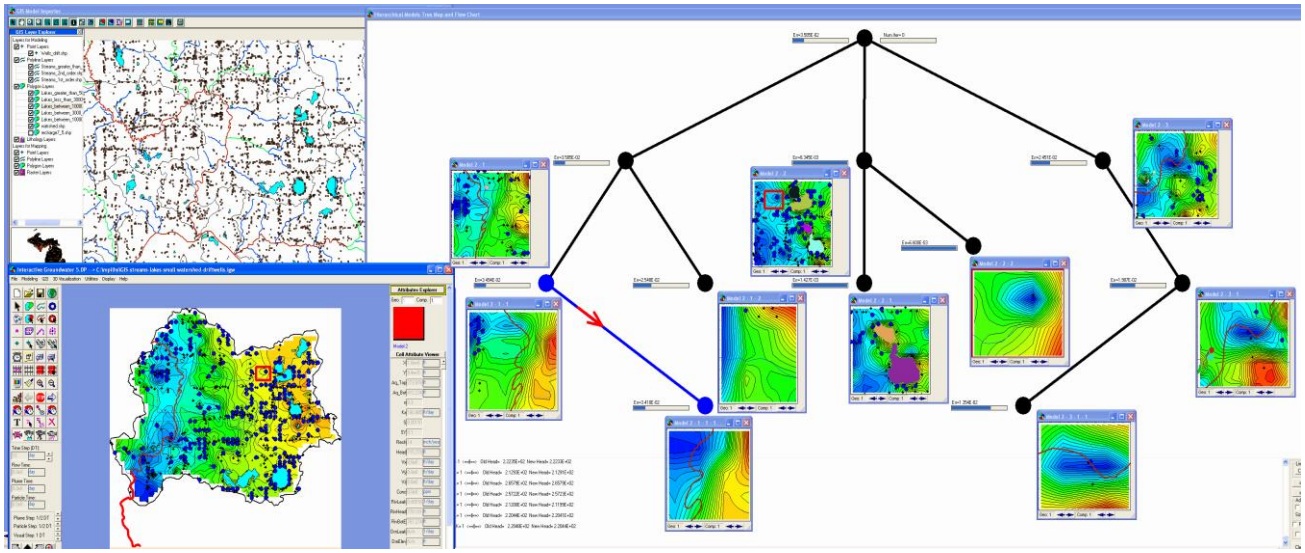
Algorithm and Numerical Schemes in Multi-scale Modeling



By Huasheng Liao

Laboratory of Excellence for Realtime Computing and Multiscale Modeling

Michigan State University

March 2010

Contents

Algorithm and Numerical Schemes in Multiscale Modeling

---------- By Huasheng Liao
March, 2010
MSU

1 Governing Equations and Algorithms

In hierarchical modeling, governing equation in each model is the same as the general groundwater governing equation as shown below

$$S_s \frac{\partial H}{\partial t} = \nabla(K \bullet \nabla H) + q \tag{1}$$

Where S_s is the aquifer specific storage coefficient [L^{-1}], H is piezometric head [L], t is time [T], ∇ is the gradient operator [L^{-1}], K is the saturated hydraulic conductivity tensor [L/T], q represents source (positive) or sink (negative) terms [$L^3/(L^3T)$] including pumping/injecting wells, streams, lakes, drains etc.

With the following well-defined boundary conditions (BCs) and initial conditions (ICs), Eq(1) can be solved numerically.

$$\begin{aligned} BC &\begin{cases} H|_{\Gamma 1} = f(\vec{x},t) & \text{Dirichlet} \\ H|_{\Gamma 2} = g(\vec{x},t) & \text{Neumann} \end{cases} \\ IC:&\ \ H(\vec{x},0) = h(\vec{x}) \end{aligned} \tag{2}$$

Where Γ1 is the computational domain boundary on which H distribution, f, is known, Γ2 is the boundary where a known flux, g, is specified, $\vec{x}$ is the spatial vector and $h(\vec{x})$ is head distribution over the whole computational domain at initial time step t=0.

More specifically, assuming that there are L nested model levels and P(l) patches in the lth model level in a multi-scale modeling system denoting as M^{pl} [p=1, P(l), l=1,L] and illustrating hierarchically in Fig.1. The naming convention used here will be described as:

- **Main Model**: the most top level model, referred to as regional model or coarse model
- **Parent model**: finer grid, at least has one child model
- **Child model**: finer grid than its parent model, referred to as patch model, has only one parent model, orphan models are not allowable due to the absence of BCs

The governing equations in this system, then, can be given as the following,

$$\begin{cases} S_s^{p,l}\dfrac{\partial H^{p,l}}{\partial t} = \nabla \bullet (K^{p,l}\nabla H^{p,l}) + q^{p,l} \\ H^{p,l}\big|_{\Gamma_1^{p,l}} = f^{p,l}(\vec{x},t) \\ (K^{p,l}\nabla H^{p,l} \bullet \vec{n})\big|_{\Gamma_2^{p,l}} = g^{p,l}(\vec{x},t) \\ H^{p,l}(\vec{x},0) = h^{p,l}(\vec{x}) \end{cases} \qquad (p = 1, P(l); l = 1, L) \tag{3}$$

where the combination subscript (*p, l*) refers to as a submodel or a child model that is the *p*th patch in the *l*th model level, $\vec{n}$ is outward normal vector of boundary $\Gamma_2^{p,l}$.It should be noted that material parameters or source/sink terms may vary across scales, which means that material parameters or source/sink terms in different model could be different --- more details would be resolved as model grid becomes finer and finer.

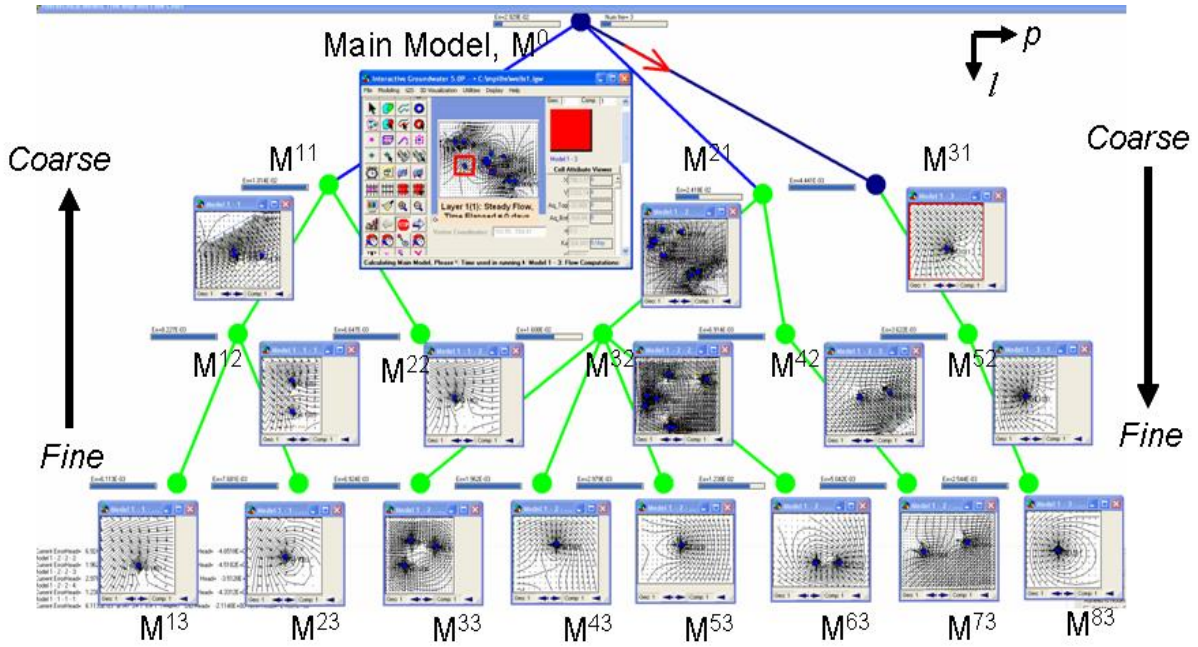


Figure 1 Hierarchical Modeling System

In general, boundary conditions (BCs) and initial conditions (ICs) are only provided in the main model (i.e. the largest scale model). In order to obtain solutions to each level's models, their BCs and ICs must be well-imposed. This can be achieved by an interactive way:

(1) With the given BCs and ICs, main model can be solved numerically and its head, H^0, will be obtained throughout the whole computational domain.
(2) Head, $H^{p,1}$, along the interfaces of main model and its subsequent models (patches) can be interpolated from H^0; or fluxes, $K^{p,1}\nabla H^{p,1} \bullet \vec{n}$, crossing the interfaces can be calculated from H^0 too. ICs of child models can be obtained by interpolating heads inside the domains from H^0. These calculation details will be given in next sections.
(3) With $H^{p,1}$ or $K^{p,1}\nabla H^{p,1} \bullet \vec{n}$ known on domain boundaries and interpolated ICs,

$H^{p,1}$ $(p=1,P(l))$ can be solved and ready for use in its next level models.

(4) Similarly, once $H^{p,l}$ $(p=1,P(l);l=1,L-1)$ are solved with the appropriate BCs and ICs derived from $H^{p,l-1}$, head of their parent model; their $H^{p,l+1}$ or $K^{p,l+1}\nabla H^{p,l+1}\bullet\vec{n}$ along the interfaces of parent-child models can be calculated in the same way until $l=L$. This procedure is called as downscaling.

(5) As the head, $H^{p,L}$, in the last model level have been calculated, they will be used as the base head to update the heads along the interfaces. This will result in a change in the BCs of their parent models (upper level models): $H^{p,l-1}$ or $K^{p,l-1}\nabla H^{p,l-1}\bullet\vec{n}$ along interface of parent-child models will be calculated from their child model head, $H^{p,l}$, and $H^{p,l-1}$ is updated until main model $(l=0)$ is reached. This procedure is called as up scaling.

(6) Repeat step (1) to (5) until the maximum head difference at current iteration and previous iteration meets a given convergent criterion, then the whole modeling system is stopped.

Flow chart of the mentioned down and up scaling interaction procedures in a hierarchical modeling system is shown in Fig. 2 and Fig.3.

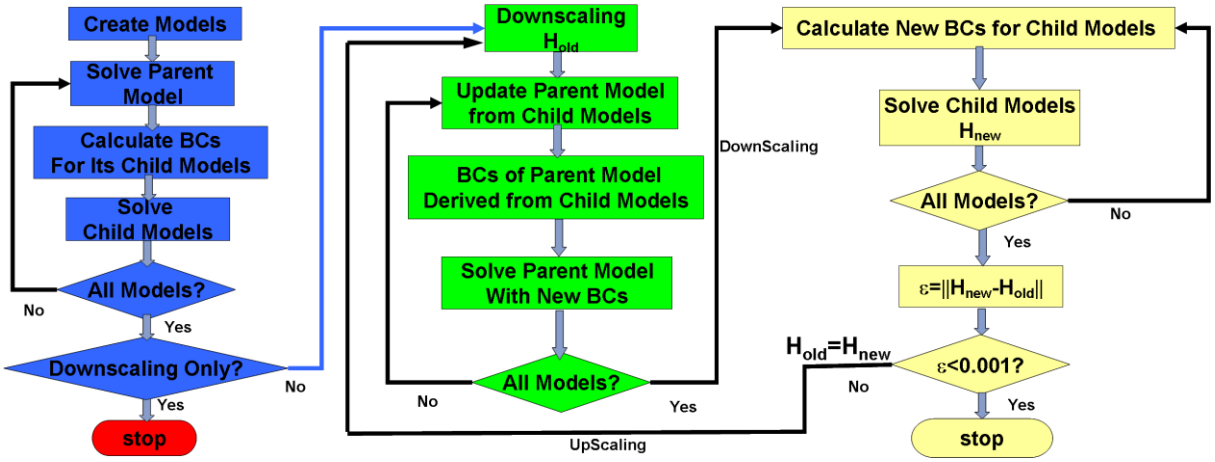


Figure 2 Flow chart of down and up scaling interaction procedures in a hierarchical **modeling system**

