**USGS**
*science for a changing world*

Techniques of Water-Resources Investigations
of the United States Geological Survey

Chapter A5

# A MODULAR FINITE-ELEMENT MODEL (MODFE) FOR AREAL AND AXISYMMETRIC GROUND-WATER-FLOW PROBLEMS, PART 3: DESIGN PHILOSOPHY AND PROGRAMMING DETAILS

By Lynn J. Torak

Book 6
MODELING TECHNIQUES

evaluated in subroutine INITCG. Details about the allocation of computer storage in MODFE are given in the following section.

Components of the C, V, and G matrices and B vector of equations (1)–(5), element areas, and node incidences are written to computer files by subroutine SETCG. This process enables storage locations that are occupied by these terms to be overwritten during solution by the MICCG method. Overwriting storage locations by the solution methods is described in the section "Solution Methods."

Computations for solving equations (1)–(5) by the MICCG method are contained in subroutine MICCG. The solution process begins by approximately factoring the reduced matrix into an upper triangular matrix, U, which is represented by program vector AF. The factorization is performed according to equation (274) in Cooley (1992), and is modified by the row-sums agreement given by equations (276)–(278) in Cooley (1992). An intermediate vector, $\tilde{y}$, is computed by forward substitution, and the displacement vector, s, is computed by backward substitution according to equations (283) and (284) in Cooley (1992). These computations are identified in the subroutine by comment statements.

Computations for the generalized conjugate-gradient algorithm of equation (271) in Cooley (1992) are performed in subroutine MICCG by formulating iteration parameters $\beta_k$ and the A-orthogonal vectors, $\underline{p}_k$. The iteration parameter used for each vector is represented by program variable S, and the vectors are represented by program variable P. The scaled displacements, $\alpha_k\underline{p}_k$, are represented by program variable TMPA, and are summed for each equation and stored in program variable B to give the total displacement from the initial condition. Similarly, scaled residuals, $-\alpha_k\underline{\underline{A}}\underline{p}_k$, are computed for each equation, and the sum is stored in program variable R. The largest absolute value of the displacements is determined according to equation (285) in Cooley (1992) by comparing absolute values of TMPA, and the result is represented as program variable PMAX. Similarly, the largest absolute scaled residual, given by equation (289) in Cooley (1992) is determined, and the result is represented by program variable RMAX. The iterative process is concluded (has converged) when values of RMAX and PMAX are less than the closure tolerance, which is represented by program variable TOL.

Messages are printed out from subroutine MICCG that describe the progress toward attaining a solution. If convergence is achieved, then a message is printed to that effect along with the number of iterations that was required. If convergence was not achieved, then a message is printed indicating that the solution failed to converge, followed by values for the maximum number of iterations (NIT), the largest absolute scaled residual (RMAX), and the computed head changes (program vector B).

# Allocation of Computer Storage and Processing Time

Several techniques for storing and representing program variables have been used during the design of MODFE to minimize computer storage and processing time. A single program vector, G, which is dimensioned at the time of execution of MODFE, is used to store most of the information that is needed to conduct a simulation. Components of matrix equations (1)–(5) are stored within vector G in a manner that decreases the range of indexes that are searched by the computer during assembly of the matrix equations. The order of the coefficient matrix is decreased by eliminating specified-head nodes from the solution process, thereby allowing a reduced-matrix equation to be solved. Use of the reduced matrix for solution results in fewer equations and storage locations than if specified-head nodes were retained in the formulation. Components of the reduced matrix are stored in a condensed form, thus eliminating as many zero entries from storage as possible. In addition, storage locations within program vector G are overwritten, or reused, routinely during solution of the matrix equations. Details of these techniques are given in the following sections.

## General-Storage Vector G

Nearly all program information that is needed to form and solve matrix equations (1)–(5) and to compute a water-balance summary is contained in the general-storage vector G. The vector G is subdivided into smaller vector lengths of storage, and all values are stored in single-subscripted form. Computer storage is allocated at the time MODFE is executed by determining the beginning locations (starting addresses) of the vector lengths that correspond to program vectors stored in general-storage vector G. Values for program variables that represent starting addresses are computed automatically by MODFE from inputs that define program dimensions and problem specifications (see section "Input Instructions" in Torak (1993)). Descriptions of the terms for which computer storage is allocated within the general-storage vector G and names of subroutines that perform the storage allocation are given in table 6. Program variables that represent starting addresses, vector lengths, and vectors stored within the general-storage vector G are given in the section "Variable Lists and Definitions."

The amount of computer storage that is allocated to the general-storage vector G is assigned by the main programs of MODFE in a Fortran statement given as COMMON/PRIME/G(nnnn), where "nnnn" defines the number of single-precision storage locations assigned to vector G. During execution of MODFE, the number of storage locations that actually is used to dimension program vectors is printed out by the subroutines listed in table 6. This value is represented by program variable ISUM. For MODFE to execute successfully, the value of nnnn must be equal to or greater than the value of ISUM. Hence, the user must determine the size of the simulation (value of ISUM) and make necessary adjustments to the dimension of the general-storage vector G in the COMMON statement so that the G vector can accommodate all storage requirements.

The first five vector lengths within the general-storage vector G are overwritten and reused routinely when forming and solving matrix equations (1)–(5). These locations contain terms for the C, V, and G matrices and B vector (stored in condensed form in program vector A) of these equations, element areas, element incidences, and x and y coordinates of nodes. Values contained in the first three vector lengths of general-storage vector G (nodal coordinates are excluded) are written to computer files by subroutines SETB and SETCG prior to being overwritten in subroutines FMEQ and WTFMEQ, where the matrix equations are assembled in condensed form. These locations in general-storage vector G are overwritten again by the equation-solving subroutines, BAND and MICCG, as the condensed-matrix equations are expanded to the reduced-matrix form. Values that were written to the computer files are read and placed in the original storage locations in general-storage vector G by subroutine RDTP for forming and solving equations on the corrector step (if appropriate) and for computing the water-balance summary. Details about forming the condensed matrix, reduced matrix, and the expansion of the condensed matrix to the reduced form during solution are given in the following sections.

To ensure that the first five storage locations in the general-storage vector G provide at least as much computer storage as required by subroutines BAND and MICCG, the value of ISUM is compared with the storage requirements of both solution methods after computer storage for the fifth location is calculated. The amount of computer storage required by either solution method is represented in the corresponding subroutines, INITB and INICG, as program variable NC. If ISUM is less than NC, then ISUM is assigned the value of NC + 1.

