CHAPTER 7 SOLVER PACKAGES

This chapter documents three solvers used in MODFLOW–2005. The focus is on how to use the solvers to produce an acceptable solution to the flow equations with a minimum of computational time. Details of how the solvers work are not included here. These details can be found in the references.

The best solver to use for a problem is usually problem dependent, which is why there are multiple solvers in MODFLOW. The fundamental criterion for solver selection is the ability to solve the equations for the simulation. All of the solvers can solve the equations of typical simulations, but some very large or highly nonlinear problems cannot be solved by all of the solvers. Assuming that a solver can solve the equations, the next most important objective for solver selection is minimizing the time to solve the equations. Solution time of course varies with the power of the computer, but for a given computer, different solvers can take substantially different amounts of time. Another consideration regarding solver selection is the amount of computer memory required. Each solver uses a different amount of memory. The amount of computer memory used, although not nearly the constraint that it once was, can still be of concern for very large simulations. Insufficient memory will of course prevent the solution of the equations. Minimizing the amount of memory used for the model when other programs are running concurrently can be advantageous.

All of the solvers documented here incorporate iteration. Iteration means repeated partial solution of the flow equation resulting in each solution being more accurate than the previous one. Iteration usually is more efficient than direct solution because a single iteration requires a small fraction of the work of direct solution. Doing multiple iterations is still less work than a direct solution. Further, the Direct solver incorporates the capability of iteration to allow the solution of nonlinear problems.

Iteration can occur at two levels. The primary level, called outer iteration, is implemented in the Formulate and Approximate Procedure shown in the overall flow chart for MODFLOW (fig. 3–1). The Formulate Procedure begins by constructing the flow equation. Any head-dependent (nonlinear) terms, such as transmissivity of water-table layers, will change as a result of the head computed in the prior outer iteration. The Approximate Procedure then performs one partial solution of the flow equations. Each invocation of the Formulate and Approximate Procedures is an outer iteration. The second iteration level is called inner iteration. Inner iteration is incorporated within the Approximate Procedure of some solvers. Inner iteration is used to improve accuracy without changing the nonlinear terms. So, the overall flowchart does not directly indicate the possibility of inner iteration. The outer iterations are repeated until a satisfactory solution is obtained, which is called solver convergence, or until a user-specified maximum number of iterations is reached.

Solver convergence cannot be directly determined by the solvers because a solver does not know what the actual solution is. Accordingly, indirect measures of convergence are used. One measure is called the head-change criterion. For every iteration, a solver compares the head from the prior iteration to the new head at all cells. The maximum absolute value of the head change for the iteration is compared to a user-specified value. If the maximum change is less than or equal to the user-specified value, the head-change convergence criterion has been met. The head-change criterion is implemented in all of the solvers.

Another convergence criterion, which is used by the Preconditioned Conjugate-Gradient solver, is called the residual criterion. For every iteration, the solver computes the difference between the inflows and outflows for each cell. The maximum absolute value of the difference between inflows and outflows is compared to a user-specified value. If the maximum change is less than or equal to the user-specified value, the residual criterion has been met. The difference between inflows and outflows is commonly called the residual, which is why this criterion is called the residual criterion.

7–2 MODFLOW–2005, The U.S. Geological Survey Modular Ground-Water Model

Strongly Implicit Procedure Package

The Strongly Implicit Procedure (SIP) is a method for solving a large system of simultaneous linear equations by iteration developed by Weinstein, Stone, and Kwan (1969). The implementation of SIP as a MODFLOW package is fully described in McDonald and Harbaugh (1988).

Iteration

The SIP Package uses only outer iteration. The SIP algorithm uses a user-specified number of iteration variables (described below) that are generated from an iteration seed. The iteration seed can be user specified; alternatively, the program will calculate a value. To avoid excessive computation time when the solution process is failing to converge, the user specifies the maximum number of outer iterations, MXITER. The simulation is stopped if the convergence criterion is not met within MXITER iterations. Convergence is determined using the head-change criterion described above.

User Guidance

The SIP iteration variables (sometimes called iteration parameters) mentioned above are used empirically. In practice, iteration variables are used in a repeated cycle. The user must specify the number of iteration variables, NPARM, which normally ranges from 4 to 10. The actual values of the iteration variables are computed from a single value called the seed, which is input variable WSEED. WSEED can be generated automatically from an empirical algorithm, or the user can specify WSEED, which is less than 1 and greater than 0.8.

