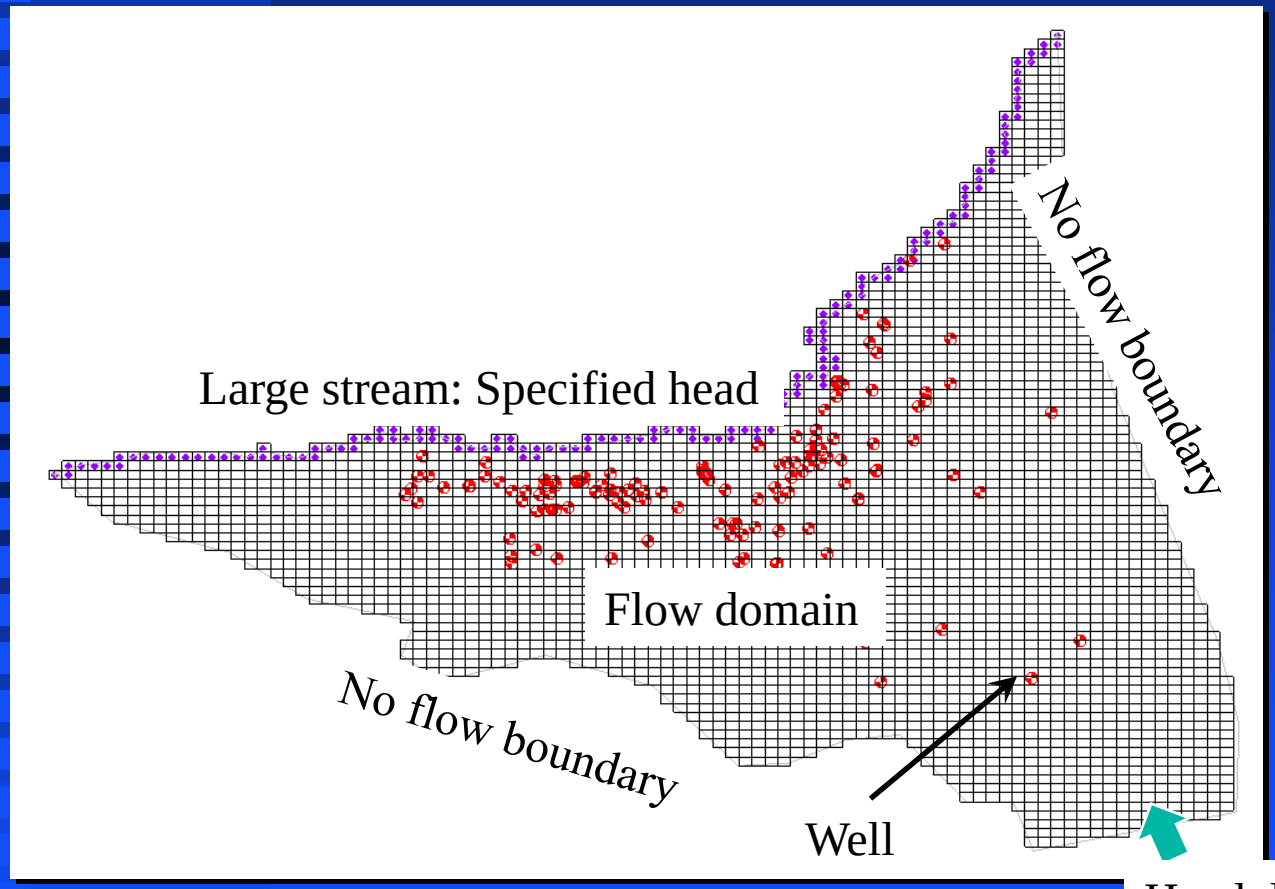
ERTH655/CEE623
Groundwater Modeling

Finite Difference Technique

Aly I. El-Kadi

Finite difference grid



