USGS
science for a changing world

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

Chapter A4

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

By Richard L. Cooley

Book 6
Chapter A4

$$Q_{Hi,n} = \left[\frac{1}{3}\sum_{e_i} R^e \Delta^e\right] \sum_{m=1}^{N_2} e^{-\beta_m \gamma_i \Delta t_{n+1}} \hat{J}_{mi,n}, \tag{202}$$

$$C_{hi,n+1} = \left[\frac{1}{3}\sum_{e_i} R^e \Delta^e\right] \frac{M_1\left(\gamma_i \Delta t_{n+1}\right)}{\gamma_i}, \tag{203}$$

$$C_{Hi,n+1} = \left[\frac{1}{3}\sum_{e_i} R^e \Delta^e\right] \frac{M_2\left(\gamma_i \Delta t_{n+1}\right)}{\gamma_i}, \tag{204}$$

$$C_{Ri} = \frac{1}{3}\sum_{e_i} R^e \Delta^e. \tag{205}$$

Therefore, the final leakage function is

$$\begin{aligned}
&- \frac{1}{3}\left(P_{Hi,n} + 2Q_{Hi,n}\right) - \frac{2}{3}C_{Hi,n+1}\frac{H_{i,n+1} - H_{i,n}}{\Delta t_{n+1}} + \frac{1}{3}\left(P_{hi,n} + 2Q_{hi,n}\right) \\
&+ \frac{2}{3}C_{hi,n+1}\frac{\hat{h}_{i,n+1} - \hat{h}_{i,n}}{\Delta t_{n+1}} - C_{Ri}\left[\frac{1}{3}\left(H_{i,n} - \hat{h}_{i,n}\right) + \frac{2}{3}\left(H_{i,n+1} - \hat{h}_{i,n+1}\right)\right] \\
&= \left[\frac{C_{hi,n+1}}{\Delta t_{n+1}} + C_{Ri}\right]\delta_i - \frac{1}{3}\left(P_{Hi,n} + 2Q_{Hi,n}\right) - \frac{2}{3}C_{Hi,n+1}\frac{H_{i,n+1} - H_{i,n}}{\Delta t_{n+1}} \\
&+ \frac{1}{3}\left(P_{hi,n} + 2Q_{hi,n}\right) - \frac{1}{3}C_{Ri}\left[H_{i,n} - \hat{h}_{i,n} + 2\left(H_{i,n+1} - \hat{h}_{i,n}\right)\right]
\end{aligned} \tag{206}$$

The leakage process described by equation (206) is time dependent, but linear. Therefore, equation (206) is added into equation (58), unless the predictor-corrector method is required to include other phenomena, in which case equation (206) is added into the appropriate predictor and corrector equations. The coefficient of δ_i is added into $\underline{V}$ for every node i where leakage occurs, and the remaining terms are subtracted from the right-hand side.

FINITE-ELEMENT FORMULATION IN AXISYMMETRIC CYLINDRICAL COORDINATES

GOVERNING FLOW EQUATION AND BOUNDARY CONDITIONS

Axially symmetric ground-water flow in an aquifer is assumed to be governed by the following unsteady-state flow equation written in axisymmetric cylindrical coordinates (Bear, 1979, p. 116):

$$\frac{1}{r}\frac{1}{\partial r}\left(K_{rr} r \frac{\partial h}{\partial r}\right) + \frac{\partial}{\partial z}\left(K_{zz}\frac{\partial h}{\partial z}\right) - S_s \frac{\partial h}{\partial t}, \tag{207}$$

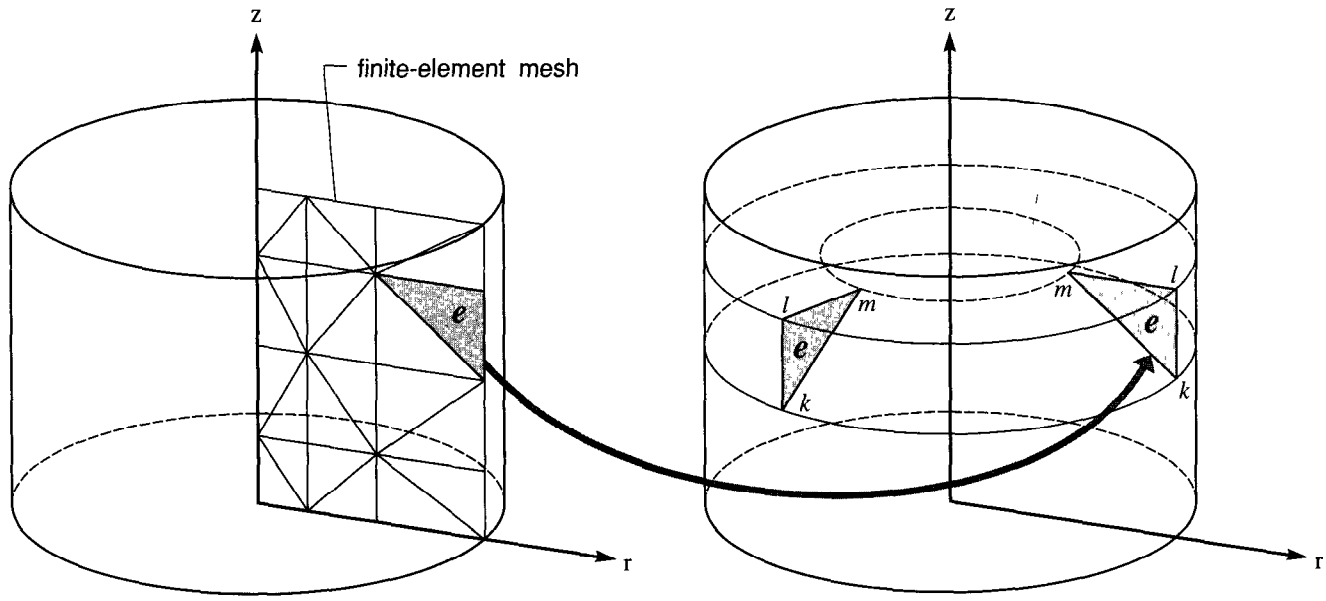

Figure 17. Axisymmetric aquifer subdivided into spatial finite elements.

where the new symbols used are

r = radial (horizontal) coordinate direction [length] from the axis of symmetry, which is vertical,
z = vertical coordinate direction [length],
$K_{rr}(r,z), K_{zz}(r,z)$ = the principal components of the hydraulic conductivity tensor [length/time] in the radial and vertical coordinate directions, respectively, and
$S_s(r,z)$ = specific storage [length^{-1}].

The orientation of the r and z axes is shown in figure 17.

The principal directions of the hydraulic conductivity tensor are assumed to be parallel to the r and z axes in equation (207). Equation (207) was not written in full component form like equation (1), because any rotation of the principal directions from the r and z axes (see figure 5) must be revolved around the z axis to maintain axial symmetry. This produces the physically unusual case of an axially symmetric rotation of the principal directions, which seemed to the author to be an unnecessary complication to include.

Equation (207) is subject to boundary and initial conditions analogous to equations (2) through (5) used for equation (1). However, equations (3) and (4) must be changed to reflect the change from flow integrated over aquifer thickness in equation (1) to flow in a radial cross-section in equation (207). Thus, equation (3), which expresses flow continuity across a discontinuity in the porous medium, is replaced by

$$v_n\big|_a = v_n\big|_b, \tag{208}$$

where $v_n(r,z,t)$ is the normal component of specific discharge, and equation (4), which expresses either a specified-head or Cauchy-type boundary condition, is replaced by

$$v_n = v_B + \alpha'(H_B - h), \tag{209}$$

where

$v_B(r,z,t)$ = specified specific discharge normal to the boundary [length/time] (positive for inflow), and

$\alpha'(r,z,t)$ = a parameter that, like α of equation (4), approaches infinity for a specified-head (Dirichlet) condition, is zero for a specified flow (Neumann) condition, and is finite and positive for a general or mixed (Cauchy) condition [$time^{-1}$].