Other storage locations in the general-storage vector G than those described above are reused during simulation. These are identified in the section "Variable Lists and Definitions" as multiple descriptions for storage locations in the general-storage vector G.

## Reduced Matrix A

Matrix equations (1)–(5) are placed in what is termed a reduced form in MODFE by eliminating equations at specified-head boundaries (nodes) from the solution process. This decreases the order of the coefficient matrix that is formed, and creates a smaller matrix termed the reduced matrix A. Use of the reduced matrix A results in a savings of computer storage and processing time from what would be needed to form and solve a larger matrix containing entries for specified-head nodes.

Formation of the reduced Matrix A is explained by the following example. Given a finite-element mesh consisting of 3 elements and 5 nodes (fig. 15*A*), the coefficient matrix contains nonzero entries at the locations indicated in figure 15*B*. Values of the coefficients are determined according to the matrix equation that is solved, but generally consist of elements of the C, V, and G matrices in equations (1)–(5). Each entry to the coefficient matrix multiplies an entry in the solution vector, which is either a head change, $\delta$, or a displacement, $\delta_l$, depending on the the equation that is solved. Because the head change or displacement is known at specified-head nodes, these equations are be removed from the matrix equation, and terms in the remaining equations that multiply the known head change or displacement at the specified-head nodes are placed on the right side of the matrix equation that is formed. The result is a reduced matrix A of a lower order and fewer equations to be solved than the coefficient matrix (fig. 15*C*); thus, decreasing computer storage and processing time.

Computer storage for forming and solving the finite-element matrix equations is allocated on the basis of the dimensions of the reduced matrix A. Values for the number of nodes and specified-head boundaries, program variables NNDS and NHDS, respectively, are used to determine the order of the reduced matrix (fig. 15*C*) and the number of equations that are formed. The order of the reduced-matrix, given by the difference NNDS–NHDS, is used in MODFE to eliminate matrix assembly and solution at nodes that represent specified-head boundaries. The reduced-matrix bandwidth, program variable MBW (fig. 15*C*), is used to allocate storage for elements of

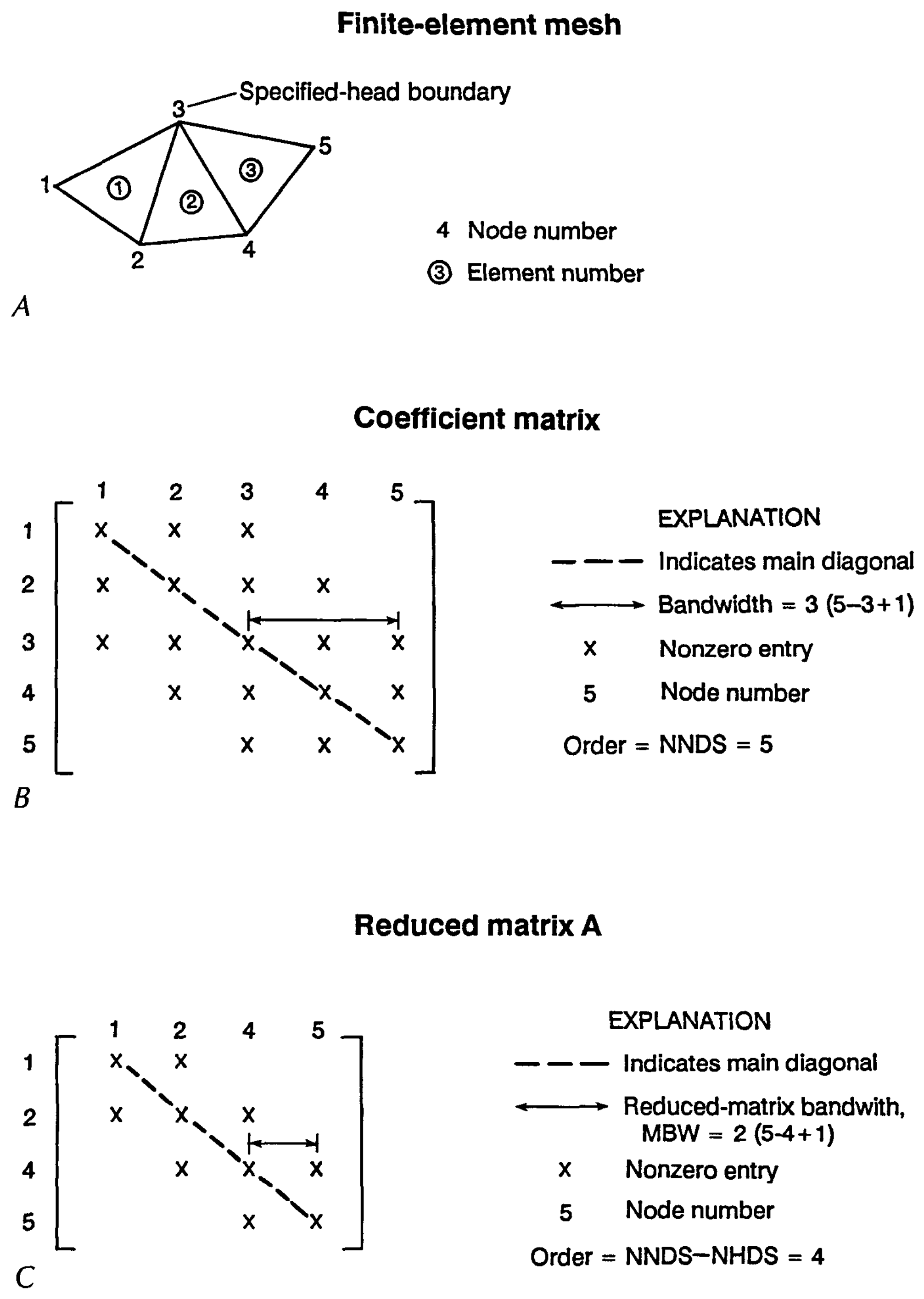

Figure 15.—(*A*) three-element, five-node, finite-element mesh containing a specified-head boundary at node 3, (*B*) coefficient matrix, and (*C*) reduced matrix A.

reduced matrix A that are needed to solve matrix equations (1)–(5) by the direct-solution method (subroutine BAND). Because of symmetry, only entries on and to the right of the main diagonal in reduced matrix A (fig. 15*C*) are needed for solution by subroutine BAND.

