solved.

IRM = 1: The response matrix is computed and saved to a nonformatted file specified by RMNAME1 in item 4f. The LP is then solved.

IRM = 2: The response matrix is computed and the LP solved. The response matrix is not saved or printed.

IRM = 3: The response matrix is computed and printed to a formatted file specified by RMNAME1 in item 4f. The LP is then solved.

IRM = 4: The response matrix is computed, saved to a nonformatted file specified by RMNAME1 in item 4g, and printed to a formatted file specified by RMNAME2 in item 4g. The LP is then solved.

IRM = 5: An existing nonformatted file containing the response matrix is read from the file specified by RMNAME1 in item 4g, and the LP solved. The response matrix is printed to a formatted file specified by RMNAME2 in item 4g.

LPITMAX—is an integer variable. Its value is the maximum number of iterations allowed for the linear program solver; this limit prevents the solver from iterating indefinitely if it does not converge to a solution. If the linear solver is being used, and the value of LPITMAX is reached, the program will be terminated, and the output file will indicate that the maximum number of iterations has been reached. A typical value for LPITMAX is ten times the number of constraints.

BBITMAX—is an integer variable. BBITMAX is relevant only if the management problem includes one or more binary variables; otherwise, its value is ignored. BBITMAX is the maximum number of iterations allowed for the branch and bound program solver. Each iteration consists of one solution of the linear program. If the value of BBITMAX is reached, the program will be terminated, and the output file will indicate that the maximum number of iterations has been reached.

DELTA—was defined for item 2a.

NSIGDIG, NPGNMX, PGFACT, CRITMFC—were defined for item 2b.

BBITPRT—is an integer variable that specifies whether output describing the details of the branch and bound algorithm for solving mixed binary problems will be written to the GWM **OUT** file. A value of 1 indicates that this output will be written, and a value of 0 indicates that it will not. For problems with many binary variables, values of 1 can substantially increase the size of the GWM **OUT** file. If the management problem includes no binary variables, this value will be ignored.

RANGE—is an integer variable that indicates the status of the range analysis. A value of 1 indicates that range analysis is to be done and the results written to the GWM **OUT** file. A value of 0 indicates that no range analysis is to be done. Range analysis is based on the assumption that the optimization problem is strictly linear with continuous variables. If binary variables or nonlinear responses are significant in the problem, then the range analysis may be inaccurate.

RMNAME1—is a filename (including pathname, if required) from which the response matrix will be read if IRM equals 0 or 5 and to which the response matrix will be written if IRM equals 1, 3, or 4. RMNAME1 will be a formatted file if IRM equals 3 and a nonformatted file otherwise. The type of nonformatted file is defined in the openspec.inc file distributed with GWM.

RMNAME2— is a filename (including pathname, if required) to which a response matrix will be written if IRM equals 4 or 5.

SOLNTYP is SLP:

SLPITMAX—is an integer variable. Its value is the maximum number of iterations allowed for the SLP algorithm. If the value of SLPITMAX is reached, the program will be terminated, and the output file will indicate that the maximum number of iterations has been reached.

LPITMAX—was defined for item 4b.

BBITMAX—was defined for item 4b.

SLPVCRIT—is a real variable. Its value is the convergence criterion ε_1 (see eq. 69A in Ahlfeld and others, 2005), which is the first of two termination criteria when the SLP algorithm is used. This criterion is satisfied when the absolute value of the change in the values of all flow-rate decision variables from the previous iteration to the current iteration is less than a fraction ε_1 of the magnitude of the flow-rate decision variables at the current iteration. See Ahfled and others (2005, p. 25) for guidance on specifying the value of SLPVCRIT.

SLPZCRIT—is a real variable. Its value is the convergence criterion ε_2 (see eq. 69B in Ahlfeld and others, 2005), which is the second of two termination criteria when the SLP algorithm is used. This criterion is satisfied when the absolute value of the change in the value of the objective function is less than the specified fraction ε_2 of the value of the objective function.

DINIT, DMIN, and DSC—are real variables that control the value of the perturbation variable (see eq. 72 in Ahlfeld and others, 2005) used to compute response coefficients. DINIT is the perturbation variable used for the first iteration, DMIN is the minimum perturbation variable used, and DSC is a parameter that controls the rate of change of the perturbation variable. DINIT and DMIN must have the same sign. Positive values of DINIT and DMIN imply a forward-difference calculation of the response coefficient (that is, an increase in flow rate), whereas negative values imply a backward-difference calculation (that is, a decrease in flow rate). DSC must always be positive. See Ahlfeld and others (2005, p. 27 and 32) for discussions of DINIT, DMIN, and DSC.

NSIGDIG, NPGNMX, PGFACT, CRITMFC—were defined for item 2b.

AFACT—is a real variable equal to the relaxation parameter α (see eq. 73 in Ahlfeld and others, 2005) used to determine a temporary base solution when a base run fails. AFACT controls the interpolation between the current base solution and the most recent successful base solution. AFACT must be greater than 0 and less than 1. A value close to 0 implies that the temporary base solution will be close to the current base solution, whereas a value close to 1.0 implies that the temporary base solution will be close to the previous base solution. A value of 0.5 is suggested.