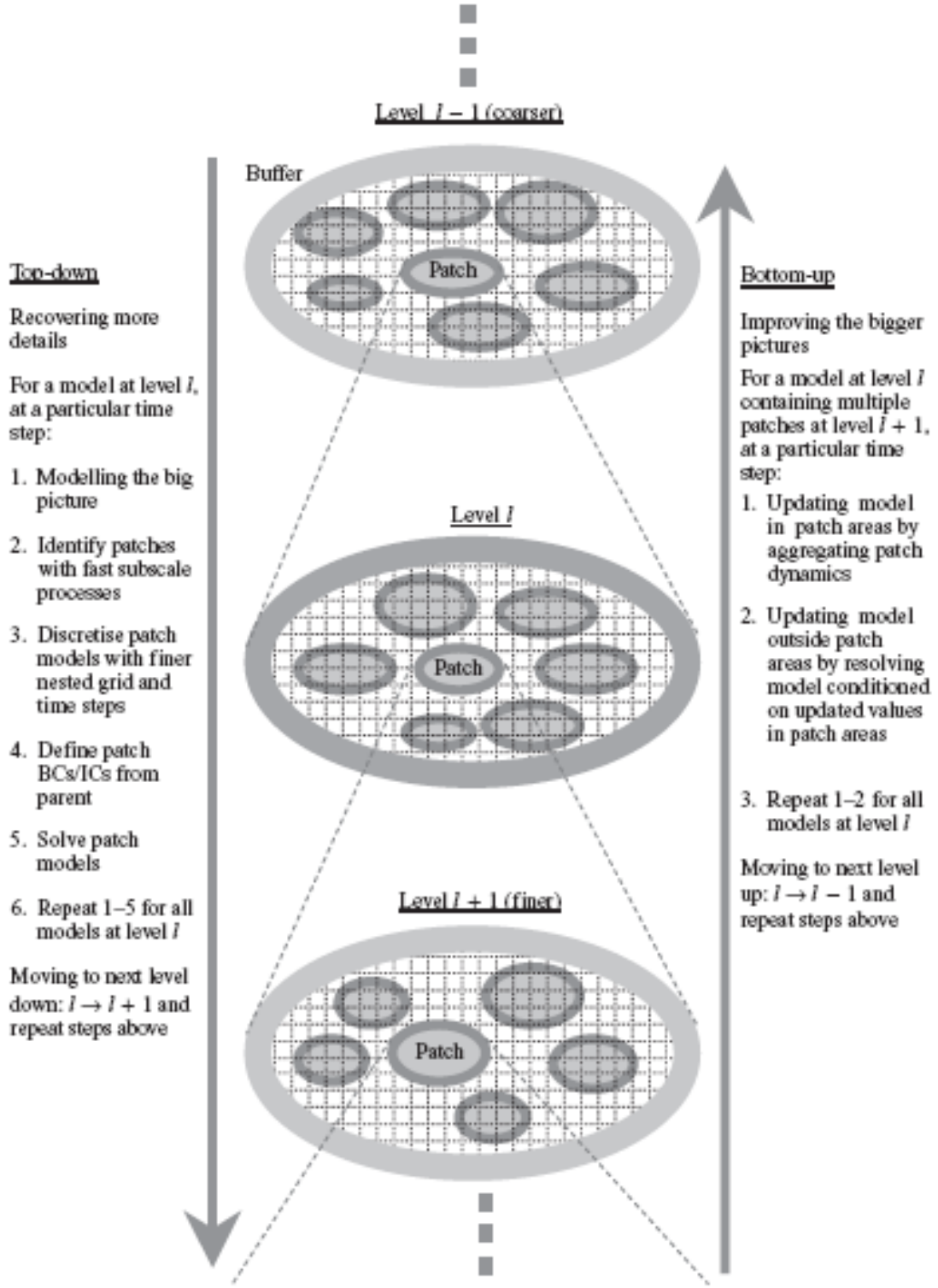


Figure 3 **Hierarchical patch dynamics modeling: iterative double sweeps from top down and bottom up**

As to a transient case, temporal loop is needed to accomplish the time marching procedure in multiscale modeling system. If no temporal scale issue is considered, the whole down and up scaling loops for all models in the system can be implemented in one time step, once a convergence is achieved, then proceed to next time step. Fig.4 shows the flow chart in transient case without nested time steps in the child model.

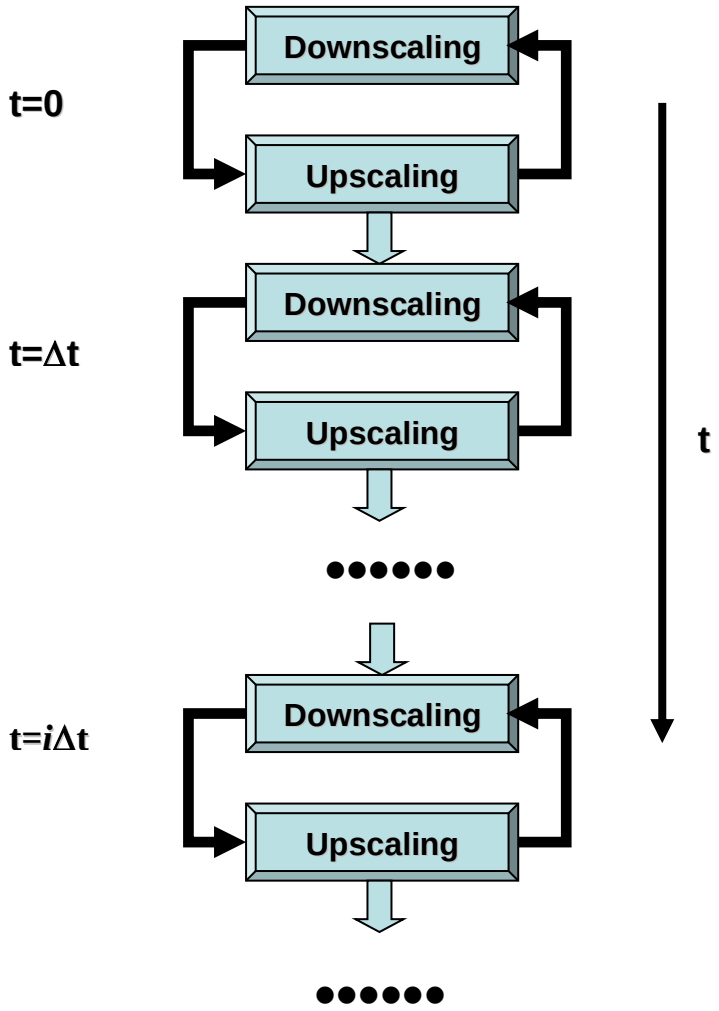


Figure 4 Flow chart of down and up scaling interaction procedure in a transient multiscale modeling system with uniform time step

Groundwater flow may include variations at well scale, site scale and regional scale across disparate length and time. This requires grid and time step constraints to be respected in order to provide accurate results and make computation more efficient. Multiple spatial scales can be resolved by appropriate selection of the grid size in every level model in a hierarchical way as illustrated in Fig.5.

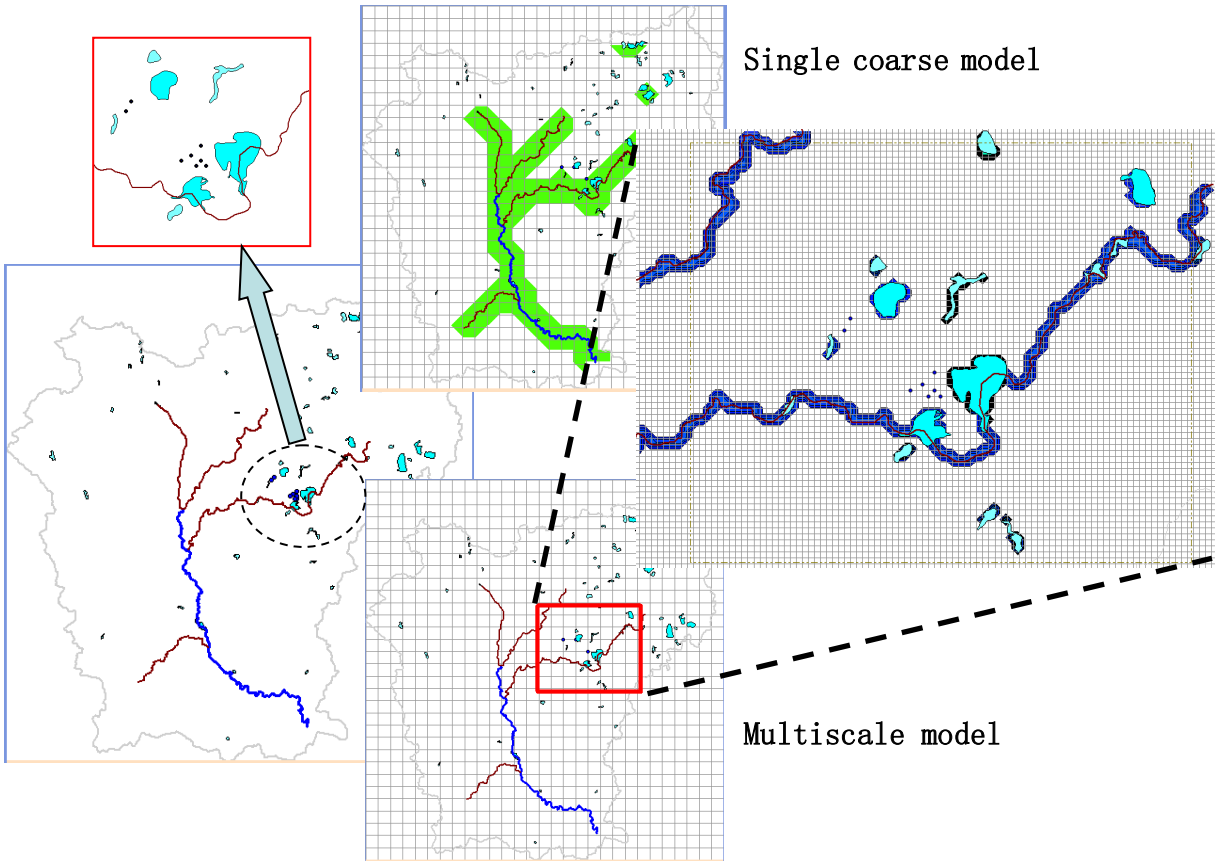


Figure 5. Hierarchical modeling to resolve multiple spatial scales

As to the multiple temporal scales, a nested time stepping should be applied, which means that multiple small time steps of a child model are nested in one large time step of its parent model and a temporal information across different scales propagating between upper and lower model levels will be needed due to the fact that BCs on the interfaces of parent-child models are time dependent in a transient state. Therefore, in multiscale modeling, the nested time stepping will also be presented in a hierarchical way, which can be described as below.

(1) Time step, Δt^l, at each model level would be different from each other, but should be the same in each patch model in the same level

(2) Downscaling procedure will start from the main model and its initial conditions are needed to conduct a steady-state downscaling-upscaling loop at t=0 for all level's models such that initial conditions for every submodel become available. Otherwise, initial conditions for every submodel must be given.

(3) To advance to next time step with known heads at previous time step in nested time stepping algorithm is totally different from that of uniform time stepping algorithm as shown in Fig.4 since there are not only shared "time nodes" in the "time grid" system (solid lines in Fig.6) but also non-shared ones (dash lines in Fig.6). Information propagating should be carried on at the same time and along the same interface of parent-child models. Interpolation would be involved both spatially and temporally in this algorithm. Considering one time step in the main model as one unit of downscaling-upscaling loop in the nested time stepping algorithm, a local information updating approach (LIUA) is presented and described in the following steps.

(4) Starting with the main model to obtain H_n^0 , head at current time level n , LIUA will

solve the transient equation with given H_{n-1}^0 , head at previous time level n-1, as its initial conditions. These heads, H_n^0 and H_{n-1}^0 will be used to interpolate those heads on the interfaces of the main and its child models to be used as BCs of thoses child models. Temporal interpolation would be necessary if time step in child models is not the same as that of the main model, whereas there is no need to do any temporal interpolation when "time node" is shared by both child model and its parent model (solid lines in Fig.6).

(5) With current BCs derived from model level $l-1$ are known and ICs from the previous time step, transient equations solving will be implemented in the most first time step, Δt^l in model level l, then temporal interpolation will be conducted for heads at the next nested time step Δt^{l+1} ($\Delta t^l > \Delta t^{l+1}$, and usually, Δt^{l+1} can be designed to be a factor of Δt^l) along with non-shared "time node", these heads would be used as BCs of models at level $l+1$. In model level $l+1$, again, with these BCs and ICs from previous time step, transient equations solving and temporal interpolation can be done in the first time step Δt^{l+1} and BCs for model level $l+2$ at nested time step Δt^{l+2} (again, $\Delta t^{l+1} > \Delta t^{l+2}$ and Δt^{l+2} could be a factor of Δt^{l+1}) can be obtained, …, this procedure can be continuously applied until the last child model is reached.

(6) Once solving of transient equations in the first time step in the last child model is completed, all heads at those time steps before the first shared "time node" will be obtained. Shared "time node" heads then will be used to create a new updating ICs for its parent model to advance to parent model's next time step calculation. If this shared "time node" does share by this parent model's parent model, the updating will be proceeding upward till this shared "time node" path line is ended.

(7) At a model level where the shared "time node" path line ended, with the updated ICs and BCs derived from its parent model, head at second (next) time step can be solved, and then, following the same procedure done for the first time step in step(5), going through step (6) again, the whole nested time steps within this unit will be covered and the updated heads at each nested time step (different temporal scales) will be given and available for next iterative loop. The down scaling-upscaling iterative loop will continue within this down scaling-upscaling unit until the convergent criterion is met.

(8) Once convergence occurs in the previous down scaling-upscaling unit, heads in each model will be used as the ICs of next down scaling-upscaling unit, then repeat step (5) to (8) until the total length of simulation time is reached.

Fig.6 shows the flow chart of the LIUA with an example of one patch-4 level system in one down scaling-upscaling unit. The sequential operation order is also illustrated in the figure. Table 1 gives the total number of each kind of operation such as PDE solving, temporal interpolation and boundary condition interpolation occurring in each model during one down and up scaling loop.