A value for the reduced-matrix bandwidth, MBW, is required as input to subroutine INITB. Details for estimating the maximum value of the reduced-matrix bandwidth for input as MBW are given in Torak (1993). Although computer storage is allocated for the direct-solution method based on the value of MBW that is input, this value is only an estimate of the reduced-matrix bandwidth. The actual reduced-matrix bandwidth, IBND, is computed by subroutine SETB and is used by subroutine BAND to determine the amount of computer storage that actually is used during solution by the direct method. Details of the allocation of computer storage for the solution methods are given in the section "Solution Methods."

## Reordering Finite-Element Equations for Solution

Formation of reduced matrix A by eliminating equations at specified-head nodes causes a reordering of the sequence in which head changes or displacements at the remaining nodes are solved. In the example shown in figure 15, coefficients to four equations are represented by the reduced matrix A, and the third and fourth equations correspond to nodes 4 and 5, respectively. An indexing method is used by MODFE that reorders nodes that are not representing specified-head boundaries and numbers equations for solution. The same indexing method also sequences specified-head boundaries for computations in the water-balance summary. This indexing method is independent of the node numbering; original node numbers are unchanged by the indexing.

The indexing method for reordering finite-element equations for solution is implemented in MODFE by storing sequential equation numbers and numbers for specified-head boundaries in an indicator vector. Each node in the finite-element mesh is ordered (sequenced) by the indicator vector, which is represented by program vector IN. Nodes on specified-head boundaries are sequenced by using negative values in the corresponding locations of the indicator vector. The remaining nodes are assigned positive sequence numbers which represent the order of equation formation and solution (table 18).

The indicator vector is evaluated in subroutine DATIN during input of specified-head boundaries. Checks are made on the value of the indicator vector in subroutines that form and solve the matrix equations so that specified-head nodes are eliminated from these computations and that equations at the remaining nodes are formed and solved sequentially.

## Condensed Matrix

Components of the C, V, and G matrices of equations (1)-(5) are stored in a compact, or condensed form that saves computer storage and processing time thereby increasing the computational efficiency of MODFE. The condensed matrix contains only the nonzero entries of reduced matrix A that are located on and to the right of the main diagonal (fig. 16*A*,*B*). Because of symmetry in the reduced matrix, entries to the left of the main diagonal are identical to entries that are located in the corresponding transpose positions to the right of the main diagonal. The main diagonal of reduced matrix A is stored as the first column of the condensed matrix, and nonzero terms to the right of the main diagonal in the reduced matrix are stored in consecutive locations to the right of the first column in the condensed matrix.

Table 18.—Values for indicator vector, IN, corresponding to nodes in finite-element mesh in Figure 15

| Node I | IN(I) |
|---|---|
| 1 | 1 |
| 2 | 2 |
| 3* | -1 |
| 4 | 3 |
| 5 | 4 |

* Specified-head node

Although the reduced matrix A and the condensed matrix are shown in matrix form (figs. 15 and 16), they are computed and stored in vector form as program vector A. A fixed number of storage locations in program vector A is used to represent each row of the condensed matrix. The number of storage locations assigned to each row of the condensed matrix is termed the condensed-matrix bandwidth, and is determined by the maximum number of nonzero entries in each row of reduced matrix A. The condensed-matrix bandwidth is represented in MODFE as program variable MBWC (fig. 16*B*). The value of MBWC is input to MODFE in subroutines INITB and INITCG.

The condensed matrix is assembled in the equation-forming subroutines (table 9) by accumulating components of the C, V, and G matrices of equations (1)–(5) and storing the results in the appropriate locations within program vector A (fig. 16*C*). This assembly overwrites locations in program vector A that contain components of the C, V, and G matrices for all nodes. During assembly by subroutines FMECWT, FMEPWT, and FMEQ, main-diagonal entries are first stored in program vector AD and then transferred to program vector A.

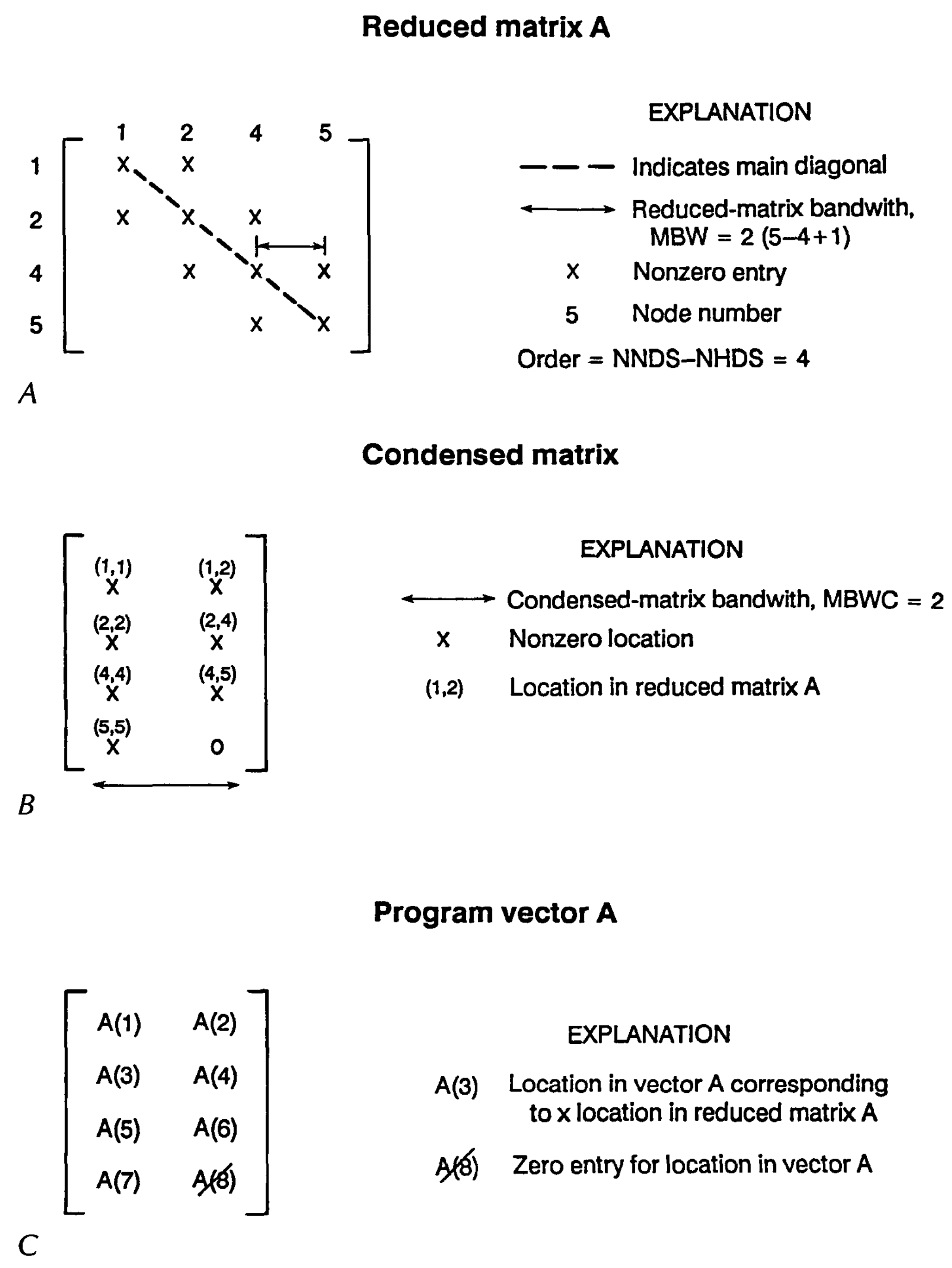
Reduced matrix A
EXPLANATION
Indicates main diagonal
Reduced-matrix bandwith, MBW = 2 (5–4+1)
X Nonzero entry
5 Node number
Order = NNDS–NHDS = 4
A
Condensed matrix
EXPLANATION
Condensed-matrix bandwith, MBWC = 2
X Nonzero location
(1,2) Location in reduced matrix A
B
Program vector A
A(1) A(2)
A(3) A(4)
A(5) A(6)
A(7) A(8)
EXPLANATION
A(3) Location in vector A corresponding to x location in reduced matrix A
A(8) Zero entry for location in vector A
C