Head dependent inter-aquifer flow

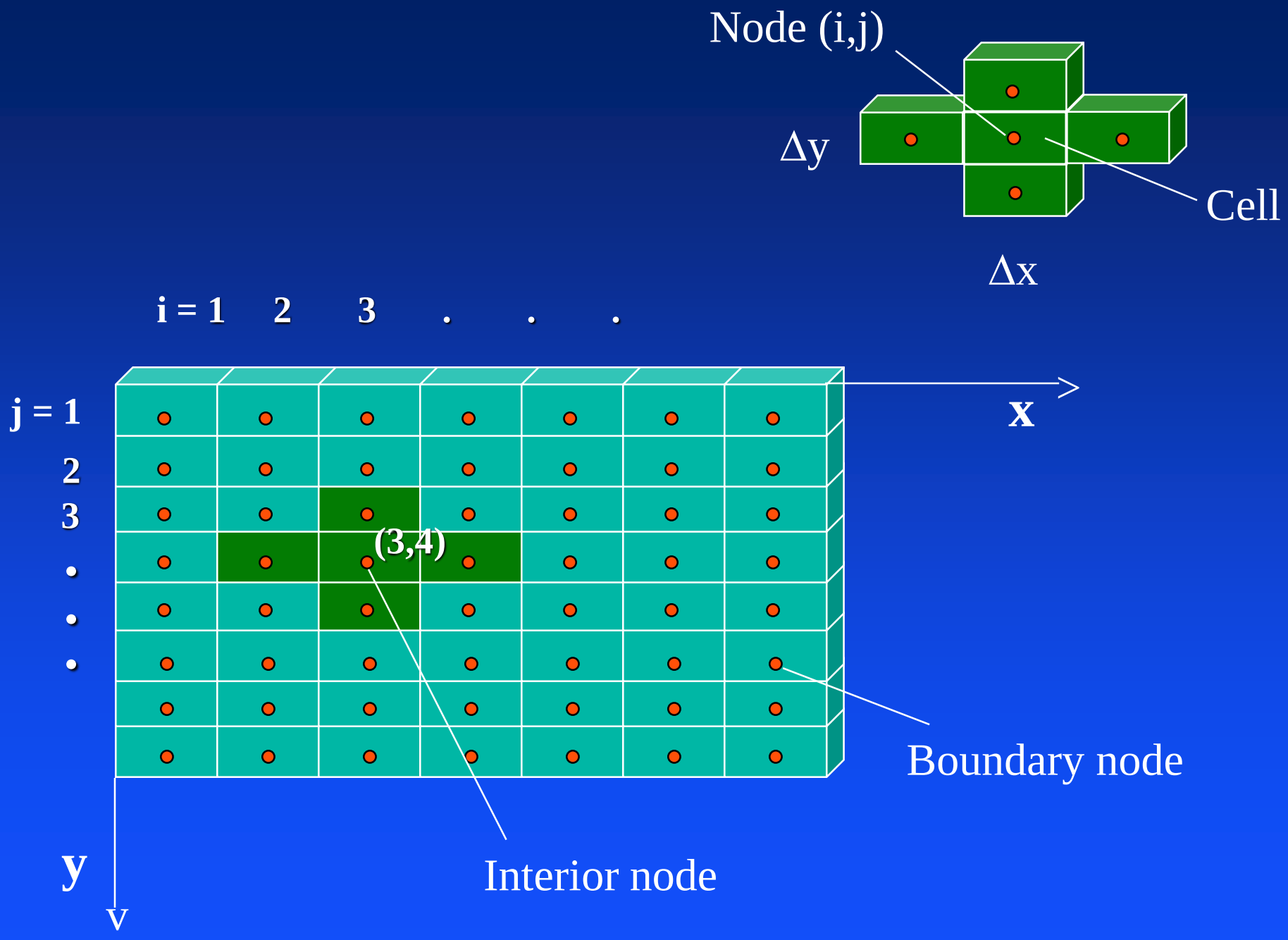
Example: Saturated fresh water flow for steady state