Table 1 Total number of operations occurring during one down and up scaling loop

Level / Operation	Main Model M^0	Level 1 M^1	Level 2 M^2	Level 3 M^3
Solving Eq(1)	1	2	4	8
Temporal Interpolation	1	2	4	/
Boundary Conditions for Downscaling	2	4	8	/
Boundary Conditions for Upscaling	/	1	2	4

The overall flow chart is shown in Fig.7. As to multiple patches, same procedure will be applied to each patch and updating in all patches in the same model level must be completed before proceeding down scaling/up scaling to their parent or child models.

It is very clear from the proposed LIUA that both spatial and temporal information are immediately propagated between parent and child models. This reflects the mechanism of time response process in the real world.

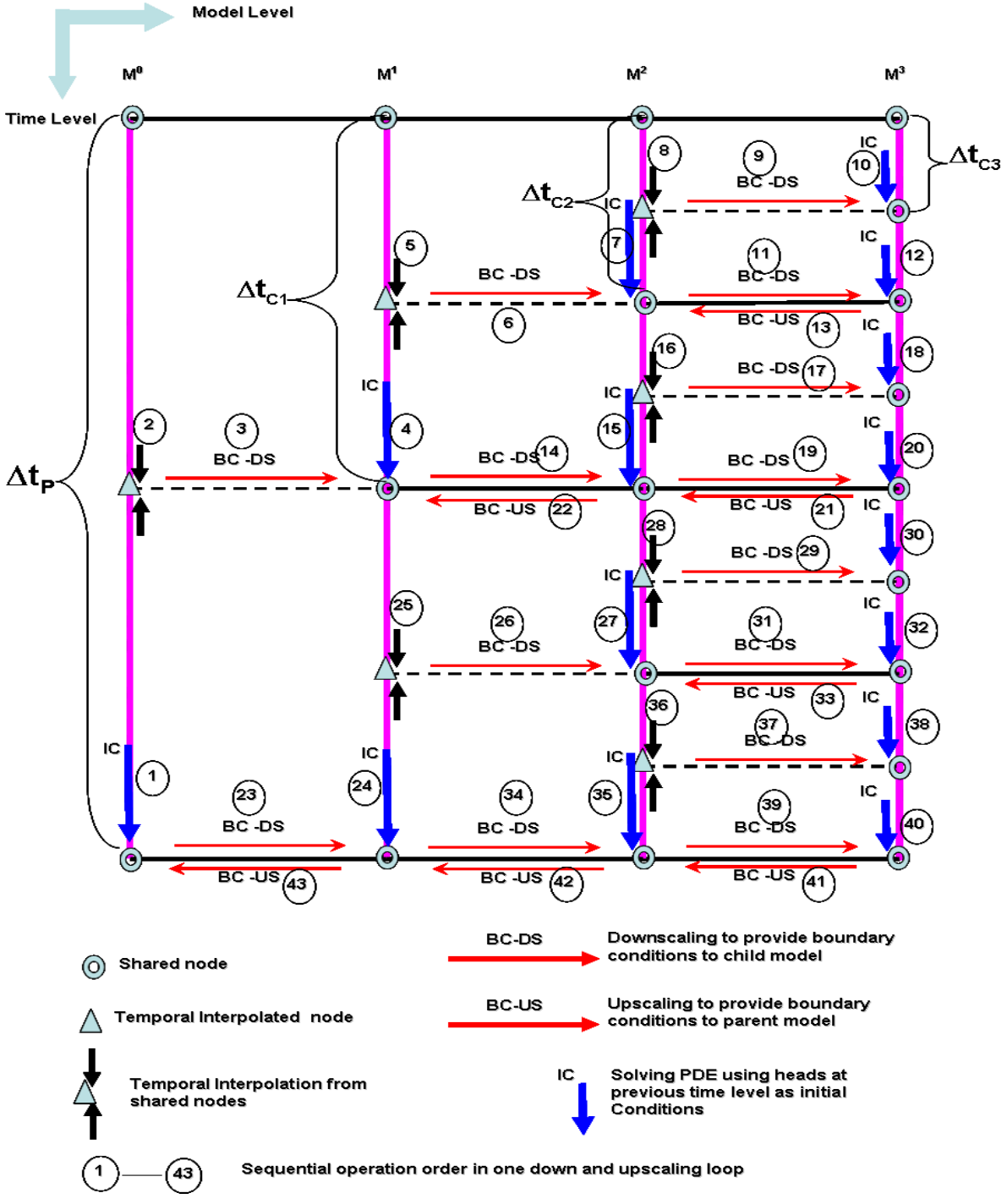


Figure 6. Hierarchical modeling to resolve multiple temporal scales (one unit of LIUA)

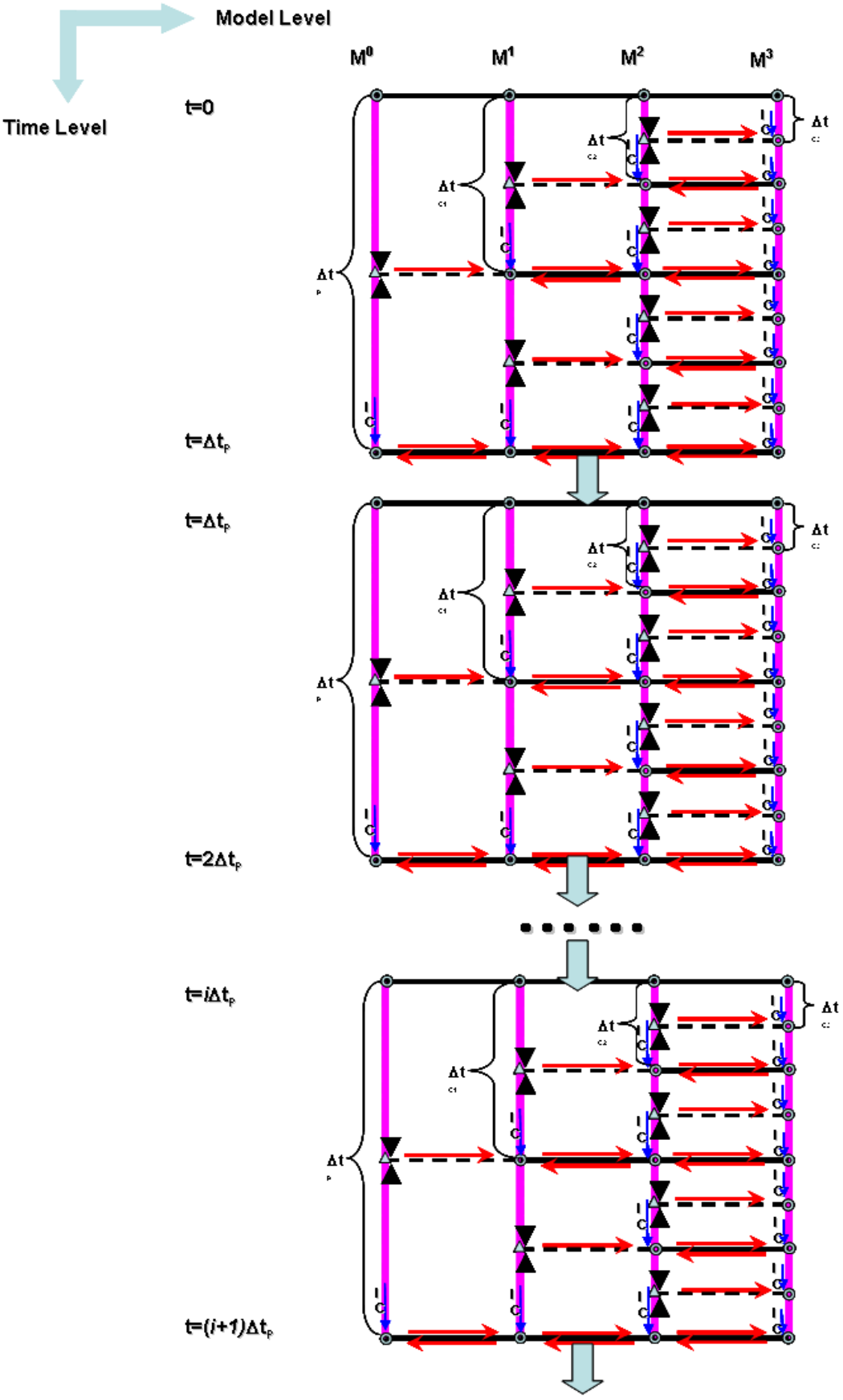


Figure 7. Flow chart of LIUA in hierarchical modeling to resolve multiple temporal scales

2 Discretization

2.1 Numerical Scheme

Solving Eq(1) or Eq(3) numerically needs discretizing the PDE in a computational domain to form a linear algebra system with head at each discretized node as unknown. In our multiscale modeling system, computational domain is discretized with no gap and not overlapping brick (3D) or rectangular (2D) cells and node at center of the cell as shown in Fig.8. The PDE would be approximated in each cell by using finite volume method (FVM) and then yields a nodal based discretized equation.

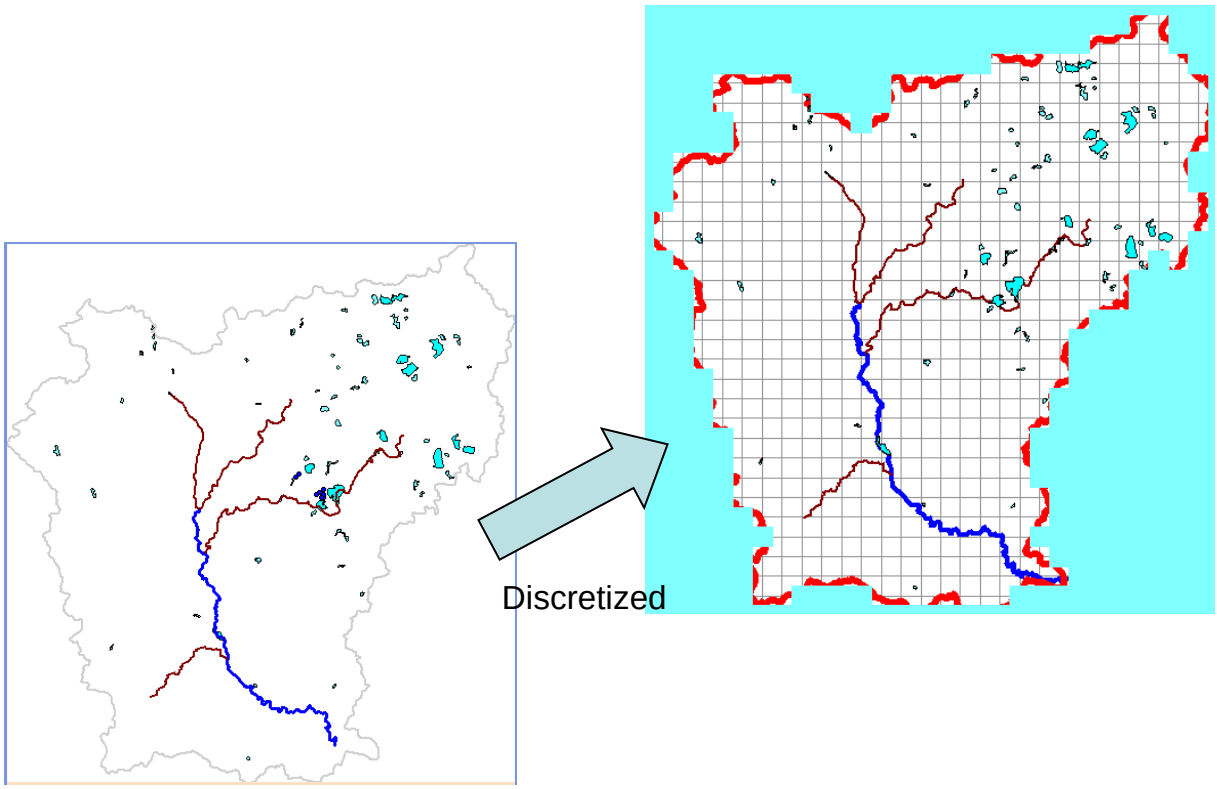


Figure 8. Domain discretization

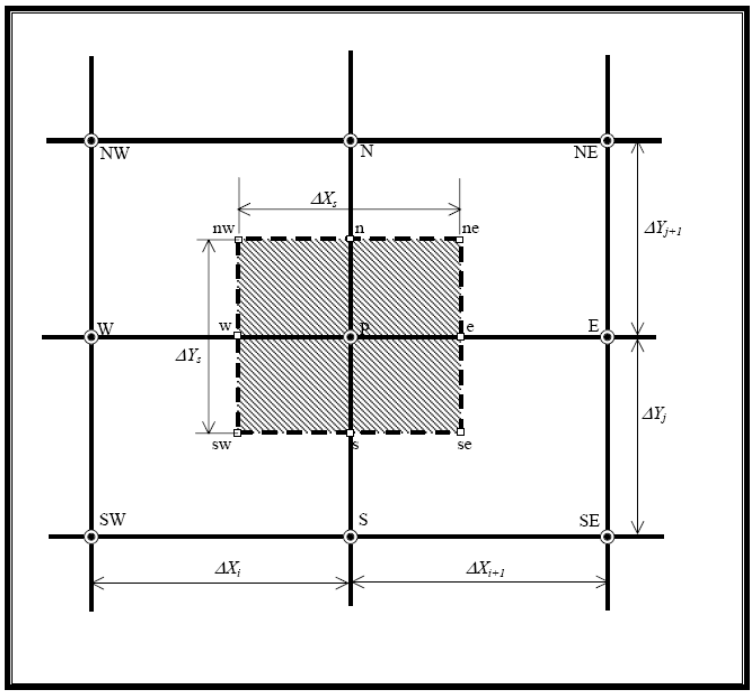
Fig.9 shows a typical cell of node P and its neighboring nodes E, S, W, N, T, B. Lines connecting node P and its neighboring nodes E, S, W, N, T, B have intersection with cell faces at face node e, s, w, n, t, b respectively.