Figure 16.—(*A*) reduced matrix A, (*B*) condensed matrix, and (*C*) condensed matrix represented by program vector A.

Components of the C, V, and G matrices that are used to form the condensed matrix are stored in a compact manner within program vector A, which allows matrix assembly to be computationally efficient. For each node, components of the C, V, and G matrices are stored in consecutive locations within program vector A (fig. 17). The amount of computer storage that is assigned to program vector A for matrix components of all nodes is computed in subroutines INITB and ININTCG as program variable NA (fig. 17). The amount of storage in program vector A that is assigned to each node is a function of the maximum condensed-matrix bandwidth, MBWC, and is defined as program variable IW, where IW = MBWC + 1. Components for the condensed matrix are assembled by incrementing indexes to program

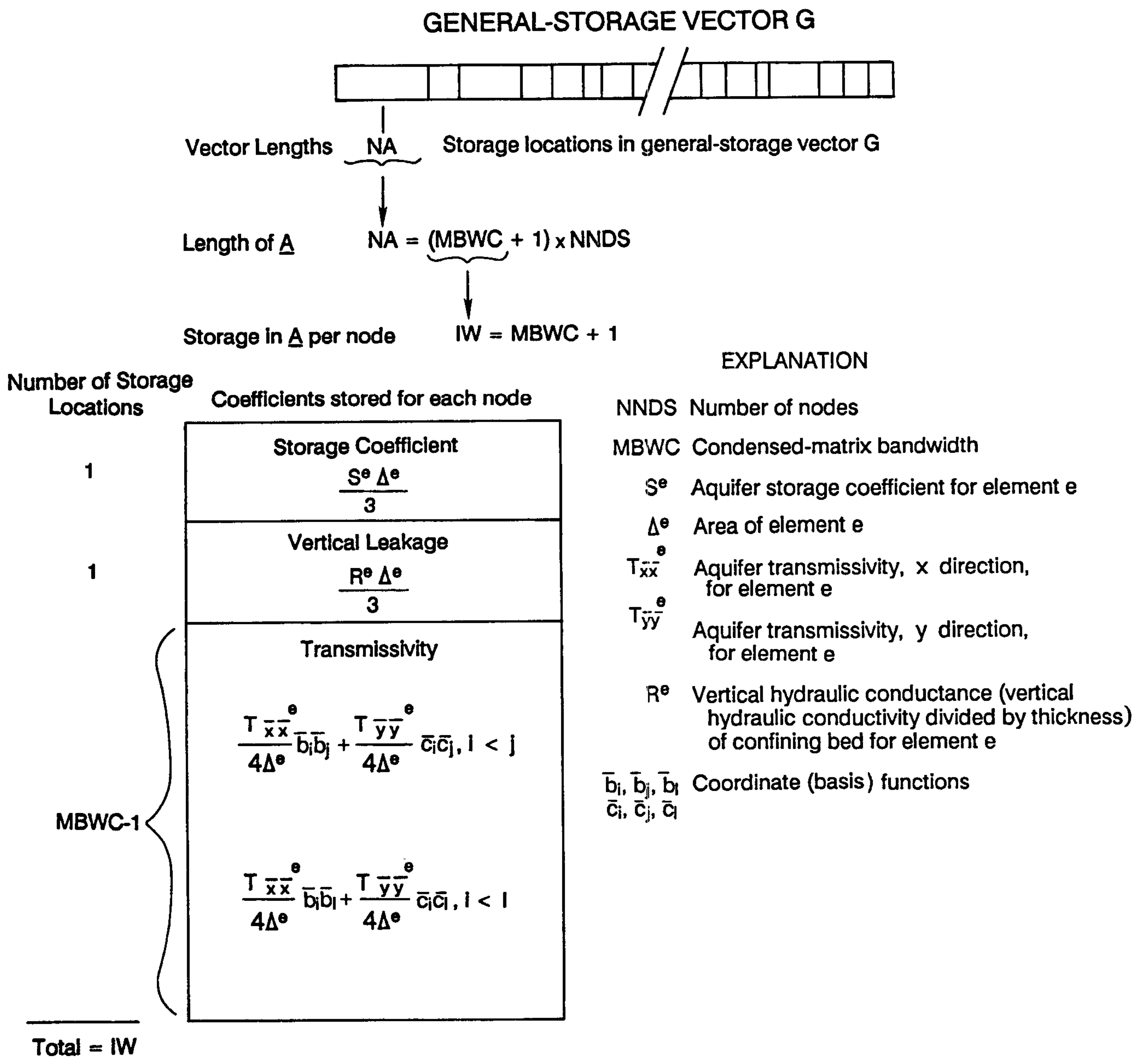
GENERAL-STORAGE VECTOR G
Vector Lengths
NA
Storage locations in general-storage vector G
Length of A
NA = (MBWC + 1) x NNDS
Storage in A per node
IW = MBWC + 1
Number of Storage Locations
Coefficients stored for each node
1
Storage Coefficient
$\frac{S^e \Delta^e}{3}$
1
Vertical Leakage
$\frac{R^e \Delta^e}{3}$
MBWC-1
Transmissivity
$\frac{T_{\bar{x}\bar{x}}^e}{4\Delta^e}\bar{b}_i\bar{b}_j + \frac{T_{\bar{y}\bar{y}}^e}{4\Delta^e}\bar{c}_i\bar{c}_j, i < j$
$\frac{T_{\bar{x}\bar{x}}^e}{4\Delta^e}\bar{b}_i\bar{b}_l + \frac{T_{\bar{y}\bar{y}}^e}{4\Delta^e}\bar{c}_i\bar{c}_l, i < l$
Total = IW
EXPLANATION
NNDS Number of nodes
MBWC Condensed-matrix bandwidth
$S^e$ Aquifer storage coefficient for element e
$\Delta^e$ Area of element e
$T_{\bar{x}\bar{x}}^e$ Aquifer transmissivity, x direction, for element e
$T_{\bar{y}\bar{y}}^e$ Aquifer transmissivity, y direction, for element e
$R^e$ Vertical hydraulic conductance (vertical hydraulic conductivity divided by thickness) of confining bed for element e
$\bar{b}_i, \bar{b}_j, \bar{b}_l$ $\bar{c}_i, \bar{c}_j, \bar{c}_l$ Coordinate (basis) functions