Objective: estimate water levels

Input: well fluxes, recharge boundary conditions, aquifer data, (conductivity, storage coefficient, aquifer thickness)

$$\frac{\partial}{\partial x}\left(K_{xx}\frac{\partial h}{\partial x}\right)+\frac{\partial}{\partial y}\left(K_{yy}\frac{\partial h}{\partial y}\right)+\frac{\partial}{\partial z}\left(K_{zz}\frac{\partial h}{\partial z}\right)\pm R=S_s\frac{\partial h}{\partial t}$$

Saturated 3-D equation


Node (i,j)
Δy
Cell
Δx
i = 1 2 3 . . .
j = 1
2
3
.
.
.
x
(3,4)
Boundary node
Interior node
y
v

Mass balance equation for a set of neighboring nodes:

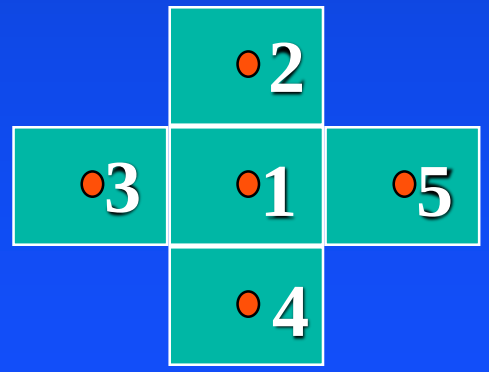
$$Q_{21} + Q_{31} + Q_{41} + Q_{51} = \Delta x_1 \Delta y_1 S_1 \frac{\partial h_1}{\partial t} \qquad (1)$$

Darcy's law:

$$Q_{21} = -\Delta x_1 T_{21} \left(\frac{\partial h}{\partial y} \right)_{21} \qquad (2)$$

Difference form:

$$Q_{21} \approx \Delta x_1 T_{21} \left(\frac{h_2 - h_1}{\Delta y} \right)_{21} \qquad (3)$$



Definition of a partial derivative:

$$\frac{\partial h_1}{\partial t} \approx \frac{h_1^n - h_1^{n-1}}{\Delta t} \qquad (4)$$

n = current time level ($t+\Delta t$), n-1 = previous time level (t). Inserting equations (3) and (4) into (1) will provide a relationship between head values at the neighboring nodes:

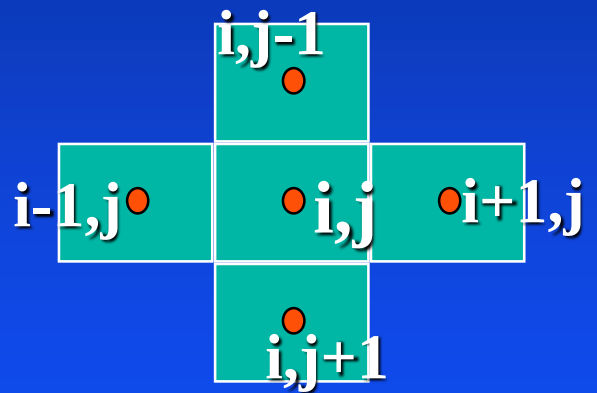
(5)

$$a_1 h_2^n + b_1 h_3^n + c_1 h_1^n + d_1 h_4^n + e_1 h_5^n = f_1$$

in which a, b, c, d, e, and f are known coefficients, functions of Δx, Δy, Δt, T, and S, with values that depend on the type of finite-difference approximation.

Equation (5) can be generalized at node i,j to read:

$$a_i h^n_{i-1,j} + b_i h^n_{i,j-1} + c_i h^n_{i,j} + d_i h^n_{i+1,j} + e_i h^n_{i,j+1} = f_i \qquad (6)$$



Types of finite-difference approximation

$$Q_{21} = -\Delta x_1 T_{21} \left(\frac{\partial h}{\partial y} \right)_{21} \qquad (7)$$

$$Q_{21} \approx \Delta x_1 T_{21} \left(\frac{h_2 - h_1}{\Delta y} \right)_{21}$$