FINITE-ELEMENT DISCRETIZATION

Finite-element discretization in axisymmetric cylindrical coordinates is accomplished in the same manner as for the Cartesian case. An r-z plane is subdivided into triangular elements (figure 17) over each of which the approximate solution h is assumed to vary linearly. Because of axial symmetry, each element is revolved around the symmetry axis so that it is a ringlike volume with triangular cross section. The time domain is subdivided into time elements over each of which the solution is also assumed to vary linearly. Therefore, the approximate solution is stated as an equation that is analogous to equation (14):

$$\hat{h} = \sum_i \left(\hat{h}_{i,n} \sigma_n + \hat{h}_{i,n+1} \sigma_{n+1} \right) N_i^e, \quad i = k,l,m, \tag{210}$$

where

$$N_i^e = \left(a_i^e + b_i^e r + c_i^e z \right) / 2\Delta^e, \quad i = k,l,m, \tag{211}$$

$$\begin{aligned}
a_k^e &= r_l z_m - r_m z_l, \\
b_k^e &= z_l - z_m, \\
c_k^e &= r_m - r_l, \\
a_l^e &= r_m z_k - r_k z_m, \\
b_l^e &= z_m - z_k, \\
c_l^e &= r_k - r_m, \\
a_m^e &= r_k z_l - r_l z_k, \\
b_m^e &= z_k - a_l, \\
c_m^e &= r_l - r_k,
\end{aligned} \tag{212}$$

$$2\Delta^e = \left(r_k - r_m \right) \left(z_l - z_m \right) - \left(r_m - r_l \right) \left(z_m - z_k \right), \tag{213}$$

and σ_n and σ_{n+1} are defined by equation (13).

DERIVATION OF FINITE-ELEMENT EQUATIONS

The error-functional equation in axisymmetric cylindrical coordinates is analogous to equation (15). It is written for an r-z plane as

$$I(\hat{e}) = \sum_e \int_0^{\Delta t_{n+1}} \left\{ \iint_{\Delta^e} \left[\frac{\partial \hat{e}}{\partial r} K_{rr} \frac{\partial \hat{e}}{\partial r} + \frac{\partial \hat{e}}{\partial z} K_{zz} \frac{\partial \hat{e}}{\partial z} + S_s \left(\frac{\partial \hat{e}}{\partial t} \right)^2 t' \right] r dr dz + \int_{C_2^e} \alpha' \hat{e}^2 r dC \right\} dt'. \quad (214)$$

Minimization of equation (214) with respect to $\hat{h}_{i,n+1}$ and separation of the result into two parts as for equation (17) gives

$$\sum_{e_i} \int_0^{\Delta t_{n+1}} \sigma_{n+1} \left\{ \iint_{\Delta^e} \left[N_i^e S_s \frac{\partial \hat{h}}{\partial t} + \frac{\partial N_i^e}{\partial r} K_{rr} \frac{\partial \hat{h}}{\partial r} + \frac{\partial N_i^e}{\partial z} K_{zz} \frac{\partial \hat{h}}{\partial z} \right] r dr dz - \int_{C_2^e} N_i^e \left[v_B + \alpha' \left(H_B - \hat{h} \right) \right] r dC \right\} dt' - \sum_{e_i} \int_0^{\Delta t_{n+1}} \sigma_{n+1} \left\{ \iint_{\Delta^e} \left[N_i^e S_s \frac{\partial h}{\partial t} + \frac{\partial N_i^e}{\partial r} K_{rr} \frac{\partial h}{\partial r} + \frac{\partial N_i^e}{\partial z} K_{zz} \frac{\partial h}{\partial z} \right] r dr dz - \int_{C_2^e} N_i^e \left[v_B + \alpha' \left(H_B - h \right) \right] r dC \right\} dt' = 0. \quad (215)$$

By following the procedures used in appendix A, the second part of equation (215) can be shown to equal zero. Therefore, when the integrals involving $S_s \partial \hat{h}/\partial t$ and $\alpha'(H_B - \hat{h})$ are converted to diagonal form using approximations analogous to equation (19), the following integral form of the finite-element equations results:

$$\sum_{e_i} \int_0^{\Delta t_{n+1}} \sigma_{n+1} \left\{ \iint_{\Delta^e} \left[N_i^e S_s \frac{d\hat{h}_i}{dt} + \frac{\partial N_i^e}{\partial r} K_{rr} \frac{\partial \hat{h}}{\partial r} + \frac{\partial N_i^e}{\partial z} K_{zz} \frac{\partial \hat{h}}{\partial z} \right] r dr dz - \int_{C_2^e} N_i^e \left[v_B + \alpha' \left(H_{Bi} - \hat{h}_i \right) \right] r dC \right\} dt' = 0, \; i = 1,2,\ldots,N. \quad (216)$$

Equation (216) is analogous to equation (21). However, because the principal axes of the hydraulic conductivity tensor were originally assumed to be parallel to the r and z coordinate axes, no coordinate rotations are performed.

Equation (216) is integrated to obtain the final finite-element equations. As before, the spatial integrations are performed first. It is assumed that S_s, K_{rr}, and K_{zz} are all constant in each spatial element, and that v_B and α' are constant along any Cauchy-type boundary side of each element. The extra r in the integrals presents a complication not present for the Cartesian case. However, by writing r as the identity

$$r = N_k^e r_k + N_l^e r_l + N_m^e r_m, \tag{217}$$

equations (24) and (25) can be used to perform the integrations. Therefore, by substituting the appropriate expressions for $\hat{h}$, N_i^e, $\partial N_i^e/\partial r$, and $\partial N_i^e/\partial z$, $i = k,l,m$, the spatial integrals in equation (216) are evaluated for $i = k$ (for example) as

$$\int_{\Delta^e}\int N_k^e S_s \frac{d\hat{h}_k}{dt} r\,dr\,dz = \frac{1}{12} S_s^e \left(2r_k + r_l + r_m\right)\Delta^e \frac{d\hat{h}_k}{dt}, \tag{218}$$

$$\int_{\Delta^e}\int \frac{\partial N_k^e}{\partial r} K_{rr} \frac{\partial \hat{h}}{\partial r} r\,dr\,dz = \int_{\Delta^e}\int \frac{\partial N_k^e}{\partial r} K_{rr}^e \left(\frac{\partial N_k^e}{\partial r}\hat{h}_k + \frac{\partial N_l^e}{\partial r}\hat{h}_l + \frac{\partial N_m^e}{\partial r}\hat{h}_m\right) r\,dr\,dz$$

$$= \frac{K_{rr}^e}{4\Delta^e}\left(b_k^e b_k^e \hat{h}_k + b_k^e b_l^e \hat{h}_l + b_k^e b_m^e \hat{h}_m\right)\bar{r}, \tag{219}$$

$$\int_{\Delta^e}\int \frac{\partial N_k^e}{\partial z} K_{zz} \frac{\partial \hat{h}}{\partial z} r\,dr\,dz = \int_{\Delta^e}\int \frac{\partial N_k^e}{\partial z} K_{zz}^e \left(\frac{\partial N_k^e}{\partial z}\hat{h}_k + \frac{\partial N_l^e}{\partial z}\hat{h}_l + \frac{\partial N_m^e}{\partial z}\hat{h}_m\right) r\,dr\,dz$$