Taking integral from Eq(1) over the computational domain, V, gives

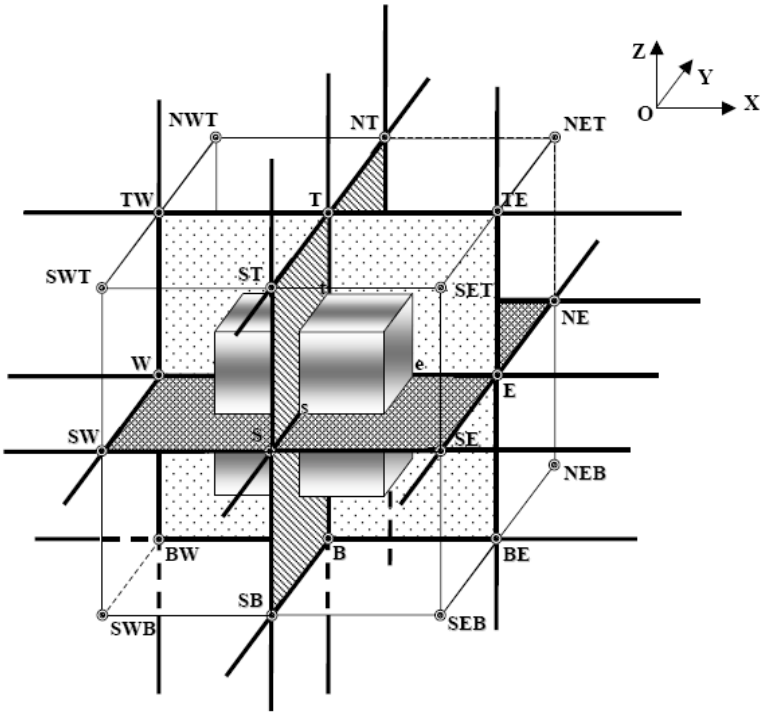
$$\iiint_V S_s \frac{\partial H}{\partial t} dV = \iiint_V \nabla \bullet (K \nabla H) dV + \iiint_V q dV \tag{4}$$

Dividing V into different N cells, Eq(4) can be rewritten as

$$\sum_{e}^{N} \iiint_{\Delta V_e} S_s \frac{\partial H}{\partial t} dV = \sum_{e}^{N} \iiint_{\Delta V_e} \nabla \bullet (K \nabla H) dV + \sum_{e}^{N} \iiint_{\Delta V_e} q dV \qquad (5)$$



2D Cell



3D Cell

Figure 9. Cell and Nodes

where ΔV_e is volume of 3D cell e, which is bounded and closed by cell faces E, S, W, N, T, B; or volume of 2D cell e with unit thickness in TB direction. There would be only cell faces E, S, W, N in a 2D cell.

For each cell e, from Eq(5), gives

$$\iiint_{\Delta V_e} S_s \frac{\partial H}{\partial t} dV = \iiint_{\Delta V_e} \nabla \bullet (K \nabla H) dV + \iiint_{\Delta V_e} q dV \tag{6}$$

Applying the divergence theorem, also known as Gauss-Ostrogradsky theorem to the first term of right hand side in Eq.(6), Eq.(6) becomes

$$\iiint_{\Delta V_e} S_s \frac{\partial H}{\partial t} dV = \oiint_{S_e} K \nabla H \bullet d\vec{S} + \iiint_{\Delta V_e} q dV \tag{7}$$

where Se in the surface integral term is denoting cell face in cell e, $d\vec{S} = \vec{n} dS$ is an area vector of a cell face with the same direction as its outward normal vector and magnitude of the area dS. Assuming that quantities are constantly distributed within each cell face and represented by the nodal value, Eq(7) can be rewritten as

$$(S_s \frac{\partial H}{\partial t})_P \iiint_{\Delta V_e} dV = \sum_{S_e} (K \nabla H)_{Se} \bullet d\vec{S}_e + (q)_P \iiint_{\Delta V_e} dV \tag{8}$$

Or

$$\begin{aligned} (S_s \frac{\partial H}{\partial t})_P \Delta V_e = {} & \Delta S_e K_e \left. \frac{\partial H}{\partial n} \right|_e - \Delta S_w K_w \left. \frac{\partial H}{\partial n} \right|_w + \\ & \Delta S_n K_n \left. \frac{\partial H}{\partial n} \right|_n - \Delta S_s K_s \left. \frac{\partial H}{\partial n} \right|_s + \\ & \Delta S_t K_t \left. \frac{\partial H}{\partial n} \right|_t - \Delta S_b K_b \left. \frac{\partial H}{\partial n} \right|_b + q_P \Delta V_e \end{aligned} \tag{9}$$

where ΔS_e, ΔS_w, ΔS_n, ΔS_s, ΔS_t and ΔS_b are area of the cell faces of e, w, n, s, t, and b ($\Delta V = \Delta S_e \Delta S_n \Delta S_t = \Delta S_w \Delta S_s \Delta S_b$). Accordingly, K_e, K_w, K_n, K_s, K_t and K_b are conductivities evaluated on the cell faces of e, w, n, s, t, and b respectively. Applying backward finite different scheme to the time variation term and implicit central scheme to those first order derivatives with respect to cell face normal directions on RHS of Eq(9), gives

$$\begin{aligned} S_s \Delta V_e \frac{H_P^{m+1} - H_P^m}{\Delta t} = {} & \Delta S_e K_e \frac{H_E^{m+1} - H_P^{m+1}}{\delta_{EP}} - \Delta S_w K_w \frac{H_P^{m+1} - H_W^{m+1}}{\delta_{PW}} + \\ & \Delta S_n K_n \frac{H_N^{m+1} - H_P^{m+1}}{\delta_{NP}} - \Delta S_s K_s \frac{H_P^{m+1} - H_S^{m+1}}{\delta_{PS}} + \\ & \Delta S_t K_t \frac{H_T^{m+1} - H_P^{m+1}}{\delta_{TP}} - \Delta S_b K_b \frac{H_P^{m+1} - H_B^{m+1}}{\delta_{PB}} + q_P \Delta V_e \end{aligned} \tag{10}$$

where δ_{ij} is the distance from node *i* to node *j*, m is denoted as time level, and Δt is the time step. If q_P is head dependent source/sink, then it can be linearized as:

$$q_P = -S_p H_P^{m+1} + S_C \tag{11}$$

where S_p and S_C are the slope and the intercept of the linearization respectively.
Rearranging Eq(10) and substituting Eq(11) into Eq(10), gives

$$A_P H_P^{m+1} + A_E H_E^{m+1} + A_W H_W^{m+1} + A_N H_N^{m+1} + A_S H_S^{m+1} + A_T H_T^{m+1} + A_B H_B^{m+1} = S_r \tag{12}$$

Where

$$\begin{cases} A_E = -\dfrac{\Delta S_e K_e}{\delta_{EP}},\ \mathrm{A_W} = -\dfrac{\Delta S_w K_w}{\delta_{PW}} \\ \mathrm{A_N} = -\dfrac{\Delta S_n K_n}{\delta_{NP}},\ \mathrm{A}_S = -\dfrac{\Delta S_s K_s}{\delta_{PS}} \\ A_T = -\dfrac{\Delta S_t K_t}{\delta_{TP}},\ A_B = -\dfrac{\Delta S_b K_b}{\delta_{PB}} \\ A_P = -(A_E + A_W + A_N + A_S + A_T + A_B + S_P \Delta V_e + \dfrac{S_s \Delta V_e}{\Delta t}) \\ S_r = S_C \Delta V_e + \dfrac{S_s \Delta V_e}{\Delta t} H_P^m \end{cases} \tag{13}$$

Application of Eq.(12) to each cell in the flow domain results in a system of linear equations, which would be a septem-diagonal matrix:

$$\begin{bmatrix} A_P^1 & A_E^1 & A_N^1 & A_T^1 & & & & & & & & & \\ A_W^2 & A_P^2 & A_E^2 & A_N^2 & A_T^2 & & & & & & & & \\ A_S^3 & A_W^3 & A_P^3 & A_E^3 & A_N^3 & A_T^3 & & & & & & & \\ A_B^4 & A_S^4 & A_W^4 & A_P^4 & A_E^4 & A_N^4 & A_T^4 & & & & & & \\ & A_B^5 & A_S^5 & A_W^5 & A_P^5 & A_E^5 & A_N^5 & A_T^5 & & & & & \\ & & A_B^6 & A_S^6 & A_W^6 & A_P^6 & A_E^6 & A_N^6 & A_T^6 & & & & \\ & & & & & & \cdots\cdots & & & & & & \\ & & & & & & & & \cdots\cdots & & & & \\ & & & & A_B^{N-3} & A_S^{N-3} & A_W^{N-3} & A_P^{N-3} & A_E^{N-3} & A_N^{N-3} & A_T^{N-3} & & \\ & & & & & A_B^{N-2} & A_S^{N-2} & A_W^{N-2} & A_P^{N-2} & A_E^{N-2} & A_N^{N-2} & & \\ & & & & & & A_B^{N-1} & A_S^{N-1} & A_W^{N-1} & A_P^{N-1} & A_E^{N-1} & & \\ & & & & & & & A_B^N & A_S^N & A_W^N & A_P^N & & \end{bmatrix} \begin{Bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \\ H_5 \\ H_6 \\ \vdots \\ \vdots \\ H_{N-3} \\ H_{N-2} \\ H_{N-1} \\ H_N \end{Bmatrix} = \begin{Bmatrix} Sr_1 \\ Sr_2 \\ Sr_3 \\ Sr_4 \\ Sr_5 \\ Sr_6 \\ \vdots \\ \vdots \\ Sr_{N-3} \\ Sr_{N-2} \\ Sr_{N-1} \\ Sr_N \end{Bmatrix} \tag{14}$$

Or

$$[\mathbf{A}]\{\mathbf{H}\} = \{\mathbf{Sr}\} \tag{15}$$

where $[A]$ is a square symmetric positive definite matrix consisting of the coefficients $A_P, A_E, A_W, A_N, A_S, A_T$ *and* A_B of Eq.(12), $\{H\}$ consists of the unknown hydraulic head values for current time step , and $\{Sr\}$ is the forcing vector consisting of known values from the previous time step and given fluxes.

The linear system of equations Eq.(14) is solved using matrix solver with given BCs. As shown in Fig.10, there are 12 matrix solvers available in IGW including the Algebraic Multigrid (AMG), the Successive Over the Relaxation (SOR), the Conjugate Gradient (CG), the Conjugate Gradient with Normal Residual Equations, the Biconjugate Gradient (BCG), the Full Orthogonalization, Biconjugate Gradient with Partial Pivoting, the Biconjugate Gradient Stablized, the Transpose Free Quasi-Minimum Residual, Generalized Minimum Residual, Flexible Generalized Minimum Residual and the Direct Quasi-Generalized Minimum Residual.

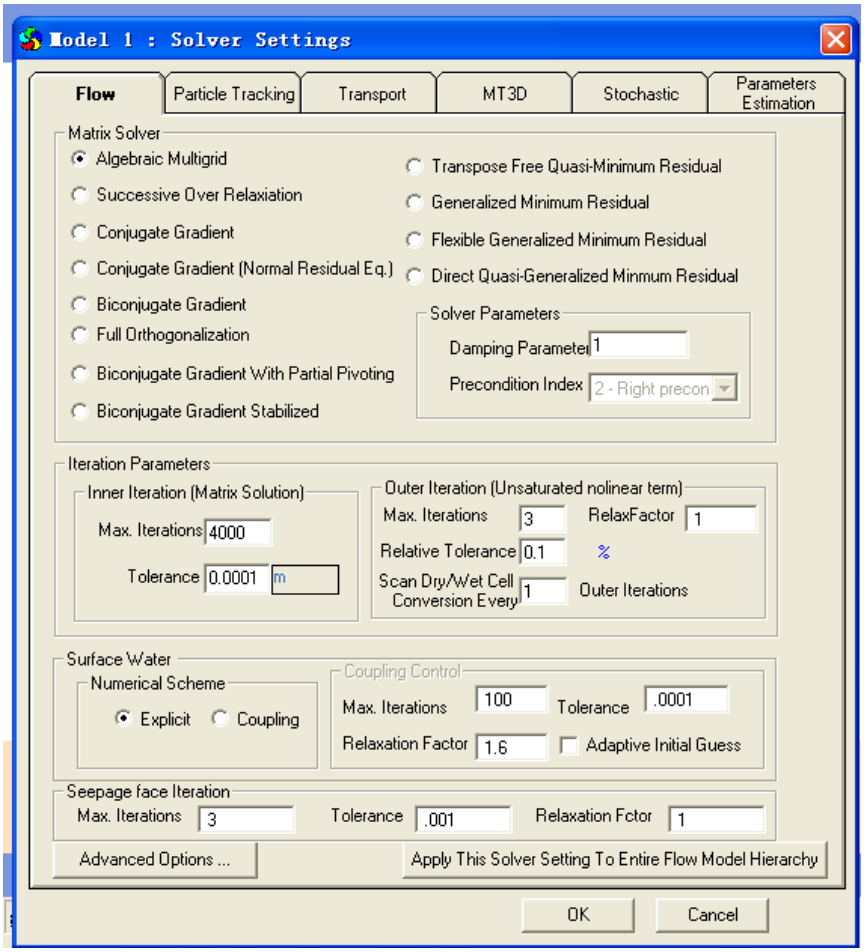


Figure 10. Matrix solver available in IGW

2.2 Grid Layout Design in Parent-Child Models

As mentioned above, in the hierarchical modeling, information propagates in both down and up scaling directions through interfaces of parent-child models. Information propagating from parent model to child model is a process of how to pass coarse grid information to fine grid. In the contrast, information propagating from child model to parent model is that of how to pass fine grid information to coarse grid. Therefore, well-designed grid layout (including temporal gridding) can make information to be propagated more accurately, efficiently and reasonably. Interpolation schemes including both spatial and temporal are also grid layout dependent.