Several points should be made regarding iteration variable selection. First, the process is essentially empirical and little is understood about why one sequence of variables performs better than another. Second, the variables chosen affect the rate of convergence but (assuming that convergence is achieved) should not influence the final solution. Third, the influence of the variables on the rate of convergence is extremely important.

The SIP solver provided in MODFLOW–2005 includes an additional variable that can influence convergence. This variable, which is designated ACCL in the program, functions as a multiplier of the head change computed by the SIP algorithm; thus, where no impact is desired, ACCL should be assigned a value of 1. Variables similar to ACCL have been used in various versions of SIP (Peaceman, 1977, p. 130) although Weinstein, Stone, and Kwan (1969) do not employ a variable of this type. ACCL is not cycled, but rather has a single user-assigned value. As a general rule, ACCL initially should be given a value of 1, and improvement in the rate of convergence should be pursued through adjustment of the seed term, as explained below. If problems with convergence persist, values of ACCL other than 1 can be tried.

Experience has shown that setting ACCL to 1 and using the seed value calculated by the program does not always produce optimum convergence—that is, the number of iterations required to achieve convergence is not minimum. Convergence rates will deviate from the optimum if the absolute value of head change in each iteration is consistently either too small or too large. When the head change is too large, the computed head overshoots the correct value, and oscillations occur as the head change repeatedly reverses to compensate for the overshoot. Severe overshoot causes divergence, while moderate overshoot can slow down convergence. When head change is too small, the opposite problem occurs; head tends to approach the correct value monotonically, but very slowly. In severe situations, the head changes at each iteration may be so small that the criterion for convergence is satisfied at all points, even though the computed heads are still far from the correct values. In such situations, a substantial volumetric budget imbalance will occur.

Weinstein, Stone, and Kwan (1969) suggest that a trial and error method can be used to improve the choice of WSEED. This can be done by making an initial run using the WSEED calculated by the program or chosen from experience, and using 1 for ACCL. A value of 5 for NPARM is usually acceptable. The trend of the maximum head change per iteration, with increasing iteration number, is observed for the iterations of a single time step. A table showing the cell at which the maximum absolute value of head change occurs for each iteration will be printed in the Listing File when variable IPRSIP is 1. The head change at these cells also is printed, including the sign. A positive sign means head increased. Normally, some variation in head change occurs from one iteration to the next caused by the cycling of iteration variables, but this variation is often superimposed on an overall trend in which head change

Chapter 7. Solver Packages 7–3

tends either to increase or decrease as iterations continue; this overall trend (which is often most evident in the later iterations of the test) is of interest here. Some oscillations (reversals in sign) of the computed head change are normal during convergence; however, repeated oscillation is a sign of overshoot, indicating that computed head changes are too great for optimal convergence. On the other hand, head changes that are too small are indicated by a very flat overall trend. For proper evaluation of the trend, the trial generally should be run for a number of iterations equal to 4 or 5 times the number of iteration variables, unless convergence occurs before this.

Following the initial trial, WSEED is multiplied by a number between 2 and 10 if head changes in the initial trial appear to be too large, and divided by a number between 2 and 10 if those head changes appear to be too small. If the trend in the initial trial is unclear, either multiplication or division of the seed may be tried. In any case, a second run is made using the new WSEED value, and the trend of head change versus iteration level is again examined. The results are compared with those of the initial trial to see if the rate of convergence has improved. If both runs have converged, the comparison is based on the number of iterations required for convergence; if they have failed to converge, the comparison is based on the magnitude of the head changes observed in the final iterations.

The trial runs can be continued to refine further the choice of WSEED; in general, the WSEED value will be multiplied or divided by progressively smaller numbers at each step of the procedure. Carrying the process too far usually is not worthwhile; multiplication or division of the seed by factors less than 2 is seldom warranted.

In most cases, a satisfactory WSEED value developed by this procedure will remain satisfactory even though changes in the model are introduced—for example, additional stresses, modifications in boundary conditions, or changes in the model mesh. If, however, convergence problems arise after such changes, the trial and error procedure can be repeated. It should be noted that the more strongly diagonal the coefficient matrix, the less important the choice of WSEED will be. Thus, source terms such as evapotranspiration or stream seepage, which affect only coefficients on the main diagonal, normally tend to make the choice of WSEED less critical; and the addition of such terms to a model seldom necessitates modification of WSEED.