$$= \frac{K_{zz}^e}{4\Delta^e}\left(c_k^e c_k^e \hat{h}_k + c_k^e c_l^e \hat{h}_l + c_k^e c_m^e \hat{h}_m\right)\bar{r}, \tag{220}$$

$$\int_{C_2^e} N_k^e\left[v_B + \alpha'\left(H_{Bk} - \hat{h}_k\right)\right] dC = \frac{1}{6}\left(2r_k + r_l\right)\left[\left(v_B L\right)_{kl} + \left(\alpha' L\right)_{kl}\left(H_{Bk} - \hat{h}_k\right)\right]$$

$$+ \frac{1}{6}\left(2r_k + r_m\right)\left[\left(v_B L\right)_{km} + \left(\alpha' L\right)_{km}\left(H_{Bk} - \hat{h}_k\right)\right], \tag{221}$$

where S_s^e, K_{rr}^e, and K_{zz}^e are the constant values of S_s, K_{rr}, and K_{zz} in element e,

$$\bar{r} = \frac{1}{3}\left(r_k + r_l + r_m\right), \tag{222}$$

and other terms were defined earlier.

The spatially integrated finite-element equation for node k is obtained by substituting equations (218) through (221) into equation (216) and using equations analogous to equations (41) and (42). The result is

$$\sum_{e_i}\int_0^{\Delta t_{n+1}} \sigma_{n+1}\left\{c^e_{kk}\frac{d\hat{h}_k}{dt} + \left(g^e_{kk} + v^e_{kk}\right)\hat{h}_k + g^e_{kl}\hat{h}_l + g^e_{km}\hat{h}_m\right.$$

$$- \frac{1}{6}\left[\left(2r_k + r_l\right)\left(v_B L\right)_{kl} + \left(2r_k + r_m\right)\left(v_B L\right)_{km}\right] - \frac{1}{6}\left[\left(2r_k + r_l\right)\left(\alpha' L\right)_{kl}\right.$$

$$\left.\left. + \left(2r_k + r_m\right)\left(\alpha' L\right)_{km}\right]H_{Bk}\right\}dt' = 0, \tag{223}$$

where

$$c^e_{kk} = \frac{1}{12}S^e_s\left(2r_k + r_l + r_m\right)\Delta^e, \tag{224}$$

$$v^e_{kk} = \frac{1}{6}\left[\left(2r_k + r_l\right)\left(\alpha' L\right)_{kl} + \left(2r_k + r_m\right)\left(\alpha' L\right)_{km}\right], \tag{225}$$

$$g^e_{kk} = -g^e_{kl} - g^e_{km}, \tag{226}$$

$$g^e_{kl} = \left[\frac{K^e_{rr}}{4\Delta^e}b_k b_l + \frac{K^e_{zz}}{4\Delta^e}c_k c_l\right]\bar{r}, \tag{227}$$

$$g^e_{km} = \left[\frac{K^e_{rr}}{4\Delta^e}b_k b_m + \frac{K^e_{zz}}{4\Delta^e}c_k c_m\right]\bar{r}, \tag{228}$$

As before, equation (223) must apply to all N nodes of the finite-element mesh. These N equations are written in matrix form as equation (45), where C_{ij}, V_{ij}, and G_{ij} are given by equations (47) through (49), and

$$B_i = \sum_{e_i}\left\{\frac{1}{6}\sum_{j'}\left(2r_i + r_{j'}\right)\left[\left(v_B L\right)_{ij'} + \left(\alpha' L\right)_{ij'}H_{Bi}\right]\right\}. \tag{229}$$

Specified-head boundaries are handled using equation (51).

To perform the final time integration, parameters S_s, K_{rr}, K_{zz}, and α' are all assumed to be constant in time, and specified boundary flux v_B is assumed to be linearly variable through each time element. Thus, the time integrated finite-element equations for axisymmetric flow are given by equation (58) with $\underline{B}$ replaced by $\bar{\underline{B}}$ defined by equation (62). No nonlinear or other extensions are used.

FINITE-ELEMENT FORMULATION FOR STEADY-STATE FLOW

By definition, steady-state flow occurs when hydraulic heads do not vary with time, or $\partial h/\partial t = 0$ in equation (1) or (207). Steady-state flow equations are obtained by setting S or S_s to zero and letting all quantities in equations (1) through (4) or (207) through (209) be time invariant.

LINEAR CASE

The finite-element equations for steady-state confined flow in the absence of any of the nonlinear source-sink functions may be derived from equation (56) by setting $\underline{\underline{C}}$ (which contains S or S_s) to zero and setting $\hat{\underline{h}}_{n+1} = \hat{\underline{h}}_n = \hat{\underline{h}}$. The resulting equation is

$$\underline{\underline{A}}\hat{\underline{h}} = \underline{B}, \tag{230}$$

where $\underline{\underline{A}} = \underline{\underline{G}} + \underline{\underline{V}}$. As for unsteady flow, round-off error may be minimized by solving for head change rather than head. Thus, by defining this head change, $\underline{\delta}_o$, as

$$\underline{\delta}_o = \hat{\underline{h}} - \underline{h}_o, \tag{231}$$

where $\underline{h}_o$ is an arbitrary initial set of heads close to $\hat{\underline{h}}$, equation (230) is modified to become

$$\underline{\underline{A}}\underline{\delta}_o = \underline{B} - \underline{\underline{A}}\underline{h}_o. \tag{232}$$

To solve a linear, steady-state flow problem, first equation (232) is solved for $\underline{\delta}_o$, then equation (231) is solved for $\hat{\underline{h}}$. Mass balance components are obtained from equation (232) using analogous procedures to those used for unsteady-state flow.

NONLINEAR CASE

If steady-state flow is unconfined or a nonlinear source-sink function is employed, then $\underline{\underline{A}}$ and $\underline{B}$ are functions of $\hat{\underline{h}}$ and a nonlinear problem results. In the case of unconfined flow, an off-diagonal entry of $\underline{\underline{G}}$ is given by equation (74) in which $b_i = \hat{h}_i - z_{bi}$. The particular form of the nonlinear source-sink term incorporated into $\underline{\underline{V}}$ and $\underline{B}$ is dependent on the type of function: point head-dependent discharge, areal head-dependent leakage, areal head-dependent discharge, or line head-dependent leakage.

The general algorithm used to solve the nonlinear problem is derived before stating the specific terms used for unconfined flow and nonlinear sources and sinks. For nonlinear problems, equation (230) is restated as

$$\underline{\underline{A}}\left(\hat{\underline{h}}\right)\hat{\underline{h}} = \underline{B}\left(\hat{\underline{h}}\right), \tag{233}$$

where the notation $\underline{\underline{A}}\left(\hat{\underline{h}}\right)$ and $\underline{B}\left(\hat{\underline{h}}\right)$ signifies that $\underline{\underline{A}}$ and $\underline{B}$ are functions of $\hat{\underline{h}}$. An iterative solution method for equation (233) may be derived by adding and

subtracting $\underline{\underline{A}}\left(\hat{\underline{h}}\right)$ to the right-hand side, then restating the result as an iteration equation of the form

$$\underline{\underline{A}}\left(\hat{\underline{h}}_\ell\right)\hat{\underline{h}}_{\ell+1} = \underline{\underline{A}}\left(\hat{\underline{h}}_\ell\right)\hat{\underline{h}}_\ell + \underline{B}\left(\hat{\underline{h}}_\ell\right) - \underline{\underline{A}}\left(\hat{\underline{h}}_\ell\right)\hat{\underline{h}}_\ell,$$

or

$$\underline{\underline{A}}_\ell \underline{\delta}_\ell = \underline{r}_\ell \tag{234}$$

where ℓ is the iteration index, and

$$\underline{\underline{A}}_\ell = \underline{\underline{A}}\left(\hat{\underline{h}}_\ell\right), \tag{235}$$

$$\underline{B}_\ell = \underline{B}\left(\hat{\underline{h}}_\ell\right), \tag{236}$$

$$\underline{\delta}_\ell = \hat{\underline{h}}_{\ell+1} - \hat{\underline{h}}_\ell, \tag{237}$$

$$\underline{r}_\ell = \underline{B}_\ell - \underline{\underline{A}}_\ell \hat{\underline{h}}_\ell. \tag{238}$$