Shared node based grid layout is one of the most efficient grid system in grid refinement and nested modeling. Shared nodes are those nodes that shared by both parent and child models such as node A, B, C and D in Fig.11. Nodes A' and B' are not shared nodes. Connecting shared nodes on the child model boundary will form the interfaces of the parent and child models such as line AB in Fig.11, that is, boundaries of child models are part of grid lines of parent model or some of grid lines of parent model will be the boundaries of child models. This kind of grid configuration can greatly simplify the interpolation efforts, which will be used very intensively in down and up scaling iterative loop in the hierarchical modeling and therefore can save computational time considerably.

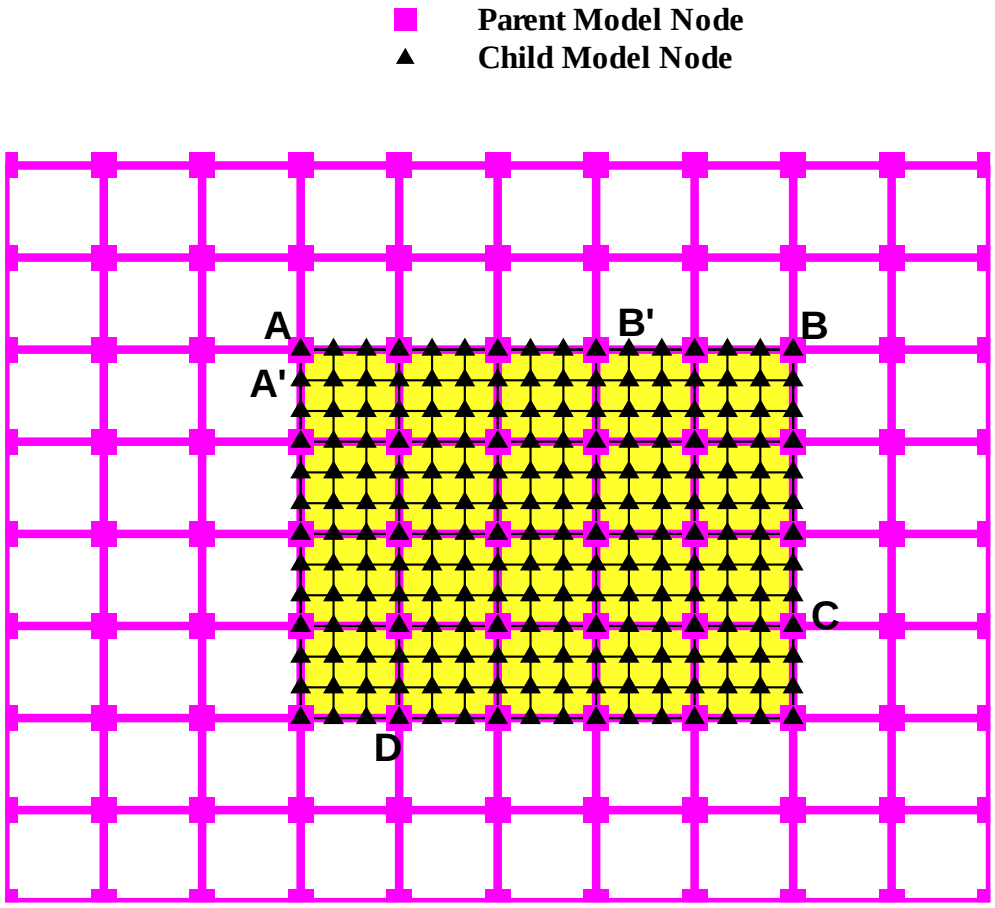


Figure 11 Shared node grid layout in a hierarchical modeling system (Thicker solid lines are parent grid lines, thiner solid lines are child grid lines)

Another grid layout called shared cell faces grid system is especially good for flux calculation and interpolation. In this system, boundaries of child model will be part of the cell faces of the parent model or some of the cell faces will be boundaries of the child models. Fig.12 illustrates one of this kind of grid system. This kind of grid system, however, may cause difficulty in nodal head calculation if a prescribed head boundary condition is needed along the interfaces. In our multiscale modeling system, shared node based grid layout is used and the following context related to grid system would be referred to as shared node grid system.

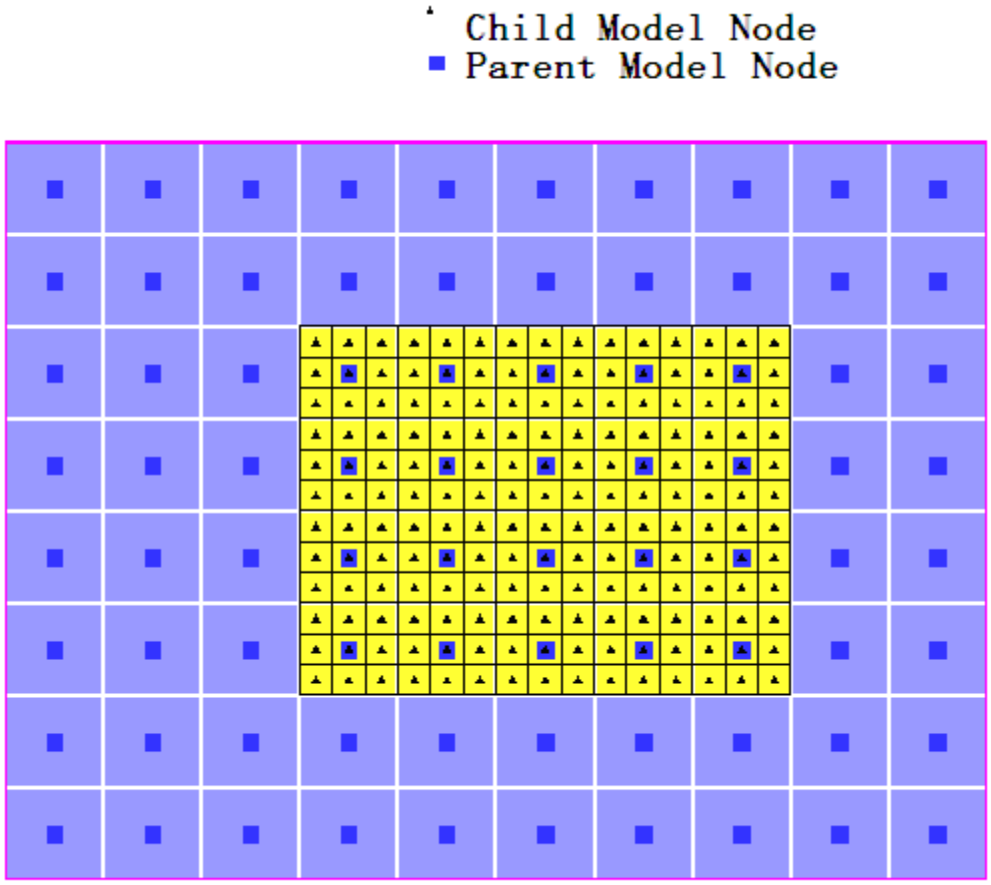


Figure 12 Shared Cell Face grid system

2.3 Boundary Condition Propagation in Parent and Child Models

Usually, child models contain finer mesh spacing and smaller time step than the parent model. The function of the child model is to simulate phenomena that require a finer grid than the parent model contains, such as sharp changes in hydraulic gradient, abrupt changes in hydraulic properties that would otherwise be smeared by representation on the parent grid. The role of down scaling from parent to child is to provide boundary conditions to the child model that are consistent with the regional flow system; that of up scaling from child model to parent model is to provide a feedback to the parent model that its aggregated feature is consistent with the details resolved in child model. The coupling between the two grids occurs via boundary conditions at the interface between the parent model and its child models. Boundary conditions along the interfaces of parent-child models can be in the form of a prescribed head (Type 1) or a prescribed flux (Type 2) as mentioned in Eq(3) and as shown in Fig.13. Therefore, combinations of boundary conditions in the parent-child models could be

one of the following: 1) prescribed head in parent model and prescribed head in child model (H-H), 2) prescribed head in parent model and prescribed flux in child model (H-F), 3) prescribed flux in parent model and prescribed head in child model (F-H), 4) prescribed flux in parent model and prescribed flux in child model (F-F). The following sections will describe the approaches to obtain heads or fluxes boundary conditions in either parent model derived from child model or child model derived from parent model.

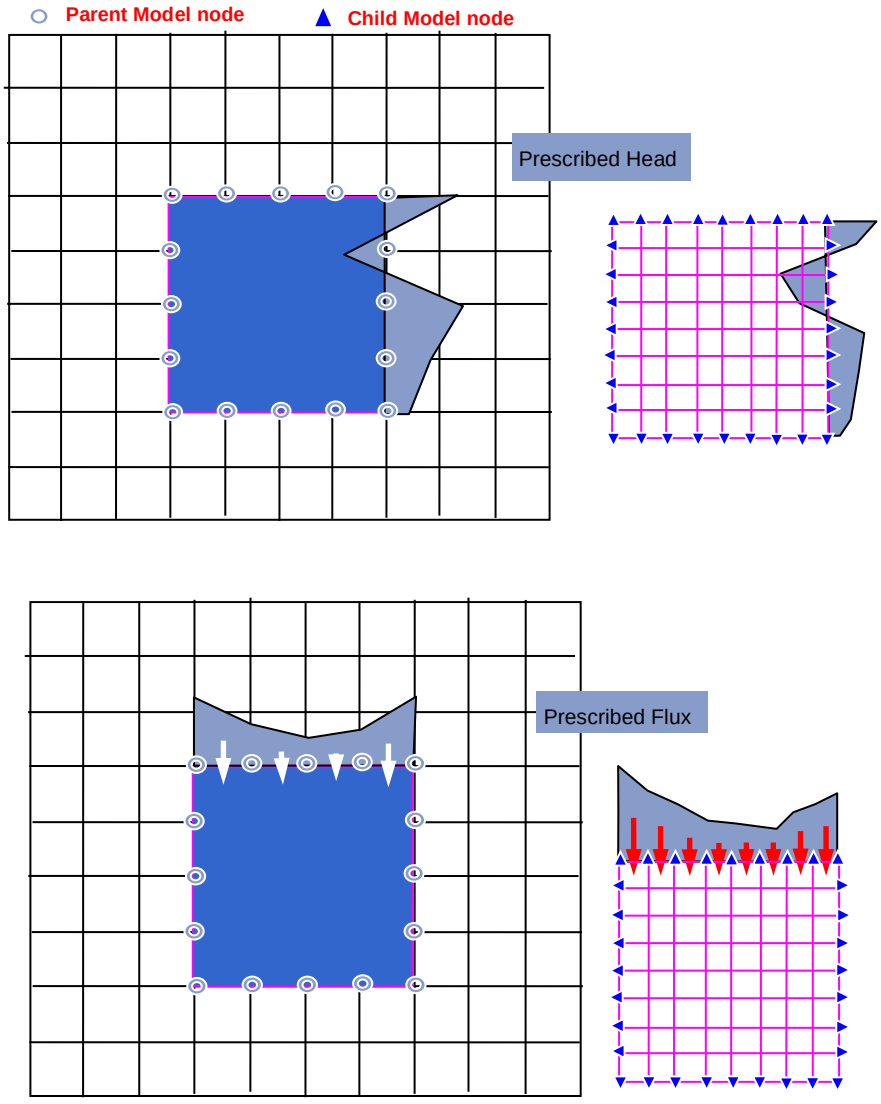


Figure 13 Boundary Conditions in Multiscale Modeling

2.4 Parent Specified Head Boundary Conditions

Head in our multiscale modeling system is referred to here as nodal head. Given heads in child model and under a shared node based grid layout, head values at the parent model 's grid nodes are very easily calculated.