Equations above estimates the average volumetric flux between cells over a time interval Δt. Heads used in this equation can be taken as those at certain time t' in the interval between t and t+Δt:

$$h(t') = (1-\omega)h(t) + \omega h(t+\Delta t) \qquad (8)$$

in which ω is a weighting factor between 0 and 1.

t = t’ or ω = 0 will produce an explicit finite difference scheme.

$$h(t') = h(t)$$

$$h_{i,j}^{n} = g_i h_{i,j}^{n-1} + m_i h_{i-1,j}^{n-1} + u_i h_{i+1,j}^{n-1} + v_i h_{i,j-1}^{n-1} + q_i h_{i-1,j+1}^{n-1} \quad (9)$$

The solution is obtained by starting from the initial values and estimating directly the values after a time increment Δt for all nodes in the system. The values of g_i, m_i, etc. are functions of Δx, Δy, Δt, T, and S.

The solution is easy to program but a restriction should be made on the value of Δt in order to obtain a stable solution. For stability of the solution, value of Δt should satisfy the condition at every node:

$$\frac{T}{S}\left(\frac{\Delta t}{\Delta x^2}+\frac{\Delta t}{\Delta y^2}\right)\leq\frac{1}{2} \qquad (10)$$

→ Small Δt.

Plus large rounding error O(Δx).

- For $\omega = 1$ → fully implicit scheme
- For $0 < \omega < 1$ → implicit scheme
- For $\omega = 0.5$ → well known Crank-Nicholson implicit scheme.

$$a_i h^n_{i-1,j} + b_i h^n_{i,j-1} + c_i h^n_{i,j} + d_i h^n_{i+1,j} + e_i h^n_{i,j+1} = f_i$$

- The exact expressions for the quantities a_i, b_i, c_i, etc., will depend on ω

For example, for fully implicit scheme, and homogenous and isotropic systems:

$$a_i = \frac{T}{\Delta x^2} \qquad b_i = \frac{T}{\Delta y^2}$$

$$c_i = -\frac{2T}{\Delta x^2} - \frac{2T}{\Delta y^2} - \frac{S}{\Delta t} \tag{11}$$

$$d_i = \frac{T}{\Delta x^2} \qquad e_i = \frac{T}{\Delta y^2}$$

$$f = -\frac{Sh_{i,j}^{n-1}}{\Delta t}$$

Implicit formulations (systems of equations)can be put in the matrix form:

$$[K]\{h\} = \{F\} \qquad (12)$$

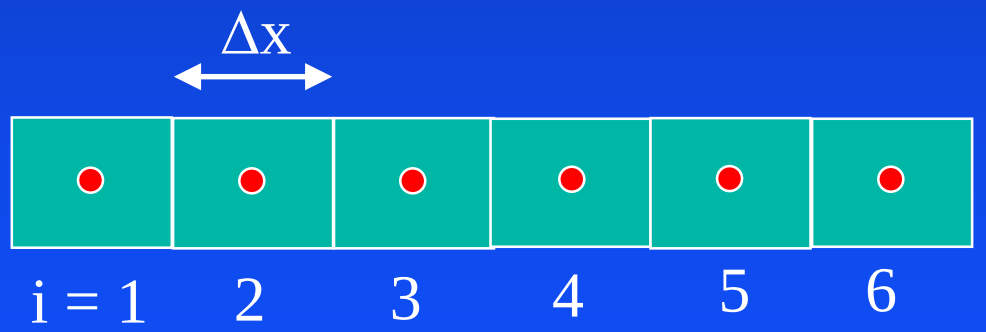
In which K is matrix of known coefficient, h is unknown vector, and F is known vector. Can be solved by matrix techniques either by direct or indirect (iterative) techniques.