Head-change vector $\underline{\delta}_\ell$ in equation (234) frequently requires damping to reduce undesirable oscillations from one iteration to the next. Addition of a damping parameter ρ_ℓ $(0 < \rho_\ell \leq 1)$ to equation (237) yields the following iteration algorithm:

$$\left.\begin{aligned} \underline{r}_\ell &= \underline{B}_\ell - \underline{\underline{A}}_\ell \hat{\underline{h}}_\ell \\ \underline{\delta}_\ell &= \underline{\underline{A}}_\ell^{-1} \underline{r}_\ell \\ \hat{\underline{h}}_{\ell+1} &= \rho_\ell \underline{\delta}_\ell + \hat{\underline{h}}_\ell \end{aligned}\right\} \ell = 0,1,\ldots \tag{239}$$

The iterations are terminated when

$$\max_i \left|\delta_i^\ell\right| \leq \epsilon_s, \tag{240}$$

where δ_i^ℓ is a component of $\underline{\delta}_\ell$ and ϵ_s is a small number about an order of magnitude smaller than the desired accuracy in $\hat{\underline{h}}$.

An effective empirical scheme for computing ρ_ℓ was developed by Cooley (1983, p. 1274). It is given in three steps. Let e_ℓ be the value of δ_i^ℓ that is largest in absolute value for all $i = 1,2,\ldots,N$, and let e_{max} be the largest value of $\left|e_\ell\right|$ permitted on any iteration. Then,

Step 1

$$\begin{aligned} p &= \frac{e_\ell}{\rho_{\ell-1} e_{\ell-1}}, \quad \ell > 1 \\ p &= 1 \qquad\quad , \quad \ell = 1 \end{aligned} \tag{241}$$

Step 2

$$\begin{aligned} \rho^* &= \frac{3 + p}{3 + |p|}, \quad p \geq -1 \\ \rho^* &= \frac{1}{2|p|} \quad , \quad p < -1 \end{aligned} \tag{242}$$

Step 3

$$\rho_\ell = \rho^*, \ \rho^*\left|e_\ell\right| \le e_{max}$$
$$\rho_\ell = \frac{e_{max}}{\left|e_\ell\right|}, \ \rho^*\left|e_\ell\right| > e_{max} \tag{243}$$

A good trial value for e_{max} is about half the maximum value of $\left|\underline{h}_{oi} - \hat{\underline{h}}_i\right|$ expected (where $\underline{h}_{oi}$ is a component of the initial head vector). Much smaller values may be needed for highly nonlinear problems.

At the beginning of each iteration ℓ, $\underline{\underline{A}}_\ell$ and $\underline{B}_\ell$ must be recomputed using the newly computed values of $\hat{\underline{h}}_\ell$. The way in which $\underline{\underline{A}}_\ell$ and $\underline{B}_\ell$ are recomputed depends on the source of the nonlinearity. Nonlinearity from unconfined flow results when $\hat{h}_i < z_{ti}$. To allow for the possibility of unconfined flow, off-diagonals of $\underline{\underline{G}}$ are computed from equation (74) written for iteration ℓ as

$$G_{ij}^\ell = \frac{1}{2}\left(b_i^\ell + b_j^\ell\right)D_{ij}, \tag{244}$$

where

$$b_i^\ell = b_i^{\ell-1} + \rho_\ell \delta_i^{\ell-1}, \ \hat{h}_i^\ell < z_{ti}$$
$$b_i^\ell = z_{ti} - z_{bi}, \ \hat{h}_i^\ell \ge z_{ti} \tag{245}$$

Nodes that go dry are treated in the same manner as explained for unsteady-state flow. The head is allowed to decline below the base of the aquifer at a dry node i, but horizontal flow in the aquifer is allowed between adjacent nodes i and j unless node j also goes dry. If a dry node i becomes surrounded by dry nodes during the iterative solution process, then $G_{ii} = 0$ and flow can only take place vertically through an underlying confining unit or across a Cauchy-type boundary at the dry node. If there is no confining unit or Cauchy-type boundary so that V_{ii} is also zero, then A_{ii}, which is $G_{ii} + V_{ii}$, is zero so that the head at node i is undefined and must be removed from the solution. This is accomplished by setting A_{ii} to 10^{30} and setting the right-hand side of the equation to zero, which holds the head constant at the last computed value. If the sum of the known fluxes is negative at a dry node, this sum is reduced by 1/2 at each iteration until the node remains saturated. As discussed earlier, this tells the user the approximate discharge that can be sustained at the node.

Nonlinear source-sink functions require reevaluation of $\underline{\underline{V}}_\ell$ and $\underline{B}_\ell$. For point head-dependent discharge functions, reevaluation is based on equation (106) written for iteration ℓ as

$$Q_{pi}^\ell = \begin{cases} C_{pi}\left(z_{pi} - \hat{h}_i^{\ell+1}\right), & \hat{h}_i^\ell > z_{pi} \\ 0, & \hat{h}_i^\ell \le z_{pi} \end{cases}. \tag{246}$$

The terms that add into equation (234) are found by converting Q_{pi} to residual form using equation (237). That is, if $\hat{h}_i^{\ell} > z_{pi}$, then

$$Q_{pi}^{\ell} = C_{pi}\left(z_{pi} - \hat{h}_i^{\ell+1}\right) = - C_{pi}\delta_i^{\ell} + C_{pi}\left(z_{pi} - \hat{h}_i^{\ell}\right) \tag{247}$$

so that C_{pi} is added into V_{ii}^{ℓ} and $C_{pi}\left(z_{pi} - \hat{h}_i^{\ell}\right)$ is added to the right-hand side. If $\hat{h}_i^{\ell} \le z_{pi}$, then no terms are added.

Use of the other nonlinear source-sink functions is analogous. For areal head-dependent leakage, equation (118) for iteration ℓ is

$$Q_{ai}^{\ell} = \begin{cases} C_{ai}\left(H_{ai} - \hat{h}_i^{\ell+1}\right), & \hat{h}_i^{\ell} > z_{ti} \\ C_{ai}\left(H_{ai} - z_{ti}\right), & \hat{h}_i^{\ell} \le z_{ti} \end{cases} \tag{248}$$

so that, when $\hat{h}_i^{\ell} > z_{ti}$,

$$Q_{ai}^{\ell} = C_{ai}\left(H_{ai} - \hat{h}_i^{\ell+1}\right) = - C_{ai}\delta_i^{\ell} + C_{ai}\left(H_{ai} - \hat{h}_i^{\ell}\right). \tag{249}$$

Therefore, when $\hat{h}_i^{\ell} > z_{ti}$, equation (234) is modified by adding C_{ai} into V_{ii}^{ℓ} and adding $C_{ai}\left(H_{ai} - \hat{h}_i^{\ell}\right)$ to the right-hand side. When $\hat{h}_i^{\ell} \le z_{ti}$, $C_{ai}\left(H_{ai} - z_{ti}\right)$ is added to the right-hand side and V_{ii}^{ℓ} is not modified.