As shown in Fig.14, head distribution along one segment of shared grid line in the parent model, which is also part of the child model boundaries, is known once upscaling starts – heads at triangle nodes in Fig.14 are known, denoted as H_c. As mentioned above, shared nodes denoted as circles in Fig.14 are shared by both the parent and child models, which means heads at shared nodes are either heads of the parent model denoted as H_P or those of

the child model, H_c. Therefore, as illustrated in Fig.11, head value at a parent model shared node can be calculated by just simply assigning the child model head to the parent model head at the same shared node, that is,

$$H_P = H_C \tag{16}$$

Eq(16) will be applied to each shared node in the parent model, then a specified head (Dirichlet) boundary condition for the parent model is resulted.

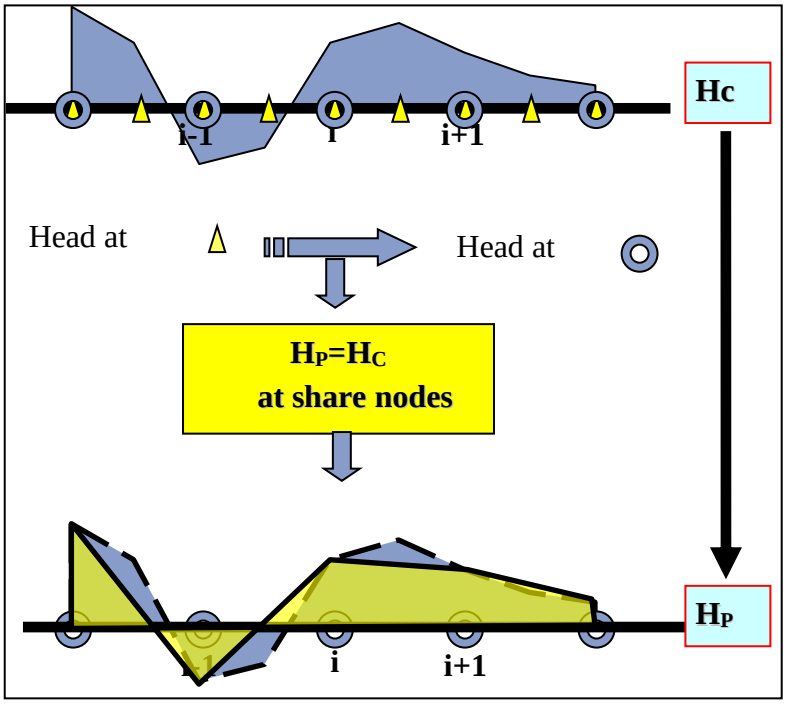


Figure 14 specified head boundary conditions calculated from child model
(Triangles are child model nodes, circles are parent model nodes and shared nodes)

2.5 Parent Specified FLux Boundary Conditions

Flux in our multiscale modeling system is referred to here as "flux across a surface". To derive the specified flux boundary condition along the parent-grid interface, flux balance on interface is required: the net flow across the interfacing boundary from parent model side equals that from child model side. Fig.15 shows a typical parent model cell represented by darker shading and its three bordering child model cells represented by lighter shading with index of *i-1*, i and *i+1*.

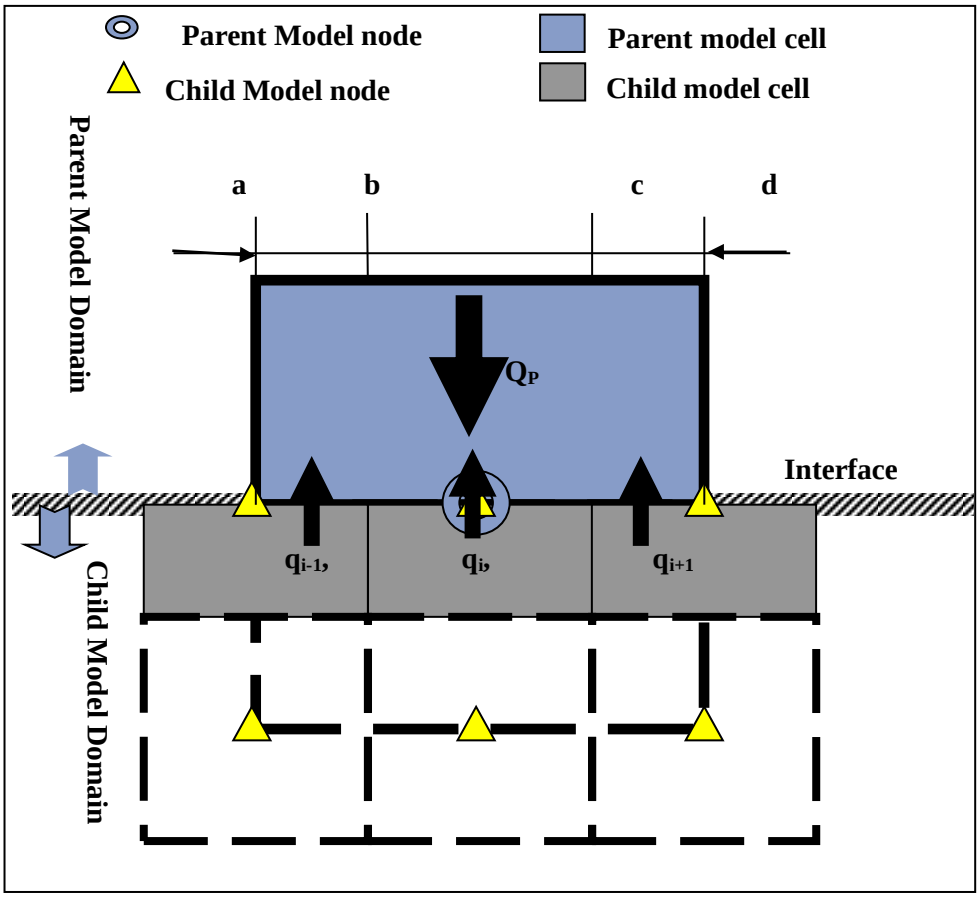


Figure 15 Mass balance on the interfacing boundary of one cell face of parent model and three bordering cells of a child model

From Fig.15, the net flow going into the parent cell is equal to the sum of the fluxes going out of the child model cells: q_{i-1}, q_i and q_{i+1}, that is,

$$Q_P = -(q_{i-1} + q_i + q_{i+1}) \tag{17}$$

where q_{i-1} or q_{i+1} represents the flux across half cell face of the child model, q_i represents the flux across a full cell face of the child model.

Given the heads in the child model, from Fig.16, q_i can be expressed in the following form:

$$q_i = q_{x1} + q_{x2} + q_y \tag{18}$$

Or

$$q_i \approx -K_e B \frac{H_E - H_N}{2\delta_{EN}} \Delta Y_C - K_w B \frac{H_N - H_W}{2\delta_{NW}} \Delta Y_C - K_s B \frac{H_N - H_S}{\delta_{NS}} \Delta X_C \tag{19}$$

where H_E, H_N, H_W and H_S are nodal head at node E, N, W and S in the child models; K_e, K_w, K_s are the cell face conductivities on cell face e, w, and s respectively; B is the thickness of the cell, δ_{ij} represents the distance between node i and node j and all other symbols were denoted in Fig.16.

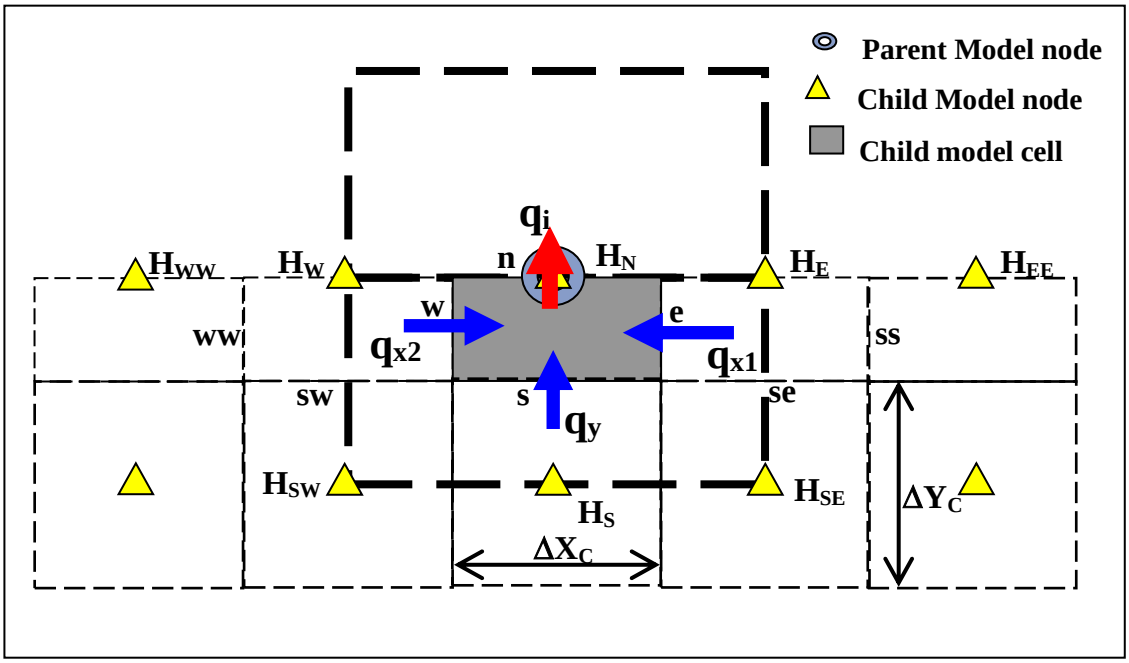


Figure 16 Calculation of a full cell flux in the child model

In similar, from Fig.16, q_{i-1} and q_{i+1} can be expressed in the following forms:

$$q_{i-1} \approx -K_{ee}B\frac{H_{EE}-H_E}{4\delta_{EEE}}\Delta Y_C - K_eB\frac{H_E-H_N}{4\delta_{EN}}\Delta Y_C - K_{se}B\frac{H_E-H_{SE}}{2\delta_{ESE}}\Delta X_C \quad (20)$$

$$q_{i+1} \approx -K_wB\frac{H_N-H_W}{4\delta_{NW}}\Delta Y_C - K_{ww}B\frac{H_W-H_{WW}}{4\delta_{WWW}}\Delta Y_C - K_{sw}B\frac{H_W-H_{SW}}{2\delta_{WSW}}\Delta X_C \quad (21)$$

where H_{EE}, H_{WW}, H_{SW} and H_{SE} are nodal head at node EE, WW, SW and SE ; K_{ee}, K_{we}, K_{se} and K_{sw} are the cell face conductivities on cell face ee, we, se and sw respectively, all other symbols were denoted in Fig.16.

Application of Eq(17) to each interfacing cell in parent model yields a specified flux (Neuman) boundary condition for the parent model.

2.6 Child Specified Head Boundary Conditions

To define the specified head boundary conditions along the interface of child model, head values need to be consistent with parent model, which means these head values would be derived from parent model.

As shown in Fig.17, head distribution along one segment of shared grid line, which is either part of child model boundaries or grid line of parent model, is known once downscaling starts – heads at circle nodes (parent nodes) in Fig.17 are known, denoted as H_P. These shared nodes are shared by both parent and child model, which means heads at shared nodes are either heads of parent model or those of child model, H_c. Therefore, for the nodes that are shared as with the parent, heads calculated by the parent model apply directly, that is,

$$H_C = H_P \quad (22)$$

For the child nodes along the interface that do not share the same location with a parent

node, values of head need to be interpolated using the values at shared nodes. If the child grid size is a factor of parent model ($n = \frac{\Delta X_P}{\Delta X_C}$), for example, n=2 as shown in Fig.16, then head values at non-shared nodes can be easily calculated in the following form by using linear interpolation scheme:

$$H_j = \frac{j}{n} H_{i+1} + (1 - \frac{j}{n}) H_i \quad (j = 1, n-1) \tag{23}$$

where H_j is the head at non-shared node between the ith shared node and the (i+1)th shared node, H_{i+1} and H_i are heads at the ith shared node and the (i+1)th shared node, at which their values are known. For example, n=2 means there is only one non-shared node between two shared nodes, from Eq(23) and as shown in Fig.16, its head value, $H_{i+1/2}$, can be given as

$$H_{i+1/2} = \frac{H_{i+1} + H_i}{2} \tag{24}$$

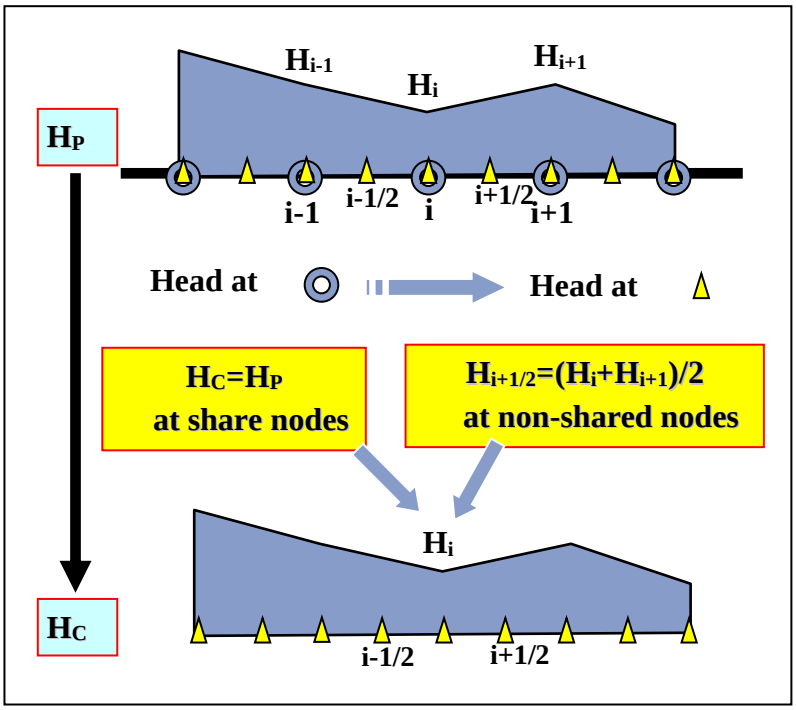


Figure 17 specified head boundary conditions calculated from parent model
(triangles are child model nodes, circles are parent model nodes and shared nodes)

Similarly, if a quadratic interpolation method is favorite, three shared nodes, H_{i+1}, H_i and H_{i-1} should be used to calculate all the non-shared nodes within node (i-1) to (i+1). If $n = \frac{\Delta X_P}{\Delta X_C}$, then the quadratic interpolation scheme can be written as

$$H_j = \frac{1}{2n^2}\left[(j-n)(j-2n)H_{i-1} - 2j(j-2n)H_i + j(j-n)H_{i+1}\right], \quad (j = 1,2n-1) \tag{25}$$

If the conductivity is spatially variable over the computational domain, a Darcy weighted interpolation scheme my be used as first suggested by Wasserman and later developed by Mehl and Hill.