Figure 17.—Storage locations in general-storage vector G and matrix components stored in program vector A.

vector A by multiples of IW, where the number of multiples of IW corresponds to the difference between node numbers in an element. Because the node numbers within an element usually are close in value, all components that are needed to assemble the condensed matrix at a node are stored within a small number of multiples of IW storage locations in program vector A. In addition, node numbering of the finite-element mesh that minimizes the reduced matrix bandwidth also minimizes the number of multiples of IW storage locations that need to be indexed by the computer in order to locate components of the condensed matrix for any node. By placing all matrix components for a node in consecutive storage locations, matrix assembly is more computationally efficient than if each component, such as storage coefficient, vertical leakage, and transmissivity, were represented by individual matrices or vectors.

## Solution Methods

Computer storage for the solution methods is allocated in the general-storage vector G by subroutine INITB for the direct method and by subroutine

INITCG for the conjugate-gradient method. The amount of single-precision storage that is required by each solution method is represented in these subroutines as program variable NC. For the direct method, NC is computed as

$$\mathrm{NC} = \mathrm{MBW} \times (\mathrm{NNDS} - \mathrm{NHDS} - \mathrm{MBW}) + (\mathrm{MBW} \times (\mathrm{MBW} + 1))/2, \quad (10)$$

where MBW is the estimate of the reduced-matrix bandwidth, NNDS is the number of nodes, and NHDS is the number of specified-head boundaries. Values for these terms are input to subroutine INITB prior to computing NC.

For the conjugate-gradient method, the amount of computer storage that is required is computed as

$$\mathrm{NC} = \mathrm{NA} + (\mathrm{MBWC} + 3) \times (\mathrm{NNDS} - \mathrm{NHDS}), (11)$$

where MBWC is the condensed-matrix bandwidth and NA = (MBWC + 1) × NNDS. A discussion of bandwidth determination is given in the section "Node Numbering and Determining Bandwidth" in Torak (1993). Descriptions of the use of MBWC and MBW in computations and in allocating computer storage have been given in preceding sections.

A comparison of terms in equations (10) and (11) that are used to define storage requirements for both solution methods indicates that computer storage for the direct method is dependent on the reduced-matrix bandwidth, MBW, while computer storage for the conjugate-gradient method is dependent on the condensed-matrix bandwidth, MBWC. As discussed in the section "Node Numbering and Determining Bandwidth" in Torak (1993), the reduced-matrix bandwidth is a function of node numbering in the finite-element mesh, and the condensed-matrix bandwidth is a function of the mesh design. Numbering nodes so that the maximum difference between node numbers is minimized in elements that do not contain specified-head boundaries will decrease the reduced-matrix bandwidth, MBW, and decrease computer storage requirements for the direct-solution method. The condensed-matrix bandwidth, MBWC, is determined by the maximum number of connections from a node in the center of a patch of elements to higher-numbered nodes in the patch. Thus, by eliminating excessive connections of element sides to a node, the condensed-matrix bandwidth can be minimized, thereby decreasing computer storage for the conjugate-gradient method.

Although computer storage for the conjugate-gradient method is not dependent on node numbering of the finite-element mesh, and computer storage for the direct method is not dependent on mesh design, both node numbering and mesh design affect computer storage and processing time for both solution methods and for computations of the water-balance summary. Node numbering to minimize the reduced-matrix bandwidth, MBW, decreases the computer-processing time needed to form the condensed matrix and to compute a water-balance summary by minimizing the number of sets of IW storage locations that separate the IW locations corresponding to nodes in an element (see discussion of storage locations for each node in the preceding section). Similarly, a mesh design that minimizes the condensed-matrix bandwidth, MBWC, also decreases computer storage and processing time by eliminating unused storage locations that are assigned to the set of IW locations for each node. Usually, excessively large values of MBWC exist at only a few nodes in a mesh that is designed by hand. However, computer storage and processing time for all nodes are affected by the excessively large value of MBWC, which satisfies storage requirements at just a few nodes. Instructions about designing a finite-element mesh to minimize MBW and MBWC are given in Torak (1993).

## References

Cooley, R.L., 1992, A MODular Finite-Element model (MODFE) for areal and axisymmetric ground-water-flow problems, part 2: derivation of finite-element equations and comparisons with analytical solutions: U.S. Geological Survey Techniques of Water Resources Investigations, Book 6, Chapter A4.

Torak, L.J., 1993, A MODular Finite-Element model (MODFE) for areal and axisymmetric ground-water-flow problems, part 1: model description and user's manual: U.S. Geological Survey Techniques of Water Resources Investigations, Book 6, Chapter A3.