Example – One dimensional problem

$$a_i h_{i-1} + b_i h_i + c_i h_{i+1} = d_i$$

i = 1, 2, … N; N total number of nodes

N = 6:



Equations can be written as:

$$
\begin{aligned}
i = 1 \quad & b_1h_1 + c_1h_2 && = d_1 \\
2 \quad & a_2h_1 + b_2h_2 + c_2h_3 && = d_2 \\
3 \quad & a_3h_2 + b_3h_3 + c_3h_4 && = d_3 \\
4 \quad & a_4h_3 + b_4h_4 + c_4h_5 && = d_4 \\
5 \quad & a_5h_4 + b_5h_5 + c_5h_6 && = d_5 \\
6 \quad & a_6h_5 + b_6h_6 && = d_6
\end{aligned}
$$

Or in matrix form:

$$\begin{bmatrix} b_1 & c_1 & & & & \\ a_2 & b_2 & c_2 & & & \\ & a_3 & b_3 & c_3 & & \\ & & a_4 & b_4 & c_4 & \\ & & & a_5 & b_5 & c_5 \\ & & & & a_6 & b_6 \end{bmatrix} \begin{Bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \end{Bmatrix} = \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \end{Bmatrix}$$

Matrix of known coefficients

Unknown vector at time n

Known vector at time n-1

Matrix direct schemes

- Matrix inversion
- Elimination – back substitution

Matrix iterative schemes

The Jacobi method

Arrange equation in the form:

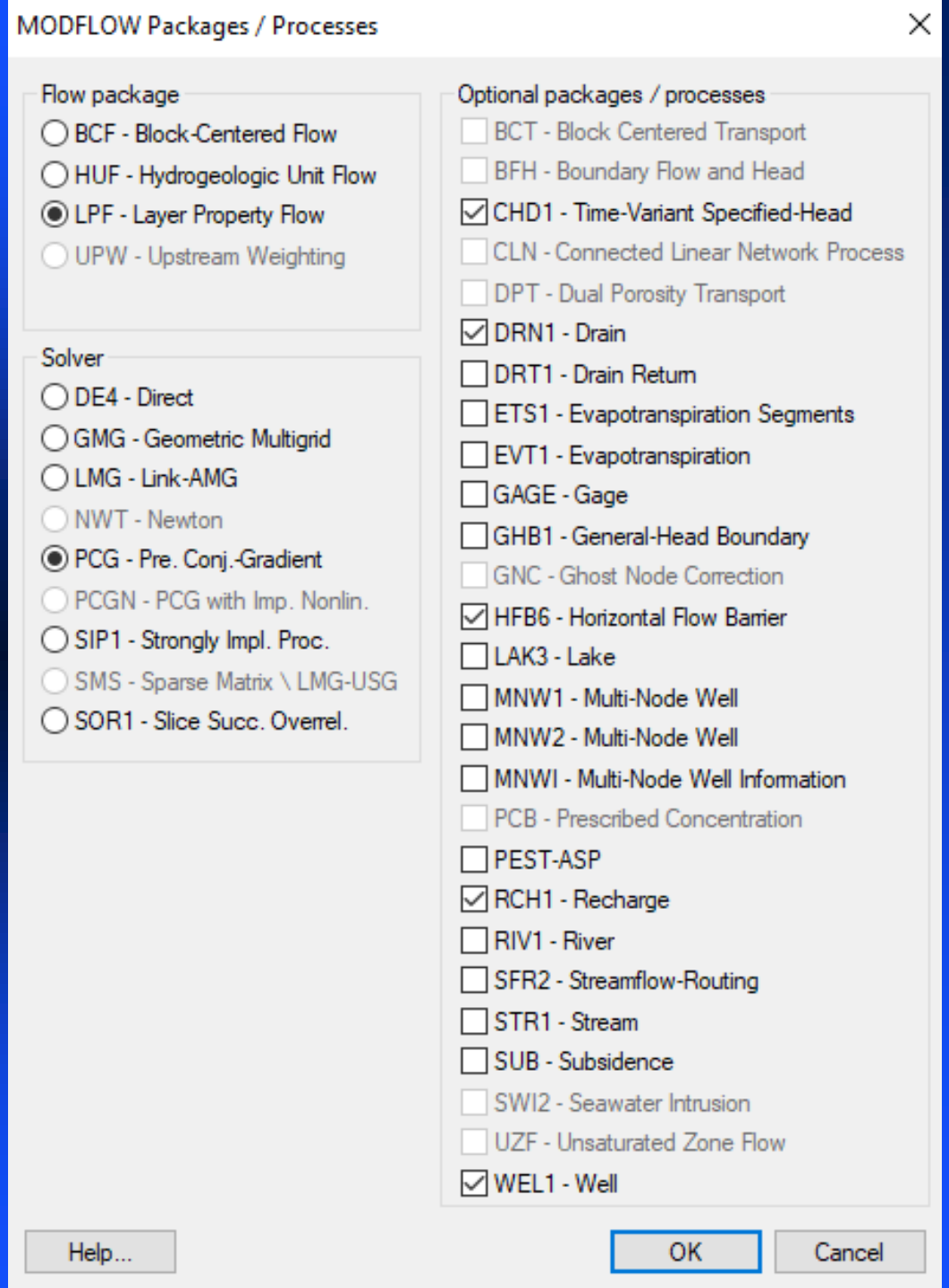
$$h_{i,j}^{n} = F\left(h_{i-1,j}^{n}, h_{i+1,j}^{n}, h_{i,j-1}^{n}, h_{i,j+1}^{n}\right)$$