Application of Eq(22) and Eq(23) or Eq(25) to each non-shared node in the child model yields a specified head (Dirichlet) boundary condition for the child model.

2.7 Child Specified Flux Boundary Conditions

As noted previously and seen clearly from the parent model flux boundary calculation, defining parent model specified flux boundary condition is a process of summing up fluxes from the child model cells. In the contrast and as to be seen later on, defining child model specified flux boundary condition will be a process of allocating fluxes from parent model cells to child model cells.

Given the heads in the parent model, from Fig.18, flux across the parent grid face AB can be expressed in the following form:

$$Q_P \approx -K_n B \frac{H_N - H_S}{\delta_{NS}} \Delta X_P \tag{26}$$

where δ_{NS} is the distance between node N and Node S in the parent model.

In order to maintain a mass balance on the grid face AB, Q_p should be distributed among those child model cell faces that shared with AB, for example, ab, bc and cd in Fig.18. The simplest way to allocate Q_p to the fluxes across child cell faces is to distribute it in an area weighted way, which can be written as

$$q_i = \frac{A_i}{\sum_{j=1}^{N_f} A_j} Q_P \quad (i = 1, N_f) \tag{27}$$

where A_i is the area of the *i*th cell face that shared with parent grid face AB, q_i is the flux across the *i*th cell face with area of A_i and N_f is the total number of cell faces that shared with parent grid face AB . From Eq(27), as an example in Fig.18, fluxes q_{i+1}, q_i and q_{i-1} can be obtained in the following forms

$$\begin{cases} q_i = \dfrac{Q_P}{2} \\ q_{i+1} = \dfrac{Q_P}{4} \\ q_{i-1} = \dfrac{Q_P}{4} \end{cases} \tag{28}$$

Application of Eq(27) to each shared cell face in the child model yields a specified flux (Neuman) boundary condition for the child model.

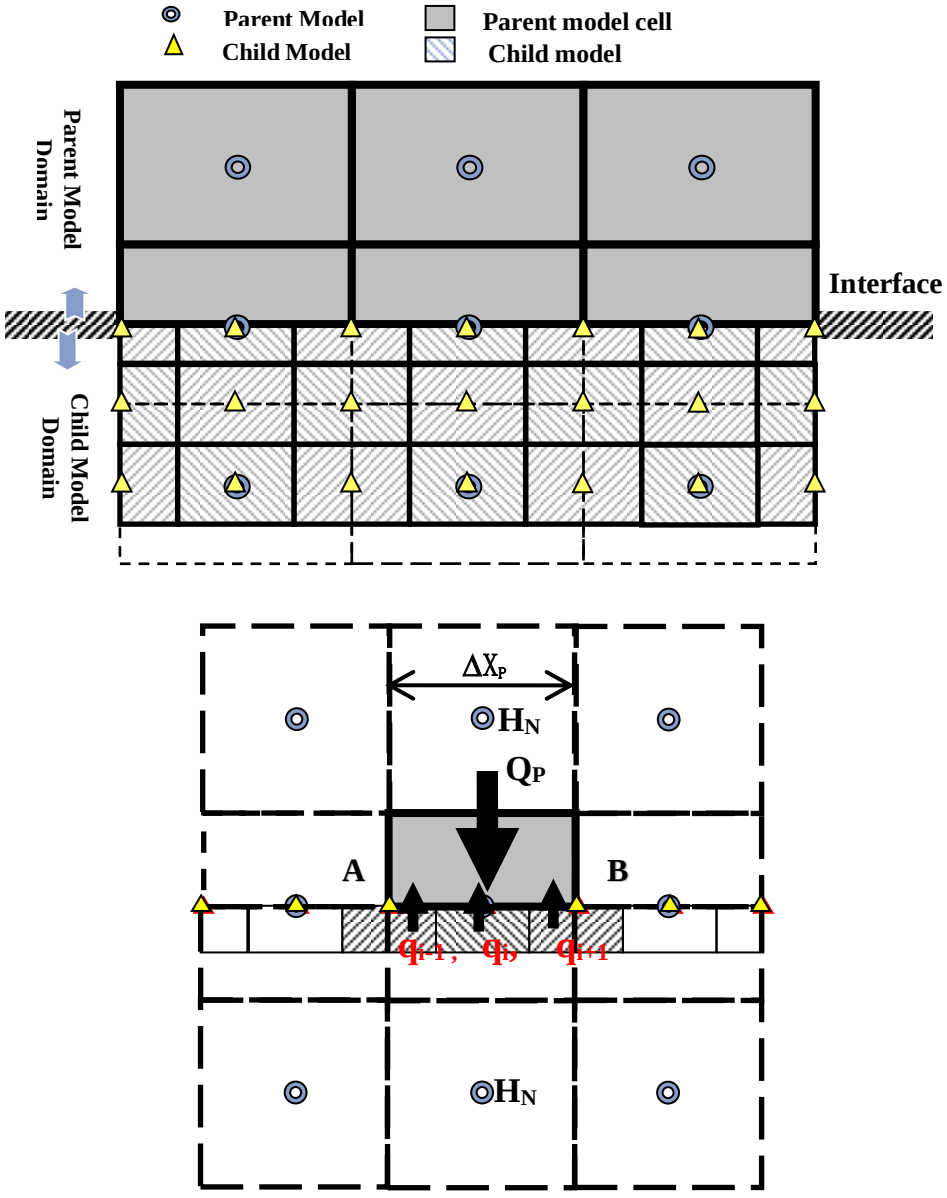


Figure 18 Calculation of a cell flux for child model

2.8 Transient Boundary Conditions

As mentioned previously, in a nested time stepping algorithm, temporal interpolation will be needed to obtain heads at the nested time step of child models. Temporal interpolation is not only applied to heads along the interfaces of parent-child models, but also over the whole computational domain of a child model since the initial conditions for nested time steps is provided by parent model in which time step is larger than its child model's nested time step. If a "shared time nodes" scheme is used in designing the nested time step system in a multiscale modeling and a nested time step of the child is a factor of parent model's time step ($n = \frac{\Delta t_P}{\Delta t_C}$), for example, n=2 as shown in Fig.19, then head values at shared time nodes and non-shared nodes would be easily calculated in the similar way as done for shared nodes based spatial grid layout.

It should be noted that all temporal interpolations will be carried on individual model and

the interpolated information, either from parent model or child model, then are propagated between parent and child models via the boundary conditions as mentioned above and initial conditions.

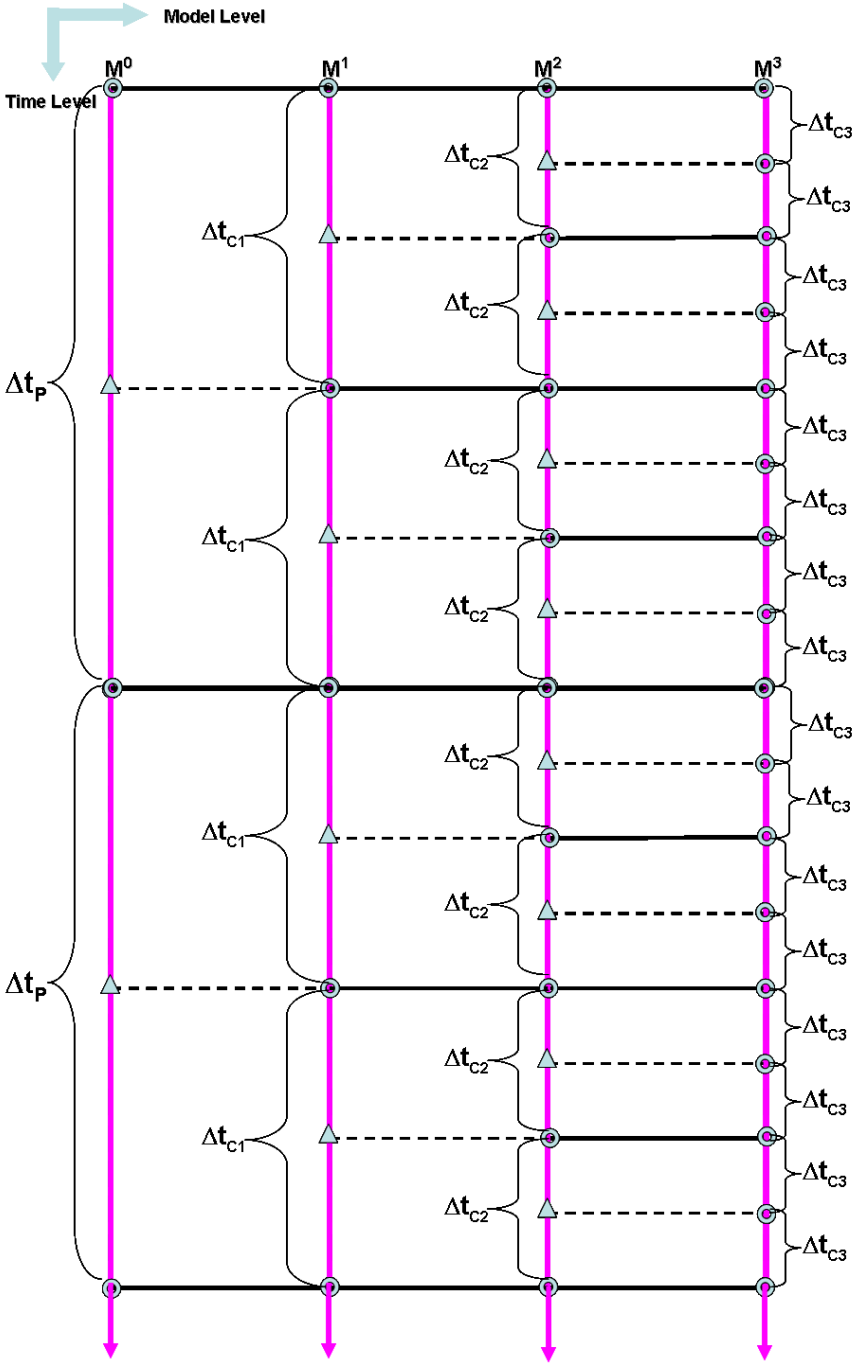


Figure 19 Nested time stepping "shared time nodes" (nodes on solid lines) and "non-shared time nodes" (nodes on dash lines) with $\Delta t_P = 2\Delta t_{C1} = 4\Delta t_{C2} = 8\Delta t_{C3}$

(1) Initial Conditions

Initial conditions for every time step in a child model will be heads at previous time step in the same model. Initial conditions for a time step in a parent model will be heads at previous time step in the same model either obtained directly by solving PDE if the previous time level is a non-shared time node or updated from its child models by mean of upscaling procedure if this previous time level is at a 'shared time node'.