Improvements in the rate of convergence can be obtained by also adjusting ACCL. Increases in ACCL will cause increases in the head change at each iteration, while decreases in ACCL will cause decreases in head change. The trial procedure described above can be used for this case as well; however, changes in WSEED and in ACCL should not be attempted in the same set of trial runs.

Slowing the process of convergence is sometimes necessary to prevent cells from converting to the no-flow condition as a result of head overshoot during iterations. This requires ACCL to be less than 1. In these situations, optimal convergence cannot be considered convergence in the minimum number of iterations, but rather convergence in the smallest number of iterations that does not involve head overshoot. The procedure of examining maximum head change per iteration and adjusting iteration variables can be used again to determine when this condition is being met, and to develop the required WSEED or ACCL values.

Preconditioned Conjugate-Gradient Package

The Preconditioned Conjugate-Gradient Solver (PCG) Package is fully documented in Hill (1990).

Iteration

The PCG Package uses both inner and outer iteration. To avoid excessive computation time when the solution process is failing to converge, the user specifies the maximum number of outer iterations, MXITER. The simulation is stopped if the convergence criteria are not met within MXITER outer iterations. Convergence is determined by using the head-change and residual criteria described above. Both criteria must be met.

Variable ITER1 specifies the maximum number of inner iterations for each outer iteration. The inner iteration ends when the closure criteria are met or after ITER1 iterations if the closure criteria are not met. When MXITER is 1, convergence is achieved if the closure criteria are met during inner iteration; convergence fails if the closure criteria ends even if the closure criteria are met during inner iteration is started after inner iteration ends even if the closure criteria are met during inner iteration. Convergence is achieved only if the convergence criteria are met in the first inner iteration of an outer iteration. This approach for achieving convergence prevents convergence

7-4 MODFLOW-2005, The U.S. Geological Survey Modular Ground-Water Model

of nonlinear problems until the changed flow equations resulting from a new outer iteration no longer cause additional head and residual changes.

User Guidance

For a linear problem, a single outer iteration with many inner iterations theoretically is adequate, but a few outer iterations should be used for large (greater than 10,000 cells) linear models. For numerical reasons described in Hill (1990, p. 8), the outer iterations improve the accuracy of large linear problems.

Nonlinear problems generally run most efficiently with five or more outer iterations and a large number of inner iterations (20 or more). The solution process becomes inefficient if the nonlinear terms are reformulated frequently by having a small (less than 20) value for ITER1.

The PCG solver provided in MODFLOW–2005 includes an additional variable that can influence convergence. This variable, which is designated DAMP in the program, functions as a multiplier of the head change computed by the PCG algorithm; thus, where no impact is desired, DAMP should be assigned a value of 1. As a general rule, DAMP initially should be given a value of 1. If problems with convergence occur, values of DAMP less than 1 can be tried. Slowing the process of convergence is sometimes necessary to prevent cells from converting to the no-flow condition as a result of head overshoot during an iteration. In these situations, a value of DAMP less than 1 should be used, even though additional iteration may be necessary.

The PCG solver actually incorporates two forms of the Preconditioned Conjugate-Gradient method: the Modified Incomplete Cholesky and the Polynomial methods. Knowledge of the details of these methods generally is not necessary. The most important aspect is that the Modified Incomplete Cholesky method is almost always faster. The Polynomial method is potentially faster only on parallel computers where some of the work can be performed simultaneously on multiple processors. Details can be found in Hill (1990).

When using the Modified Incomplete Cholesky method, the user-specified relaxation parameter called RELAX is provided to reduce the number of iterations for convergence in some situations. The value initially should be set to 1. If convergence fails or takes an unusually large number of iterations, values of 0.99, 0.98, and 0.97 can be used on a trial-and-error basis.

Direct Solver Package

The Direct Solver (DE4) Package is fully documented in Harbaugh (1995).

Iteration

The DE4 Package uses inner or outer iteration, but not at the same time. For a linear problem, a single inner iteration is theoretically adequate. If the budget error is unacceptably large in a linear model, a few inner iterations should be used. For numerical reasons described in Harbaugh (1995, p. 7), the inner iteration improves the accuracy of linear problems. Nonlinear problems require outer iteration; generally five or more outer iterations are needed. To avoid excessive computation time when the solution process is failing to converge, the user specifies the maximum number of iterations, ITMX. The simulation is stopped if the convergence criterion is not met within ITMX iterations. Convergence is determined using the head-change criterion described above.