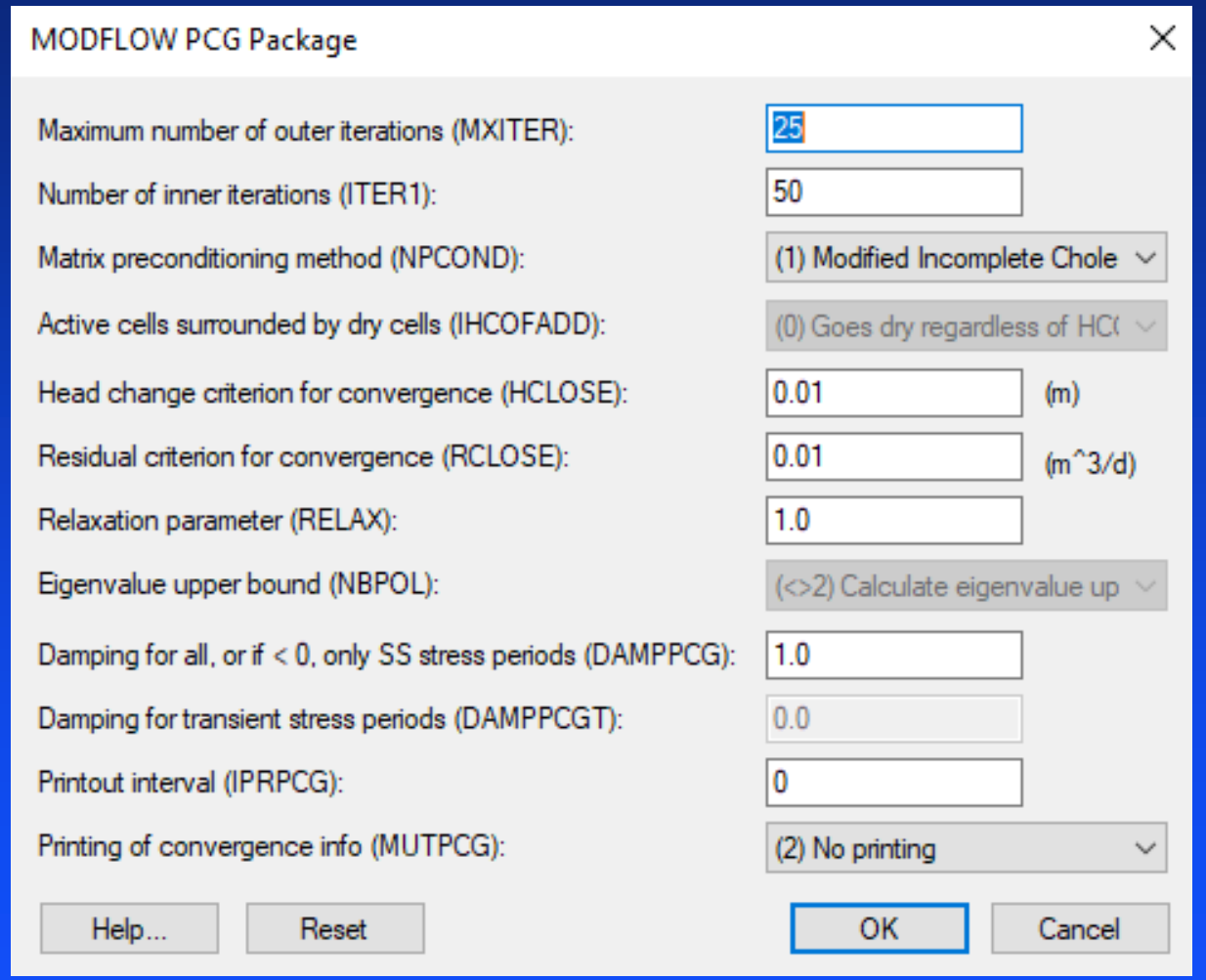
If neighboring values are known, can calculate $h_{i,j}$. But they are not, so make a guess, starting from initial condition. Then update using the new values as guesses and continue until convergence within acceptable error.