Likewise, for areal head-dependent discharge functions, equation (131) is written for iteration ℓ as

$$Q_{ei}^{\ell} = \begin{cases} C_{ei}\left(z_{ei} - z_{ti}\right), & \hat{h}_i^{\ell} \ge z_{ti} \\ C_{ei}\left(z_{ei} - \hat{h}_i^{\ell+1}\right), & z_{ei} < \hat{h}_i^{\ell} < z_{ti} \\ 0, & \hat{h}_i^{\ell} \le z_{ei} \end{cases} \tag{250}$$

so that, when $z_{ei} < \hat{h}_i^{\ell} < z_{ti}$,

$$Q_{ei}^{\ell} = C_{ei}\left(z_{ei} - \hat{h}_i^{\ell+1}\right) = - C_{ei}\delta_i^{\ell} + C_{ei}\left(z_{ei} - \hat{h}_i^{\ell}\right). \tag{251}$$

Substitutions into equation (234) are analogous to the previous case. Finally, for line head-dependent leakage functions, equation (154) is written for iteration ℓ as

$$Q_{ri}^{\ell} = \begin{cases} C_{ri}\left(H_{ri} - \hat{h}_i^{\ell+1}\right), & \hat{h}_i^{\ell} > z_{ri} \\ C_{ri}\left(H_{ri} - z_{ri}\right), & \hat{h}_i^{\ell} \le z_{ri} \end{cases} \tag{252}$$

so that, when $\hat{h}_i^{\ell} > z_{ri}$,

$$Q_{ri}^{\ell} = C_{ri}\left(H_{ri} - \hat{h}_i^{\ell+1}\right) = - C_{ri}\delta_i^{\ell} + C_{ri}\left(H_{ri} - \hat{h}_i^{\ell}\right). \tag{253}$$

SOLUTION OF MATRIX EQUATIONS

Some of the symbols used in previous sections are redefined in this section to avoid complex or nonstandard matrix-solution terminology. Thus, symbols defined in this section are for use in this section only.

DEFINITION OF MATRIX EQUATION

Equation (58) must be solved for each time level of a linear, unsteady-state flow problem, and equations (76) and (80) (the predictor-corrector equations) must be solved sequentially at each time level of a nonlinear, unsteady-state flow problem. Likewise, equation (232) must be solved for a linear, steady-state flow problem, and equation (234) must be solved for each iteration of a nonlinear, steady-state flow problem. All of these equations are linear and of the form

$$\underline{\underline{A}}\,\underline{x} = \underline{d}, \tag{254}$$

where definitions of the coefficient matrix $\underline{\underline{A}}$, the known vector $\underline{d}$, and the unknown vector $\underline{x}$ depend on the equation being solved. For example, for equation (58),

$$\underline{\underline{A}} = \frac{\underline{\underline{C}}}{(2/3)\Delta t_{n+1}} + \underline{\underline{G}} + \underline{\underline{V}}, \tag{255}$$

$$\underline{x} = \underline{\delta}, \tag{256}$$

$$\underline{d} = \underline{B} - \left(\underline{\underline{G}} + \underline{\underline{V}}\right)\hat{\underline{h}}_n. \tag{257}$$

Thus, $\underline{\underline{A}}$ is an N × N matrix, and $\underline{x}$ and $\underline{d}$ are N-vectors.

The location of nonzero entries in matrix $\underline{\underline{A}}$ depends on the finite-element mesh. Each row i of $\underline{\underline{A}}$ contains nonzero entries only corresponding to nodes in the patch of elements for node i. Therefore, unless N is very small, $\underline{\underline{A}}$ is sparse in that most entries in any row are zero. Also, if the nodes are numbered so that the difference between the largest and smallest node numbers in the patch is small compared to N, then $\underline{\underline{A}}$ is banded, which means that all nonzero entries in each row are clustered near the main diagonal. Because $\underline{\underline{A}}$ is derived from the positive definite forms in equation (15) or (214), it is symmetric and positive definite. Finally, as discussed previously, if all internal angles of the spatial elements are acute, $\underline{\underline{A}}$ is a Stieltjes matrix. Additional information on finite-element matrices can be found in Desai and Abel (1972, chap. 2).

If node i is a specified-head node, equation i of equation (254) is $x_i = \frac{2}{3}\left(H_{Bi,n+1} - \hat{h}_{i,n}\right)$ for unsteady-state flow and $x_i = H_{Bi} - \hat{h}_i$ for steady-state flow. Because x_i is known at all specified-head nodes, all of the corresponding equations may be eliminated from equation (254). This may be accomplished as a partitioning operation by numbering all specified-head nodes in the finite-element mesh last, which is accomplished automatically in the code. Terms in the remaining equations that contain values of x_i for the specified-head nodes are then transferred to the right-hand sides of these equations to become part of the known vector. In the remainder of this section, equation (254) is regarded as the reduced equation resulting from this partitioning operation.

SYMMETRIC-DOOLITTLE METHOD

The first of two alternative matrix-solution procedures is discussed in this section. This method is referred to as symmetric-Doolittle decomposition (Fox, 1965, p. 99-102, 104-105) and is generally the preferred direct solution method for finite-element equations (Desai and Abel, 1972, p. 21). It is a direct method because the solution is found directly in three steps as opposed to iteratively in an unspecified number of steps required by the second method. Direct solution is usually efficient whenever there are fewer than about 500 nodes (Gambolati and Volpi, 1982).

The symmetric-Doolittle method is based on the fact that the symmetric matrix $\underline{\underline{A}}$ can be uniquely factored into the product of three matrices (Fox, 1965, p. 105), so that

$$\underline{\underline{A}} = \underline{\underline{U}}^T \underline{\underline{D}} \underline{\underline{U}}, \tag{258}$$

where superscript T stands for transpose, $\underline{\underline{U}}$ is upper triangular of the form

$$\underline{\underline{U}} = \begin{bmatrix} \alpha_{11} & u_{12} & u_{13} & \cdots & u_{1N} \\ 0 & \alpha_{22} & u_{23} & \cdots & u_{2N} \\ 0 & 0 & \alpha_{33} & \cdots & u_{3N} \\ & & \cdots & & \\ 0 & 0 & 0 & \cdots & \alpha_{NN} \end{bmatrix}, \tag{259}$$

and $\underline{\underline{D}}$ is diagonal of the form

$$\underline{\underline{D}} = \begin{bmatrix} 1/\alpha_{11} & 0 & 0 & \cdots & 0 \\ 0 & 1/\alpha_{22} & 0 & \cdots & 0 \\ 0 & 0 & 1/\alpha_{33} & \cdots & 0 \\ & & \cdots & & \\ 0 & 0 & 0 & \cdots & 1/\alpha_{NN} \end{bmatrix}. \tag{260}$$

Factorization, which is the first step of the three-step solution, is accomplished by forming the product matrix $\underline{\underline{U}}^T \underline{\underline{D}} \underline{\underline{U}}$, setting each entry of this matrix equal to the corresponding entry of $\underline{\underline{A}}$, then solving for the unknown values of u_{ij} and α_{ii}, entry by entry.