NINFMX—is an integer variable that specifies the maximum number of consecutive infeasible iterations that will be accepted by the SLP algorithm before the algorithm terminates.

SLPITPRT—is an integer variable that specifies whether output describing the details of the sequential-iteration algorithm will be written to the GWM **OUT** file. The options are

SLPITPRT = 0: No information on the progress of the SLP algorithm is written to the **OUT** file.

SLPITPRT = 1: Constraint and convergence status at each iteration of the SLP algorithm are written to the **OUT** file.

SLPITPRT = 2: The current optimal solution, constraint status, and convergence status at each iteration of the SLP algorithm are written to the **OUT** file. If GWMWFILE is active, then a new well file will be written based on the current optimal solution.

BBITPRT—was defined in item 4e.

RANGE—was defined in item 4e.

IBASE—is an integer variable equal to 0 or 1 that indicates the source for the values of the flow-rate decision variables that will be used as the base run. For problems solved using the SLP algorithm, these values are the starting point for the iterative algorithm. A value of IBASE equal to 0 indicates that the reference flow rates (FVREF) specified for each flow-rate decision variable in file VARCON will be used in the base run (and that record 6b is not necessary). A value of IBASE equal to 1 indicates that the flow rates specified for each decision variable by FVBASE in record 6b will be used to calculate the base run. See discussion about IBASE in the "Variables related to calculation of response coefficients" section of Ahlfeld and others (2005, p. 31-32).

FVNAME—is a character variable up to 10 characters long that is one of the flow-rate decision-variable names. Each name must be unique to the parent model or one of the child models (that is, the same name cannot be used in more than one model). A flow-rate decision-variable name can only be listed once in the **SOLN** file.

FVBASE—is a real variable equal to the rate for the flow-rate decision variable. These values are used by GWM to calculate the base run. If the SLP solution algorithm is used, these values are the starting point for the iterative algorithm.

References

Ahlfeld, D.P., and Barlow, P.M., 2013, Use of multi-node wells in the Groundwater-Management Process of MODFLOW-2005 (GWM-2005): U.S. Geological Survey Techniques and Methods 6-A47, 26 p.

Ahlfeld, D.P., Barlow, P.M., and Baker, K.M., 2011, Documentation for the State Variables Package for the Groundwater-Management Process of MODFLOW-2005 (GWM-2005): U.S. Geological Survey Techniques and Methods 6-A36, 45 p.

Ahlfeld, D.P., Baker, K.M., and Barlow, P.M., 2009, GWM-2005--A Groundwater-Management Process for MODFLOW-2005 with Local Grid Refinement (LGR) capability: U.S. Geological Survey Techniques and Methods 6-A33, 65 p.

Ahlfeld, D.P., Barlow, P.M., and Mulligan, A.E., 2005, GWM-A ground-water management process for the U.S. Geological Survey modular ground-water model (MODFLOW-2000): U.S. Geological Survey Open-File Report 2005-1072, 124 p.

Harbaugh, 2005, MODFLOW-2005, The U.S. Geological Survey modular ground-water model—the Ground-Water Flow Process: U.S. Geological Survey Techniques and Methods 6-A16, variously paginated.

Konikow, L.F., Hornberger, G.Z., Halford, K.J., and Hanson, R.T., 2009, Revised multi-node well (MNW2) package for MODFLOW ground-water flow model: U.S. Geological Survey Techniques and Methods, book 6, chap. 30, 67 p.

Mehl, S.W., and Hill, M.C, 2005, MODFLOW-2005, The U.S. Geological Survey modular ground-water model—Documentation of shared node local grid refinement (LGR) and the boundary flow and head (BFH) Package: U.S. Geological Survey Techniques and Methods 6–A16, 68 p.

Mehl, S.W., and Hill, M.C., 2007, MODFLOW-2005, The U.S. Geological Survey modular ground-water model—Documentation of the multiple-refined-areas capability of local grid refinement (LGR) and the boundary flow and head (BFH) package: U.S. Geological Survey Techniques and Methods 6–A21, 13 p.

Mehl, S.W., and Hill, M.C., 2010, MODFLOW-LGR: Modifications to the Streamflow-Routing Package (SFR2) to route streamflow through locally refined grids: U.S. Geological Survey Techniques and Methods 6-A34, 15 p.

Niswonger, R.G., and Prudic, D.E., 2005, Documentation of the Streamflow-Routing (SFR2) Package to include unsaturated flow beneath streams—A modification to SFR1: U.S. Geological Survey Techniques and Methods 6–A13, 48 p.

Prudic, D.E., 1989, Documentation of a computer program to simulate stream-aquifer relations using a modular, finite-difference, ground-water flow model: U.S. Geological Survey Open-File Report 88–729, 113 p.

Prudic, D.E., Konikow, L.F., and Banta, E.R., 2004, A new Streamflow-Routing (SFR1) Package to simulate stream-aquifer interaction with MODFLOW-2000: U.S. Geological Survey Open-File Report 2004–1042, 95 p.