The Gauss-Seidel method

Same but use the last updated values (go row by row)

Others: Over relaxation, iterative alternating direction implicit procedure, etc.


MODFLOW Packages / Processes
Flow package
BCF - Block-Centered Flow
HUF - Hydrogeologic Unit Flow
LPF - Layer Property Flow
UPW - Upstream Weighting
Solver
DE4 - Direct
GMG - Geometric Multigrid
LMG - Link-AMG
NWT - Newton
PCG - Pre. Conj.-Gradient
PCGN - PCG with Imp. Nonlin.
SIP1 - Strongly Impl. Proc.
SMS - Sparse Matrix \ LMG-USG
SOR1 - Slice Succ. Overrel.
Optional packages / processes
BCT - Block Centered Transport
BFH - Boundary Flow and Head
CHD1 - Time-Variant Specified-Head
CLN - Connected Linear Network Process
DPT - Dual Porosity Transport
DRN1 - Drain
DRT1 - Drain Return
ETS1 - Evapotranspiration Segments
EVT1 - Evapotranspiration
GAGE - Gage
GHB1 - General-Head Boundary
GNC - Ghost Node Correction
HFB6 - Horizontal Flow Barrier
LAK3 - Lake
MNW1 - Multi-Node Well
MNW2 - Multi-Node Well
MNWI - Multi-Node Well Information
PCB - Prescribed Concentration
PEST-ASP
RCH1 - Recharge
RIV1 - River
SFR2 - Streamflow-Routing
STR1 - Stream
SUB - Subsidence
SWI2 - Seawater Intrusion
UZF - Unsaturated Zone Flow
WEL1 - Well
Help...
OK
Cancel


MODFLOW PCG Package
Maximum number of outer iterations (MXITER):
25
Number of inner iterations (ITER1):
50
Matrix preconditioning method (NPCOND):
(1) Modified Incomplete Chole
Active cells surrounded by dry cells (IHCOFADD):
(0) Goes dry regardless of HC(
Head change criterion for convergence (HCLOSE):
0.01
(m)
Residual criterion for convergence (RCLOSE):
0.01
(m^3/d)
Relaxation parameter (RELAX):
1.0
Eigenvalue upper bound (NBPOL):
(<>2) Calculate eigenvalue up
Damping for all, or if < 0, only SS stress periods (DAMPPCG):
1.0
Damping for transient stress periods (DAMPPCGT):
0.0
Printout interval (IPRPCG):
0
Printing of convergence info (MUTPCG):
(2) No printing
Help...
Reset
OK
Cancel