Solution of equation (254) using the factorization given by equation (258) is accomplished as follows. By defining a vector $\underline{y}$ by

$$\underline{\underline{U}}\underline{x} = \underline{y}, \tag{261}$$

the combination of equations (254) and (258) can be written as

$$\underline{\underline{U}}^T \underline{\underline{D}} \underline{y} = \underline{d}. \tag{262}$$

The lower triangular form of $\underline{\underline{U}}^T \underline{\underline{D}}$ and the upper triangular form of $\underline{\underline{U}}$ permit equations (262) and (261) to easily be solved for $\underline{y}$ and $\underline{x}$, respectively, as the second and third steps of the solution procedure. By forming the product $\underline{\underline{U}}^T \underline{\underline{D}} \underline{y}$, it can be seen that the first equation in equation (262) contains only y_1 as an unknown, the second y_1 and y_2, and so forth, so that

the first equation may be solved for y_1, which is used in the second to solve for y_2, and so forth. Solution vector $\underline{x}$ is found in exactly the opposite way. The last equation in equation (261) contains only the last unknown, x_N, the second from the last x_N and x_{N-1}, and so forth, so that the last equation is solved for x_N, which is used in the second from the last to solve for x_{N-1}, and so forth.

For N equations with N unknown values in $\underline{x}$, the calculations may be stated in algorithmic form as

$$\left.\begin{aligned} \alpha_{ii} &= a_{ii} - \sum_{k=1}^{i-1} u_{ki}u_{ki}/\alpha_{kk} \\ u_{ij} &= a_{ij} - \sum_{k=1}^{i-1} u_{ki}u_{kj}/\alpha_{kk} \\ u'_{ij} &= u_{ij}/\alpha_{ii} \end{aligned}\right\} \begin{array}{l} i = 1,2,\ldots,N \\ j = i+1,i+2,\ldots,N, \end{array} \qquad (263)$$

$$\left.\begin{aligned} y_i &= d_i - \sum_{k=1}^{i-1} u'_{ki}y_k \\ y'_i &= y_i/\alpha_{ii} \end{aligned}\right\} \quad i = 1,2,\ldots,N, \qquad (264)$$

$$x_i = y'_i - \sum_{k=i+1}^{N} u'_{ik}x_k, \qquad i = N,N-1,\ldots,1. \qquad (265)$$

Equation (263) is known as the factorization step, equation (264) is the forward substitution step, and equation (265) is the backward substitution step.

When the above algorithm is applied to the banded matrix $\underline{\underline{A}}$, entries in $\underline{\underline{U}}$ outside of the band are always found to be zero. However, $\underline{\underline{U}}$ has mostly nonzero entries within the band, even if the corresponding entries in $\underline{\underline{A}}$ are mostly zeros. Therefore, the algorithm can be coded to operate on and store only entries within the band. Storage of $\underline{\underline{A}}$ for efficient computer application of the solution algorithm is explained in part 3.

As with any direct solution method, the above method can generate inaccurate solutions for poorly conditioned equation systems, such as can occur when $\underline{\underline{A}}$ is not diagonally dominant, or has weak diagonal dominance, and(or) has highly variable entries. Matrix $\underline{\underline{A}}$ can have weak diagonal dominance if all internal angles in spatial elements are acute but R, S, and α in equations (1) and (4) (or S_s and α' in equations (207) and (209)) are zero and there are few specified-head nodes. Matrix $\underline{\underline{A}}$ may not be diagonally dominant if R, S, and α are zero, and one or more elements has an obtuse internal angle. Entries in $\underline{\underline{A}}$ can be highly variable if values of transmissivity or element shapes are highly variable. An inaccurate solution generally results in a large mass imbalance.

MODIFIED INCOMPLETE-CHOLESKY CONJUGATE-GRADIENT METHOD

If N is large or the direct solution method produces large mass balance errors, then an iterative method should be used. An iterative solution method called the generalized conjugate-gradient method (GCGM) has been found by Gambolati and Volpi (1982) to be more efficient than the direct method for solving sets of finite-element equations when $N \geq 500$. The iterative method used here is a combination of a variant of GCGM by Manteuffel (1980) with a preconditioning method by Wong (1979) designed to enhance the convergence rate.

Generalized conjugate-gradient method

The iteration equation for the GCGM method is derived by replacing $\underline{\underline{A}}$ with a coefficient matrix $\underline{\underline{M}}$ that is similar to $\underline{\underline{A}}$ but much easier to invert (Concus and others, 1975). Matrix $\underline{\underline{M}}$, known as a preconditioning matrix, is defined from the fact that $\underline{\underline{A}}$ can always be split into the sum of two matrices, $\underline{\underline{M}}$ and $\underline{\underline{N}}$ (Varga, 1962, p. 87-93), so that

$$\underline{\underline{A}} = \underline{\underline{M}} + \underline{\underline{N}}. \tag{266}$$

Therefore, because the combination of equations (254) and (266) gives

$$\underline{\underline{M}}\underline{x} = \underline{d} - \underline{\underline{N}}\underline{x}, \tag{267}$$

the iteration equation is

$$\underline{\underline{M}}\underline{x}_{k+1} = \underline{d} - \underline{\underline{N}}\underline{x}_k \tag{268}$$

or, written in residual form,

$$\underline{\underline{M}}\underline{s}_{k+1} = \underline{r}_k, \tag{269}$$

where

$$\begin{aligned} \underline{s}_{k+1} &= \underline{x}_{k+1} - \underline{x}_k, \\ \underline{r}_k &= \underline{d} - \underline{\underline{A}}\underline{x}_k. \end{aligned} \tag{270}$$

The GCGM algorithm based on iteration equation (269) can be stated as (Concus and others, 1975, p. 7-8)

$$\left.\begin{aligned} \underline{s}_k &= \underline{\underline{M}}^{-1}\underline{r}_k \\ \underline{p}_k &= \underline{s}_k \end{aligned}\right\} \; k = 0,$$

$$\left.\begin{aligned} \underline{s}_k &= \underline{\underline{M}}^{-1}\underline{r}_k \\ \beta_k &= \frac{\underline{s}_k^T \underline{r}_k}{\underline{s}_{k-1}^T \underline{r}_{k-1}} \\ \underline{p}_k &= \underline{s}_k + \beta_k \underline{p}_{k-1} \end{aligned}\right\} \; k = 1,2,\ldots, \tag{271}$$

$$\left.\begin{aligned} \alpha_k &= \frac{\underline{s}_k^T \underline{r}_k}{\underline{p}_k^T \underline{\underline{A}} \underline{p}_k} \\ \underline{x}_{k+1} &= \underline{x}_k + \alpha_k \underline{p}_k \\ \underline{r}_{k+1} &= \underline{r}_k + \alpha_k \underline{\underline{A}} \underline{p}_k \end{aligned}\right\} k = 0,1,2,\ldots,$$

Equations (271) can be derived using the idea that, if a set of linearly independent vectors $\underline{p}_k$, $k = 1,2,\ldots,N$, can be obtained, then the solution $\underline{x}$ can be written as a linear combination of the $\underline{p}_k$'s because this set of vectors spans the N-dimensional space. Such a set of linearly independent vectors can be obtained by constructing them to be A-conjugate, that is, so that $\underline{p}_i^T \underline{\underline{A}} \underline{p}_j = 0$ if and only if $i \neq j$ (Beckman, 1967, p. 63). Coefficients β_k are calculated to construct this set of vectors. The proper linear combination of the $\underline{p}_k$ vectors to give the solution $\underline{x}$ is obtained by minimizing the error in the solution along the line $\underline{x}_k + a\underline{p}_k$ at each iteration (Beckman, 1967, p. 64). The value of "a" that minimizes this error is given by α_k.