# APPENDIXES

Lists of program variables and definitions and of subroutines are given as appendixes. Program variables are listed and defined according to the criteria described in the following section. Subroutines are listed in alphabetical order following the variable lists.

## Variable Lists and Definitions

Program variables are listed and defined according to their usage in MODFE. Where appropriate, tables list program variables that are used in Fortran CALL statements and their corresponding starting locations in the general-storage vector G. The following criteria were used to group program variables and definitions for this appendix.

- Fortran COMMON statements.
  COMMON statements are listed in alphabetical order. Program variables are listed and defined in the sequence in which they appear in each COMMON statement.
- General-storage vector G.
  Starting-location variables, lengths of vector storage in the general-storage vector G, and program variables that are represented by the storage locations are listed and defined in the sequence in which vector G is filled. General-storage vector G is filled according to the following sequence:
  - starting-location variables in Fortran COMMON/ ADR/,
  - conjugate-gradient method of solution,
  - water-table (unconfined) conditions,
  - nonlinear, head-dependent (Cauchy-type) flux, point sinks, and steady vertical leakage, and
  - vertical leakage of water stored elastically in a confining bed (transient leakage).
- Text for general-storage vector G.
  Computer storage for the above simulation capabilities is not allocated and starting locations are eliminated if MODFE does not contain the corresponding program structure. Reuse of storage within the general storage vector G is defined by multiple definitions or program variables for the same storage locations.
- Main program.
  Program variables that are not listed in Fortran COMMON statements, but are used in the main program and in subroutines to control computational steps.
- Linear versions.
  Program variables that are not listed in Fortran COMMON statements are listed and defined in alphabetical order.
- Transient leakage.
  Subroutines are listed in alphabetical order and program variables within each subroutine are listed and defined in alphabetical order.
- Changing time-step sizes, stresses, and boundary conditions.
  Subroutines and program variables within each subroutine are listed in alphabetical order.
- Nonlinear versions.
  Program variables that are not listed in Fortran COMMON statements are listed and defined in alphabetical order for each nonlinear hydrologic term or simulation feature according to the following sequence:
  - water-table (unconfined) conditions,
  - head-dependent (Cauchy-type) flux and point sinks,
  - steady vertical leakage, and
  - steady-state conditions.
- Solution methods.
  Subroutine names and program variables are listed and defined in alphabetical order. Program variables for the direct-solution method are presented first, followed by program variables for the conjugate-gradient method.

## Fortran COMMON Statements

| | Variable | Definition |
|---|---|---|
| COMMON/BAL/ | SA | Volumetric rate of accumulation of water in aquifer storage [length$^3$/time]. |
| | WQI | Volumetric recharge rate to aquifer by point sources [length$^3$/time]. |
| | WQO | Volumetric discharge rate from aquifer by point sinks [length$^3$/time]. |
| | DQI | Volumetric recharge rate to aquifer by areally distributed sources [length$^3$/time]. |
| | DQO | Volumetric recharge rate from aquifer by areally distributed sinks [length$^3$/time]. |
| | VLQI | Volumetric recharge rate to aquifer by vertical leakage from a confining bed (steady and transient leakage [length$^3$/time]. |
| | VLQO | Volumetric discharge rate from aquifer by vertical leakage from a confining bed (steady and transient leakage) [length$^3$/time]. |
| | BQI | Volumetric recharge rate to aquifer by Cauchy-type boundaries [length$^3$/time]. |
| | BQO | Volumetric discharge rate from aquifer by Cauchy-type boundaries [length$^3$/time]. |
| | ER | Flow imbalance of volumetric rates for time step [length$^3$/time]. |
| COMMON/CHG | | |
| | NWCH | Number of the time step when point sources or sinks are changed. |
| | NQCH | Number of the time step when areally distributed sources or sinks are changed. |

COMMON/CHG/--continued

| Variable | Definition |
|---|---|
| NHRCH | Number of the time step when values of source-layer heads HR are changed in zones simulating steady-leakage (no transient effects from confining-bed storage). |
| NBQCH | Number of the time step when specified flux or head-dependent (Cauchy-type) flux is changed. |
| NHCH | Number of the time step when values of specified-head boundaries change. |
| NCBCH | Number of the time step when values of source-layer heads NR are changed in zones simulating transient leakage. |
| NVNCH | Number of the time step when controlling heads HS to nonlinear vertical-leakage functions are changed. |
| NGNCH | Number of the time step when controlling heads HRK and HRL to nonlinear, head-dependent (Cauchy-type) fluxes are changed. |

COMMON/GDIM/

| | |
|---|---|
| ISUM | Number of single-precision words of storage in $\underline{G}$ required to execute MODFE. |

COMMON/GNBL/

| | |
|---|---|
| BNQI | Volumetric recharge rate to aquifer by nonlinear head-dependent (Cauchy-type) fluxes [length$^3$/time]. |
| BNQO | Volumetric discharge rate from aquifer by nonlinear head-dependent (Cauchy-type) fluxes [length$^3$/time]. |
| TBNQI | Total volume of water [length$^3$] recharged to aquifer by nonlinear, head-dependent (Cauchy-type) boundaries. |
| TBNQO | Total volume of water [length$^3$] discharged from aquifer by nonlinear, head-dependent (Cauchy-type) boundaries. |

COMMON/GNBL/--continued

| Variable | Definition |
|---|---|
| PNQO | Volumetric discharge rate from aquifer by nonlinear point sinks [length$^3$/time]. |
| TPNQO | Total volume of water [length$^3$] discharged from aquifer by nonlinear point sinks. |

COMMON/IND/

| | |
|---|---|
| IRAD | Indicator for axisymmetric (radial) flow. |
| IUNIT | Indicator for converting lengths from map-scale units to field units. |
| ISTD | Indicator for steady-state simulations. |

COMMON/IPRN/

| | |
|---|---|
| IPND | Indicator variable to suppress printout of node numbers for each element. |

COMMON/ITP/

| | |
|---|---|
| IIN | Fortran-unit number for reading input data (=50). |
| IOUT | Fortran-unit number for program output (=60). |
| ITA | Fortran-unit number for writing the A vector to peripheral storage (=55). |
| ITB | Fortran-unit number for writing vectors AR and ND to peripheral storage (=56). |