User Guidance

The use of inner or outer iteration is determined according to the frequency at which the [A] matrix in the flow equation (eq. 2–27) changes. This affects the efficiency of solution; substantial work can be avoided if [A] remains constant during all or part of the simulation. Two situations can cause [A] to change during a simulation. The first is nonlinear terms in the flow equation. A nonlinear term is one in which the term varies with the head that is being computed. An example of this is dependence of conductance on water level under water-table conditions. This can happen in the Block-Centered Flow Package when LAYCON is 1 or 3 and in the Layer-Property Flow Package when LAYTYP is not 0. Similarly, storage terms in [A] vary with head when LAYCON is 2 or 3 or LAYTYP is not

Chapter 7. Solver Packages 7–5

0. Other nonlinear terms are the formulations that change based upon head, such as seepage from a river changing from head dependent flow to a constant flow when head drops below the bottom of the riverbed.

The second situation that causes [A] to change during a simulation is coefficients of simulated head that change with time. For example, riverbed conductance, drain conductance, maximum evapotranspiration rate, evapotranspiration extinction depth, and general-head boundary conductance are coefficients of simulated head, and new values can be read every stress period.

The user indicates the frequency of change in [A] through variable IFREQ. IFREQ = 1 indicates that the flow equations are linear and that coefficients of simulated head for all stress terms are constant for all stress periods. IFREQ = 2 indicates that the flow equations are linear as described for IFREQ=1, but coefficients of simulated head for some stress terms may change at the start of each stress period. For a simulation consisting of only one stress period, IFREQ = 2 has the same meaning as IFREQ = 1. IFREQ = 3 indicates that a nonlinear flow equation is being solved, which means that some terms in [A] depend on simulated head.

If time steps are not equal, [A] changes every time step even if the problem is linear because the storage term depends on the length of the time step. The user does not have to recognize this situation because the DE4 solver automatically recognizes that [A] will change when the time step length changes. The use of non-equal time steps, however, results in less efficiency (longer execution times) for linear problems. Users wanting to solve linear flow equations most efficiently should consider using stress periods with equal-length time steps. There is no execution-time penalty for unequal time steps in nonlinear problems.

If a nonlinear flow equation is being solved, outer iteration (ITMX > 1) is required. When nonlinearities caused by water-table calculations are part of a simulation, obvious signs may not necessarily be evident in the output from a simulation that does not use external iteration to indicate that iteration is needed. In particular, the budget error may be acceptably small without iteration even though error in head is substantial because of nonlinearity. Therefore, the user must be careful to specify a value of 3 for IFREQ when a problem is nonlinear.

To understand why signs that iteration is needed for nonlinear equations may not be obvious, consider the water-table computation of transmissivity. During each iteration, a new transmissivity is calculated based on the previous head. Then the flow equations are solved, and a budget is computed using the new head with the same transmissivities. No budget discrepancy results because heads are correct for the transmissivity being used at this point; however, the new heads may indicate a substantial change in transmissivity. The new transmissivity will not be calculated unless another iteration occurs, which causes the Formulate Procedure to be invoked. Therefore, when one or more layers are under water-table conditions, iteration should always be tried. The maximum change in head for each iteration (printed by DE4 when IPRD4 = 1 and MUTD4 = 0) provides an indication of the impact of all nonlinearities.

DE4 uses a specialized form of direct solution that is optimized for sparse, symmetric matrices. The user need not know the details of the solution method, but there is one aspect of solution for which the user must have knowledge. When the equations are solved, they are divided into two parts, an upper and lower part. Memory must be allocated for both parts. The program can attempt to determine the required memory, but in some situations the automatic approach is not optimal. Automatic allocation is used when variables MXLOW, MXUP, and MXBW are specified as 0. Unless a large number of constant-head or no-flow cells are present or the automatic method results in the message "INSUFFICIENT MEMORY FOR DE4 SOLVER" in the Listing File, then the automatic allocation. The Listing File will show the required values for MXLOW, MXUP and MXBW, and the user can modify the DE4 Package input file to specify these values.

If a large number of constant-head or no-flow cells exist, then some memory can possibly be saved by specifying non-zero values for MXUP, MXLOW, and MXBW. This is generally not worth the effort, however, unless computer memory is constrained. If the user desires or needs to specify values, the best way to do this is to run the model with MXUP, MXLOW, and MXBW set to 0, and then look in the Listing File to find the optimal values. The model can then be rerun using the new values.