In the absence of round-off error, the exact solution $\underline{x}$ is obtained in N iterations. Thus, if nearly N iterations were actually needed to obtain a good approximation of $\underline{x}$, the method would not be useful for large systems of equations. Concus and others (1975) argued that the method can be considered to be a general iteration method that permits the gradual loss of A-orthogonality from round-off error and never converges to the exact solution. They showed that the weighted error function $\left(\underline{x} - \underline{x}_k\right)^T \underline{\underline{A}} \left(\underline{x} - \underline{x}_k\right)$ is reduced at each iteration if $\underline{\underline{M}}$ and $\underline{\underline{A}}$ are symmetric and positive definite, and that the method has certain optimality properties, so that, for a good choice of $\underline{\underline{M}}$, it usually converges to the desired accuracy in far fewer than N iterations.

Modified incomplete-Cholesky factorization

Modified incomplete-Cholesky factorization is an extension of a method introduced by Meijerink and van der Vorst (1977) known as incomplete-Cholesky factorization.[1] The extension is a combination of methods from Wong (1979) and Manteuffel (1980).

Wong's (1979) method, known as row-sums agreement factorization, is developed from incomplete-Cholesky factorization as follows. Let matrix entries located at $(\tilde{i},\tilde{j})$ be those entries corresponding to nonzero entries of $\underline{\underline{A}}$, and let $\underline{\underline{\tilde{U}}}$ be an upper triangular matrix with the same form as $\underline{\underline{U}}$, except that the only nonzero entries of $\underline{\underline{\tilde{U}}}$ are located at $(\tilde{i},\tilde{j})$. Finally, let $\underline{\underline{\tilde{D}}}$ be a diagonal matrix with the same form as $\underline{\underline{D}}$. Then an approximate (incomplete) factorization of $\underline{\underline{A}}$ is defined by

[1]Meijerink and van der Vorst (1977) used an approximate factorization that is more like the symmetric-Dolittle method than the Cholesky method. However, it is still called incomplete-Cholesky factorization.

$$\underline{\underline{M}} = \tilde{\underline{\underline{U}}}^T \tilde{\underline{\underline{D}}}\, \tilde{\underline{\underline{U}}}, \tag{272}$$

where $\underline{\underline{M}}$ will generally contain nonzero entries in addition to nonzero entries at the $(\tilde{i},\tilde{j})$ locations because of fill-in generated by forming the product $\tilde{\underline{\underline{U}}}^T \tilde{\underline{\underline{D}}}\, \tilde{\underline{\underline{U}}}$. Both incomplete-Cholesky and row-sums agreement factorization are based on equation (272).

For incomplete-Cholesky factorization, entries of $\tilde{\underline{\underline{D}}}$ and $\tilde{\underline{\underline{U}}}$ are obtained by equating entries of $\tilde{\underline{\underline{U}}}^T \tilde{\underline{\underline{D}}}\, \tilde{\underline{\underline{U}}}$ with nonzero entries a_{ij} of $\underline{\underline{A}}$ and rearranging the results, so that

$$\tilde{\alpha}_{ii} = a_{ii} - \sum_{k=1}^{i-1} \tilde{u}_{ki}\tilde{u}_{ki}/\tilde{\alpha}_{kk}, \quad i = 1,2,\ldots,N, \tag{273}$$

$$\tilde{u}_{ij} = \begin{cases} a_{ij} - \sum_{k=1}^{i-1} \tilde{u}_{ki}\tilde{u}_{kj}/\tilde{\alpha}_{kk}, & (i,j) \text{ belongs to } (\tilde{i},\tilde{j}) \\ 0 & , \ (i,j) \text{ does not belong to } (\tilde{i},\tilde{j}). \end{cases} \tag{274}$$

It can be verified by direct calculation that entries of $\underline{\underline{M}}$ and $\underline{\underline{A}}$ located at $(\tilde{i},\tilde{j})$ are identical. The two matrices differ because of fill-in in $\underline{\underline{M}}$. By assigning the negatives of the fill-in entries in $\underline{\underline{M}}$ to $\underline{\underline{N}}$ and letting all other entries in $\underline{\underline{N}}$ be zero, $\underline{\underline{A}} = \underline{\underline{M}} + \underline{\underline{N}}$, as required.

An ideal modification would make $\underline{\underline{N}}$ near zero, but this is not possible using equation (272) without adding nonzero entries to $\tilde{\underline{\underline{U}}}$. It is possible to modify $\underline{\underline{N}}$ to have the property of a zero matrix that the sum of entries of each row (a row sum) of $\underline{\underline{N}}$ equal zero. This is Wong's (1979) row-sum agreement factorization. To develop this method, each row sum of $\underline{\underline{A}}$ is set equal to each row sum of $\underline{\underline{M}}$ using equations (273) and (274) to define entries of $\underline{\underline{M}}$. Because $a_{ij} = a_{ji}$, entries of $\underline{\underline{A}}$ below the main diagonal where $\tilde{u}_{ij} = 0$ are given from equation (274) as $a_{ij} = \tilde{u}_{ji} + \sum_{k=1}^{i-1} \tilde{u}_{kj}\tilde{u}_{ki}/\tilde{\alpha}_{kk}$ so that a row sum is

$$a_{ii} + \sum_{j=1}^{i-1} a_{ij} + \sum_{j=i+1}^{N} a_{ij}$$

$$= \tilde{\alpha}_{ii} + \sum_{k=1}^{i-1} \tilde{u}_{ki}\tilde{u}_{ki}/\tilde{\alpha}_{kk} + \sum_{j=1}^{i-1} \left(\tilde{u}_{ji} + \sum_{k=1}^{i-1} \tilde{u}_{kj}\tilde{u}_{ki}/\tilde{\alpha}_{kk} \right)$$

$$+ \sum_{j=i+1}^{N} \left(\tilde{u}_{ij} + \sum_{k=1}^{i-1} \tilde{u}_{ki}\tilde{u}_{kj}/\tilde{\alpha}_{kk} \right). \tag{275}$$

The factorization is obtained from equation (275) by computing all values of $\tilde{u}_{ij}$ using equation (274), so that all nonzero off-diagonal entries of $\underline{\underline{A}}$ cancel with their corresponding entries of $\underline{\underline{M}}$. Thus, the only remaining entries in the sums on j in equation (275) result from fill-in, for which $\tilde{u}_{ij} = 0$, so that

$$a_{ii} = \tilde{\alpha}_{ii} + \sum_{k=1}^{i-1} \tilde{u}_{ki}\tilde{u}_{ki}/\tilde{\alpha}_{kk} + \sum_{j=1}^{i-1} f_{ji} + \sum_{j=i+1}^{N} f_{ij}, \tag{276}$$

where f_{ij} is a fill-in entry of $\underline{\underline{M}}$ defined by

$$f_{ij} = \begin{cases} \sum_{k=1}^{i-1} \tilde{u}_{ki}\tilde{u}_{kj}/\tilde{\alpha}_{kk}, & (i,j) \text{ does not belong to } (\tilde{i},\tilde{j}) \\ 0, & (i,j) \text{ belongs to } (\tilde{i},\tilde{j}). \end{cases} \tag{277}$$

Diagonal entry $\tilde{\alpha}_{ii}$ is calculated from equation (276) as

$$\tilde{\alpha}_{ii} = a_{ii} - \sum_{k=1}^{i-1} \tilde{u}_{ki}\tilde{u}_{ki}/\tilde{\alpha}_{kk} - \sum_{j=1}^{i-1} f_{ji} - \sum_{j=i+1}^{N} f_{ij}. \tag{278}$$

Comparison of equations (273) and (278) shows that diagonal entries of $\underline{\underline{A}}$ no longer equal diagonal entries of $\underline{\underline{M}}$ defined by equation (272).