COMMON/NO/

| | |
|---|---|
| NELS | Number of elements. |
| NNDS | Number of nodes. |

COMMON/NO/--continued

| | |
|---|---|
| NSTEPS | Number of time steps. |
| NPER | Number of stress periods. |
| NZNS | Number of aquifer-property zones. |
| NWELS | Number of point sources or sinks. |
| NQBND | Number of Cauchy-type-boundary element sides. |
| NHDS | Number of specified-head-boundary nodes. |
| NEQ | Number of equations from reduced matrix. |
| MBWC | Maximum condensed-matrix bandwidth. |
| MBW | Maximum reduced-matrix bandwidth (if direct-solution method is used to solve equations). |
| NIT | Maximum number of iterations (if iterative method, MICCG, is used to solve equations). |

COMMON/PRIME/

| | |
|---|---|
| G(5000) | The general-storage vector G and its initial dimension. |

COMMON/SCLE/

| | |
|---|---|
| SCALE | Scaling factor to convert map units of length to field units. |

COMMON/TBAL/

| | |
|---|---|
| TSA | Total volume of water [length$^3$] accumulated in aquifer storage during the simulation. |
| TWQI | Total volume of water [length$^3$] recharged to aquifer during simulation by point sources. |
| TWQO | Total volume of water [length$^3$] discharged from aquifer during simulation by point sinks. |
| TDQI | Total volume of water [length$^3$] recharged to aquifer during simulation by areally distributed sources. |

COMMON/TBAL (continued)

| Variable | Definition |
|---|---|
| TDQO | Total volume of water [length$^3$] discharged from aquifer during simulation by areally distributed sinks. |
| TLQI | Total volume of water [length$^3$] recharged to aquifer during simulation by vertical leakage (steady and transient leakage). |
| TLQO | Total volume of water [length$^3$] discharged from aquifer during simulation by vertical leakage (steady and transient leakage). |
| TBQI | Total volume of water [length$^3$] recharged to aquifer during simulation by Cauchy-type boundaries. |
| TBQO | Total volume of water [length$^3$] discharged to aquifer during simulation by Cauchy-type boundaries. |
| THBQI | Total volume of water [length$^3$] recharged to aquifer during simulation by specified-head boundaries. |
| THBQO | Total volume of water [length$^3$] discharged to aquifer during simulation by specified-head boundaries. |
| TER | Flow imbalance of total volumes for simulation [length$^3$]. |

COMMON/VNLBL/

| | |
|---|---|
| VNLQI | Volumetric inflow rate (recharge) to aquifer by nonlinear steady-leakage functions [length$^3$/time]. |
| VNLQO | Volumetric outflow rate (discharge) from aquifer by nonlinear steady-leakage functions [length$^3$/time]. |
| TNLQI | Total volume of water (length$^3$) recharged to aquifer during simulation by nonlinear steady-leakage functions. |
| TNLQO | Total volume of water (length$^3$) discharged from aquifer during simulation by nonlinear steady-leakage functions. |

## General Storage Vector G

Starting-location variables in Fortran COMMON/ADR/

| Starting location variable | Vector length | Variable in G | Definition |
|---|---|---|---|
| IAA | (MBWC+1) × NNDS | A | Contains partially formulated coefficients for matrix equation of each node. |
| IARA | 2 × NELS | AR | One-third the area of element e, $(1/3)\Delta^e$. |
| | | AD | Temporary storage of main diagonal of upper-triangular matrix A. |
| | | DTK | Predicted thickness change [length]. |
| | | VQK | Volumetric flow rate [length$^3$/time] for node K on head-dependent (Cauchy-type) boundary. |
| INDA | 4 × NELS | ND | Node numbers for each element. |
| IXGA | NNDS | XG | Global x coordinates of nodes. |
| | | VQL | Volumetric flow rate [length$^3$/time] for node L on head-dependent (Cauchy-type) boundary. |
| IYGA | NNDS | YG | Global y coordinates of nodes. |
| | | DTK | Adjusted thickness changes over time step. |
| | | R | Volumetric flow rates at specified head boundaries [length$^3$/time]. |
| | | WVCN | Effective vertical hydraulic conductivity $\sum_{e_i} K'^{e}_{zz} \Delta^{e}$ for node i [length$^3$/time]. |
| IATA | MBW | AT | Temporary variable used to store one row of upper triangularized A matrix in subroutine BAND. |
| IQA | NNDS | Q | Sum of known stresses at each node [length$^3$/time]. |
| IBA | NNDS-NHDS | IZN | Zone numbers for each element. |

## General Storage Vector G

### Starting-location variables in Fortran COMMON/ADR/--continued

| Starting location variable | Vector length | Variable in G | Definition |
|---|---|---|---|
| IHA | NNDS | H | Hydraulic head [length]. |
| IHRA | NNDS | HR | Source-layer head [length]. |
| IHBA | NNDS | DHB | Head difference over a time step at specified-head boundary [length]. |
| IALA | NQBND | ALPH | The $\alpha$ term in equation 4 in Cooley (1992) [length/time] for head-dependent (Cauch-type) boundaries. |
| IQBA | NQBND | QBND | The $q_B$ term in equation 4 in Cooley (1992) [length$^2$/time] for specified-flux boundaries, if $\alpha = 0$, or, $q_B/\alpha$ if $q_B$ and $\alpha$ are nonzero. |
| ICKA | NQBND | CFDK | The coefficient $(\alpha L)_{ik}/2$ if $\alpha \neq 0$, or $L_{ik}/2$ if $\alpha = 0$ and $q_B \neq 0$, for node k on a Cauchy-type boundary. |
| ICLA | NQBND | CFDL | The coefficient $(\alpha L)_{ik}/2$ if $\alpha \neq 0$, or $L_{ik}/2$ if $\alpha = 0$ and $q_B \neq 0$, for node l on a Cauchy-type boundary. |
| IHKA | NQBND | HK | Head external to aquifer region for Cauchy-type boundary at node k [length]. |
| IHLA | NQBND | HL | Head external to aquifer region for Cauchy-type boundary at node l [length]. |
| IDTA | MXSTPS | DELT | Time-step sizes [time]. |
| IJPA | (MBWC-1)×NNDS | JPT | Pointer vector, stores node numbers of off-diagonal terms in the upper-triangular form of coefficient matrix A. |

Starting-location variables in Fortran COMMON/ADR/--continued

| Starting location variable | Vector length | Variable in G | Definition |
|---|---|---|---|
| INA | NNDS+1 | IN | Indicator vector for ordering equations and specified-head-boundary nodes. |
| IKA | NQBND | KQB | Node k on a head-dependent (Cauchy-type) boundary. |
| ILA | NQBDN | LQB | Node l on a head-dependent (Cauchy-type) boundary. |
| IDZA | NBCZ | IDZ | Zone number for head-dependent (Cauchy-type) boundary. |
| IDSA | NBCZ | IDS | Number of head-dependent (Cauchy-type) boundary sides in each zone. |
| <u>Conjugate-gradient method</u> | | | |
| IAFA | (MBWC+1)×(NNDS+1) | AF | Element $\tilde{\alpha}_{ii}$ and $\tilde{u}_{ij}$ of upper triangular matrix U. |
| IXA | MBWC×(NNDS-NHDS) | X | Elements of $\underline{\tilde{y}}$ and $\underline{D}$ in equation (281) and $\underline{x}$ in equation (271) in Cooley (1992). |
| IPA | NNDS-NHDS | P | Elements $\underline{p}^k$ that are A-orthogonal to $\underline{p}_{k-1}$ in equation (271) in Cooley (1992). |
| IRA | NNDS-NHDS | R | Elements of $\underline{r}_i$ of flow-balance residual interations. |
| <u>Water-table (unconfined) conditions</u> | | | |
| IDHA | NNDS-NHDS | DH | Average head change [length]. |
| ITKA | NNDS | THK | Aquifer thickness [length]. |
| ITPA | NNDS | TOP | Altitude of top of aquifer [length]. |
| ISYA | NNDS | ASY | Aquifer specific yield [dimensionless]. |

<u>Nonlinear, head-dependent (Cauchy-type) flux, point sinks, and steady-vertical leakage</u>

| Starting location variable | Vector length | Variable in G | Definition |
|---|---|---|---|
| IGCA | NBNC+NPNB | GC | $\alpha$ term for nonlinear head-dependent (Cauchy-type) boundary and nonlinear point sink. |
| IHRK | NBNC | HRK | Boundary or external head $H_r$ [length] at node on nonlinear head-dependent (Cauchy-type) boundary . |
| IHRL | NBNC | HRL | Boundary or external head $H_r$ [length] node l on nonlinear head-dependent (Cauchy-type) boundary. |
| INSA | NLZA | INLS | Number of nonlinear, head-dependent (Cauchy-type) boundary sides. |
| INZA | NLZA | INLZ | Zone number for nonlinear head-dependent (Cauchy-type) boundary. |
| IZRK | NBNC | ZRK | Controlling head or altitude [length] for nonlinear head-dependent (Cauchy-type) flux at node k. |
| IZRL | NBNC | ZRL | Controlling head or altitude [length] for nonlinear head-dependent (Cauchy-type) flux at node l. |
| IKRA | NBNC | KQB | Node k on a nonlinear head-dependent (Cauchy-type) boundary. |
| ILRA | NBNC | LQB | Node l on a nonlinear, head-dependent (Cauchy-type) boundary. |
| IZPA | NPNB | HZP | Reference altitude, $z_p$, for nonlinear point sink [length]. |
| IKPA | NPNB | KP | Node number of nonlinear point sink. |

Nonlinear, head-dependent (Cauchy-type) flux, point sinks, and steady-vertical leakage (continued)

| Starting location variable | Vector length | Variable in G | Definition |
|---|---|---|---|
| IECA | NNDS | E | Nodal coefficient $\sum_{e_i}^{e} (1/3) R_e \Delta^e$ or $\sum_{e_i}^{e} (1/3) R_a \Delta,^e$ for nonlinear steady vertical leakage [length$^2$/time]. |
| IHSA | NNDS | HS | Controlling head $H_a$, or altitude, $z_e$ or $z_t$, at node [length]. |
| Transient leakage | | | |
| ICHA | MCBN | CH | Main-diagonal term $C_{hi,\,n+1}/\Delta t_{n+1}$ of equation (206) in Cooley (1992). |
| ICQA | MCBN | CBQ | Term from equation (206) in Cooley (1992) for right side of matrix equations (1) and (3). |
| IDHRA | NNDS | DHR | Change in source-layer head $H_{i,n+1}$-$H_{i,n}$. |
| ICTQA | 5×MCBN | CBTQ | Transient-leakage terms $\sum_{m=1}^{N_1} e^{-\alpha_m \gamma_i \Delta t_{n+1}} \hat{I}_{mi,n}$ (three terms), and $\sum_{m=1}^{N_1} e^{-\beta_m \gamma_i \Delta t_{n+1}} \hat{J}_{mi,n}$ (two terms) in Cooley (1992). |
| IGMA | NNDS | GMA | (1) Effective specific storage, $\sum_{e_i} S_s^e \Delta^e$, for node i, of equation (167) in Cooley (1992). |

Transient leakage (continued)

| Starting location variable | Vector length | Variable in G | Definition |
|---|---|---|---|
| IGMA | NNDS | GMA (continued) | (2) Nodal values of $\gamma_i$, given by equation (167) in Cooley (1992). |
| IALFA | 3 | ALF | Coefficients $\alpha_m$ for approximating $S_1(\Delta t_D)$ by $M_1(\Delta t_D)$ in equation (188) in Cooley (1992). |
| IACA | 3 | AC | Coefficient $A_m$ for approximating $S_1(\Delta t_D)$ by $M_1(\Delta t_D)$ in equation (188) in Cooley (1992). |
| IBTA | 2 | BTA | Coefficient $\beta_m$ for approximating $S_2(\Delta t_D)$ by $M_2(\Delta t_D)$ in equation (189) in Cooley (1992). |
| IBCA | 2 | BC | Coefficient $\beta_m$ for approximating $S_2(\Delta t_D)$ by $M_2(\Delta t_D)$ in equation (189) in Cooley (1992). |

<u>Main Program</u>

| Variable | Definition |
|---|---|
| I | Index for time-step loop. |
| IT | Counter for iteration loop. |
| ITER | Number of current iteration. |
| JP | Index for stress-period loop. |
| MCBN | Maximum number of nodes where transient leakage is simulated. |
| NBCZ | Number of zones for head-dependent (Cauchy-type) boundaries. |
| NCBZ | Number of nodes where transient leakage is simulated. |
| NZLA | Number of nonlinear head-dependent (Cauchy-type) boundary zones. |