The method from Manteuffel (1980) forces $\underline{\underline{M}}$ to be positive definite, as required by the generalized conjugate-gradient method. If $\underline{\underline{A}}$ is not a Stieltjes matrix, then $\underline{\underline{M}}$ as defined using incomplete-Cholesky factorization may not be positive definite (Meijerink and van der Vorst, 1977), which means that $\tilde{\alpha}_{ii}$ computed by equation (273) will not be positive. In this case, $\underline{\underline{M}}$ as computed for row-sums agreement factorization also may not be positive definite because $\tilde{\alpha}_{ii}$ calculated using (278) may be even smaller. Manteuffel (1980) showed that computation of $\tilde{\alpha}_{ii} \leq 0$ for incomplete-Cholesky factorization of finite-element matrices can be prevented by adding the product of an empirically determined, small positive number, δ, and a_{ii} to the right-hand side of equation (273). The analogous modification of equation (278) is

$$\tilde{\alpha}_{ii} = (1 + \delta)a_{ii} - \sum_{k=1}^{i-1} \tilde{u}_{ki}\tilde{u}_{ki}/\tilde{\alpha}_{kk} - \sum_{j=1}^{i-1} f_{ji} - \sum_{j=i+1}^{N} f_{ij}. \tag{279}$$

The matrix approximately factored by this modification of row-sums agreement factorization is thus $\underline{\underline{A}} + \delta\underline{\underline{I}}$ (where $\underline{\underline{I}}$ is the identity matrix), which is more diagonally dominant than $\underline{\underline{A}}$.

The above method is implemented here as follows. If $\tilde{\alpha}_{ii} \leq 0$ is detected during factorization, then factorization is stopped and a new value of δ, δ_{new}, is computed from the old value, δ_{old}, using the empirical equation

$$\delta_{new} = \frac{3}{2}\delta_{old} + 0.001, \tag{280}$$

where the initial value of δ_{old} is zero. Factorization is then reinitiated, and equation (280) is applied again if $\tilde{\alpha}_{ii} \leq 0$ is detected again, and so forth. This process is continued until a large enough value of δ is computed that all $\tilde{\alpha}_{ii} > 0$.

Gustafsson (1978, 1979) also presented a method that yields equations having forms similar to those of equations (274), (277), and (279). However, his method applies to finite-difference approximations for which $\underline{\underline{A}}$ is a Stieltjes matrix so that the motivation and method of choosing δ are different.

Based on equation (272), the solution of equation (269) is obtained using the forward and backward substitution steps of the symmetric-Doolittle method as

$$\begin{aligned} \underline{\underline{\tilde{U}}}^T \underline{\underline{\tilde{D}}} \underline{y}_k &= \underline{r}_k, \\ \underline{\underline{\tilde{U}}}^T \underline{s}_k &= \underline{y}_k, \end{aligned} \tag{281}$$

where entries of $\underline{\underline{\tilde{U}}}$ and $\underline{\underline{\tilde{D}}}$ are computed using equations (274) and (279), respectively. The remaining part of the algorithm implied by equations (281) is

$$\tilde{u}'_{ij} = \tilde{u}_{ij}/\tilde{\alpha}_{ii}, \quad (i,j) \text{ belongs to } (\tilde{I},\tilde{J}), \tag{282}$$

$$\left.\begin{aligned} y_i^k &= r_i^k - \sum_{\ell=1}^{i-1} \tilde{u}'_{\ell i} y_\ell^k \\ y'^k_i &= y_i^k/\tilde{\alpha}_{ii} \end{aligned}\right\} \quad i = 1,2,\ldots,N, \tag{283}$$

$$s_i^k = y'^k_i - \sum_{\ell=i+1}^{N} \tilde{u}'_{i\ell} s_\ell^k, \quad i = N,N-1,\ldots,1. \tag{284}$$

In applying the above algorithm, note that the factorization step to compute $\underline{\underline{\tilde{D}}}$ and $\underline{\underline{\tilde{U}}}$ is only done once before applying the generalized conjugate-gradient algorithm (equations (271)). At each iteration, $\underline{s}_k$ is computed using only equations (283) and (284). The factorization, forward substitution, and backward substitution steps are all fast and require little computer storage because $\underline{\underline{\tilde{U}}}$ is sparse like $\underline{\underline{\tilde{A}}}$. This combination of GCGM and modified incomplete-Cholesky factorization is called the modified incomplete-Cholesky conjugate-gradient method (MICCG method).

Stopping criteria

One stopping criterion is to terminate the algorithm whenever the maximum value of $\left|x_i^{k+1} - x_i^k\right|$ becomes small, or whenever

$$\max_i \left|x_i^{k+1} - x_i^k\right| = \max_i \left|\alpha_k p_i^k\right| \le \epsilon, \tag{285}$$

where x_i^k is an entry of $\underline{x}_k$, p_i^k is an entry of $\underline{p}_k$, and ϵ is a small positive number, such as 10^{-4}. The value of $\max_i \left|x_i^{k+1} - x_i^k\right|$ is usually assumed to be a rough measure of the error $\max_i \left|x_i - x_i^k\right|$ in the solution. However, the

conjugate-gradient algorithm can sometimes yield a value of $\max_i \left|x_i^{k+1} - x_i^k\right|$ that is small even when the solution is inaccurate. Thus, another criterion that is also a rough measure of $\max_i \left|x_i - x_i^k\right|$ is employed.

The residual given by equation (270) can be written for any row i as

$$a_{i1}\left(x_1 - x_1^k\right) + a_{i2}\left(x_2 - x_2^k\right) + \ldots + a_{iN}\left(x_N - x_N^k\right) = r_i^k \qquad (286)$$

Thus, because a_{ii} is positive,

$$\frac{|a_{i1}|}{a_{ii}}\left|x_1 - x_1^k\right| + \frac{|a_{i2}|}{a_{ii}}\left|x_2 - x_2^k\right| + \ldots + \frac{|a_{iN}|}{a_{ii}}\left|x_N - x_N^k\right| \geq \frac{|r_i^k|}{a_{ii}}, \qquad (287)$$

or

$$\frac{1}{a_{ii}} \sum_{j=1}^{N} |a_{ij}| \max_i \left|x_i - x_i^k\right| \geq \frac{|r_i^k|}{a_{ii}}. \qquad (288)$$

The sum $\sum_{j=1}^{N} |a_{ij}|/a_{ii}$ is generally in the range of 1 to 2, so is assumed to be unity. Therefore, a rough measure of $\max_i \left|x_i - x_i^k\right|$ is $\left|r_i^k\right|/a_{ii}$, and the additional stopping criterion is

$$\max_i \left|r_i^k\right|/a_{ii} \leq \epsilon. \qquad (289)$$

Note that if MICCG is used to solve the nonlinear equation (234), then there will be an inner MICCG iteration loop and an outer loop on the nonlinearity. An efficient way of employing MICCG for these problems is to set the convergence criterion ϵ to be larger than normal (say, larger than ϵ_s by about an order of magnitude) to reduce the number of inner iterations taken at each outer iteration. Good accuracy is achieved by requiring close convergence of the outer iteration sequence.

COMPARISONS OF NUMERICAL RESULTS WITH ANALYTICAL SOLUTIONS

Results of simulating some simple ground-water flow problems for which analytical solutions have been presented in the literature are given here to demonstrate the accuracy of the finite-element code (MODFE). Each simulation is designed to test specific computational features that were discussed in preceding sections and to verify that MODFE can accurately represent the physical processes. To demonstrate that any consistent system of units may be used with MODFE, both English and metric systems of units are used in the example problems.

THEIS SOLUTION OF UNSTEADY RADIAL FLOW TO A PUMPED WELL

MODFE is used with axisymmetric cylindrical coordinates to compute unsteady flow to a well located in a confined nonleaky aquifer having homogeneous and isotropic hydraulic properties and an infinite areal extent.