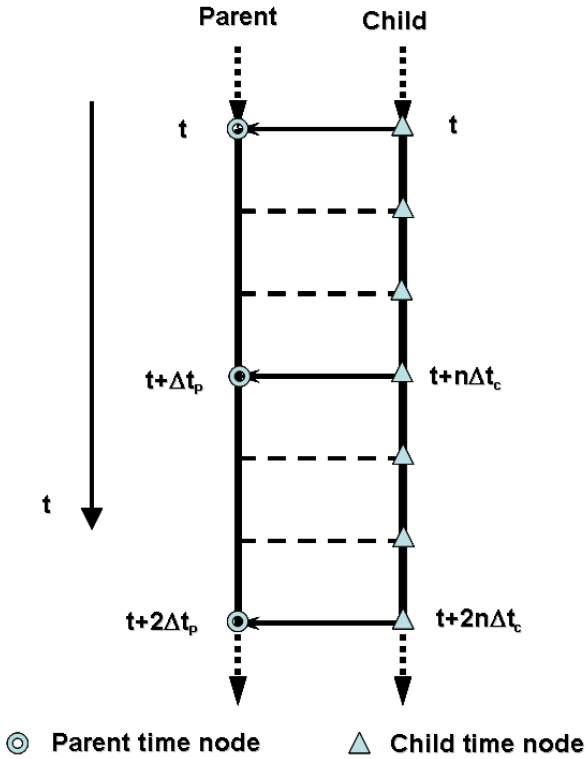


Figure 20 Nested time stepping: parent time nodes and child time nodes

Shown in Fig.20 is a parent-child temporal grid system in which circles denote parent time grid nodes and triangles denote child time grid nodes. Once head values at the triangles are calculated, values of head at the circles are obtained by directly assigning values of head at the triangles to head values at the circles or spatially averaging those heads at the triangles that are parent node's neighboring nodes as shown in Fig.21.

$$H_P(t+\Delta t_P) = H_C(t+n\Delta t_C)$$
$$or \qquad (29)$$
$$H_P(t+\Delta t_P) = \frac{1}{Nb}\sum_i H_i^C(t+n\Delta t_C)$$

where H_P is the head at the pth node in the parent model at time level $t+\Delta t_P$, H_i is the same time level head at the ith node in the child model that is the neighboring node of node P, Nb is the total number of neighboring nodes around node P.

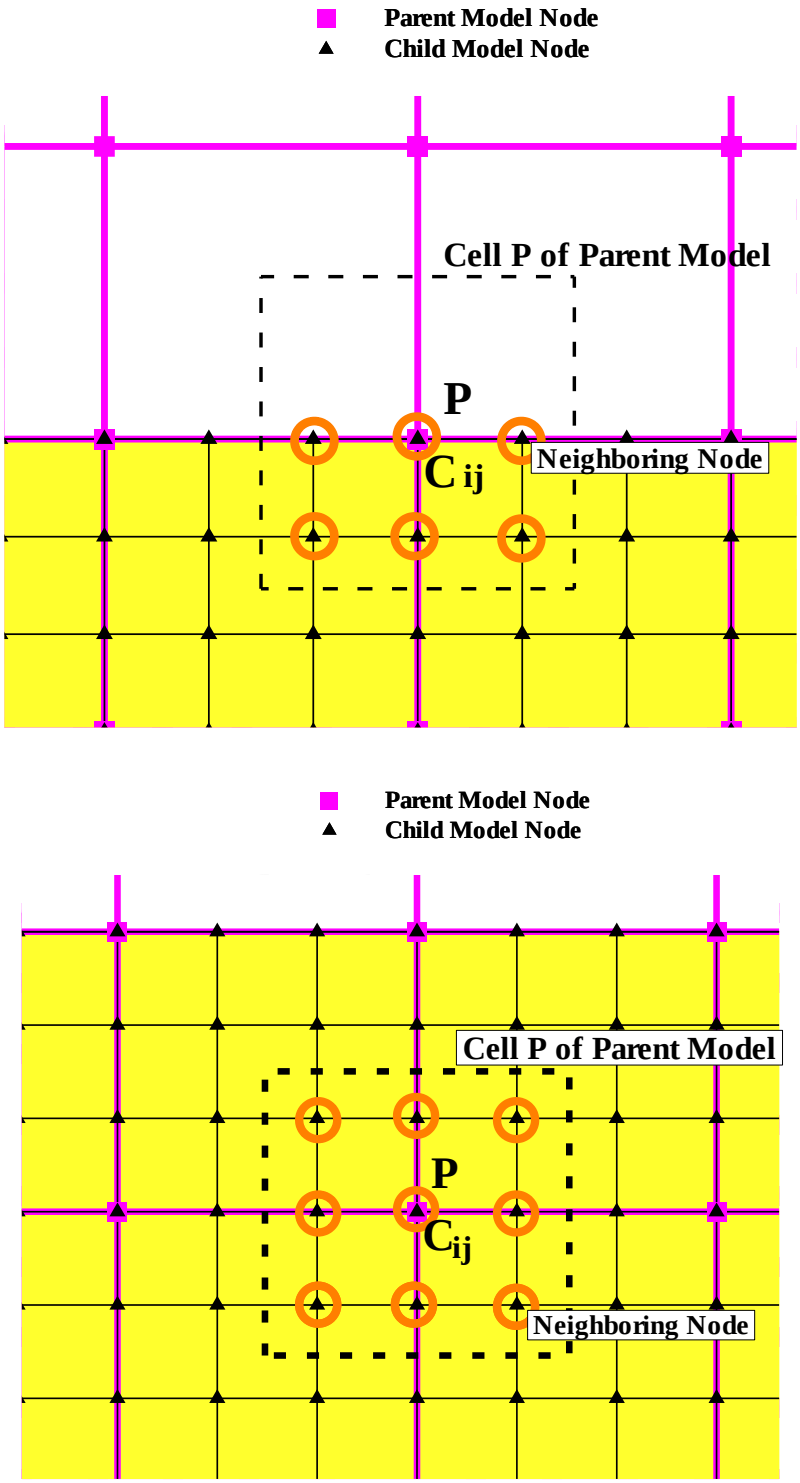


Figure 21 Head Averaging at P over Neighboring Child Cells

(2) Boundary conditions updating along "shared time nodes" path lines

In this case, both the parent and child models are at the same time level and it is not necessary to do any temporal interpolation. Spatial interpolation will be required when to define updated boundary conditions from either parent model or child model. This has been described in the previous sections already.

(3) Boundary conditions updating at the "non-shared time nodes"

As seen in Fig.6 and Fig.19, dash lines actually are the extension of those solid lines with missing 'current' information from their parent. Function of dash lines is to provide missing current information from parent model at the broken (non-shared) nested time step of a child model so that calculation in child models can be able to advance to the next nested time step. For example, let's assume that time step in a parent model is Δt_P and Δt_C is time step in a child model. The calculation starts from time level t and advances to time level $t+\Delta t_P$. If there are 3 nested time steps in one Δt_P, which means $\Delta t_P = 3\,\Delta t_C$, the nesting structure can be seen in Fig.20.

It is seen clearly that to advances to time level $t+\Delta t_P$ from time level t in the child model needs go through time level t , $t+\Delta t_c$, $t+2\Delta t_c$ and $t+3\Delta t_c$ while only t , $t+\Delta t_P$ in parent model. Transient equation solving in child model will be stopped at time level $t+\Delta t_c$, $t+2\Delta t_c$ due to the fact that boundary conditions are missing. To circumvent this, head values along the interface of parent and child models would be interpolated temporally at time level $t+\frac{\Delta t_P}{3}$ (equals to $t+\Delta t_c$) and $t+\frac{2\Delta t_P}{3}$ (equals to $t+2\Delta t_c$) from heads at time level t and $t+\Delta t_P$ in parent model. If a linear interpolation method is used, then head values at time level $t+\Delta t_c$ and $t+2\Delta t_c$ in child model can be expressed as

$$\begin{aligned} H_C(t+\Delta t_C) &= \frac{2}{3}H_P(t)+\frac{1}{3}H_P(t+\Delta t_P) \\ H_C(t+2\Delta t_C) &= \frac{1}{3}H_P(t)+\frac{2}{3}H_P(t+\Delta t_P) \end{aligned} \tag{30}$$

Generally, if number of nested time steps of a child model is n, then their boundary conditions can be derived from the parent model in the following form

$$H_C(t+i\Delta t_C) = \frac{n-i}{n}H_P(t)+\frac{i}{n}H_P(t+\Delta t_P) \qquad (i=1,\ n-1) \tag{31}$$

Fig.22 shows the temporal interpolation graphically.

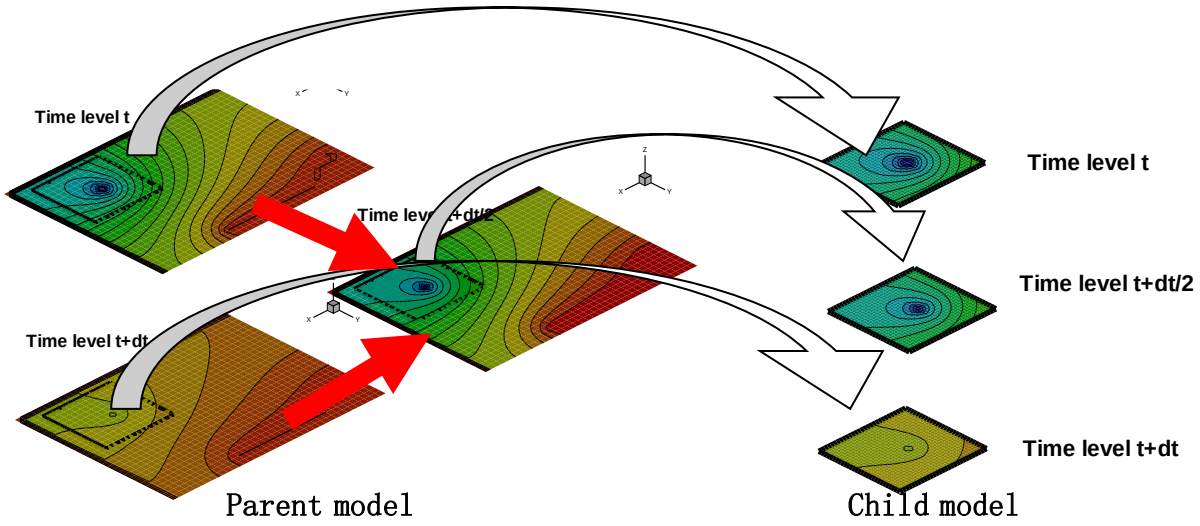


Fig.22 Temporal interpolation for time level t+dt/2 in parent model (red arrows) and updating boundary conditions for child model (white arrows)

2.9 Notes in Boundary Conditions

Because the coupling occurs through boundary conditions, which are accounted for in the right hand side of the matrix equations, both the parent and child models maintain a conventional stencil. Thus, the commonly available linear equation solvers developed for these regular stencils can be applied without any special consideration.

Relaxation Factor needed in coupling iteration loop

Flux boundary condition in both child and parent models + prescribed flux inside only may cause an unreasonable solution due to the fact that mass conservative law may be invalidated

Corner cell flux Calculation --- vertical face or horizontal face?

3 Interaction of surface water and groundwater

The mathematical model of interaction of surface and groundwater water is described by a system of two partial differential equations. These two governing equations must be solved in a coupling way due to the fact that the system simulated involves interaction terms between surface water and groundwater. It is noted that surface water body is referred to here as lakes, wetlands and reservoirs only.

(1) Groundwater governing equation

Groundwater flow is governed by Eq.(1), that is

$$
\begin{aligned}
&S_s \frac{\partial H^{GW}}{\partial t} = \nabla(K \bullet \nabla H^{GW}) - q_{GW} + q_{SW} \\
&q_{SW} = \sum_i^N (H_i^{SW} - H_i^{GW}) \frac{L_i}{B_i} \\
&H_i^{GW} = \begin{cases} H_i^{GW} & \text{if } H_i^{GW} > Elev_{\mathrm{b}} \\ Elev_{\mathrm{b}} & \text{if } H_i^{GW} < Elev_{\mathrm{b}} \end{cases}
\end{aligned}
\tag{32}
$$

Where H^{GW} is the groundwater head, [L] H_i^{GW} is the groundwater head in the ith cell that having interaction with surface water, [L], L_i and B_i are the leakancy and thickness of the interactive cell, [1/L] and [L], respectively, H_i^{SW} is the surface water head corresponding to its counterpart of groundwater head, H_i^{GW}, [L], $Elev_b$ is the bed elevation of surface water body (could be spatially variable), [L], q^{GW} is the groundwater source/sink terms, [L^3L^{-3}/T], q_{SW} is the incoming or outgoing discharge from surface water body, [L^3L^{-3}/T], N is the total number of surface water cells.

(2) Surface water governing equation

Lake water level H^{SW} is governed by the following continuity equation:

$$
\begin{aligned}
&\frac{\partial V_{SW}}{\partial t} = Q_{SW} - Q_{GW} \\
&\frac{\partial V_{SW}}{\partial t} = S_{SW} \frac{\partial H^{SW}}{\partial t} + H^{SW} \frac{\partial S_{SW}}{\partial t} \\
&Q_{GW} = \sum_i^N (H_i^{SW} - H_i^{GW}) L_i A_i \\
&H_i^{GW} = \begin{cases} H_i^{GW} & \text{if } H_i^{GW} > Elev_{\mathrm{b}} \\ Elev_{\mathrm{b}} & \text{if } H_i^{GW} < Elev_{\mathrm{b}} \end{cases}
\end{aligned}
\tag{33}
$$

Where V_{SW} is the storage volume of the water body, [L^3], S_{SW} is the water surface area, [L^2], A_i is the area of the ith interactive cell, [L], Q_{SW} is the incoming discharges, which

could be function of time and of water elevation, [L^3/T], Q_{GW} is the incoming or outgoing discharges from groundwater, [L^3/T]. Assuming S_{SW} is time independent (no deposit or scouring), then gives

$$S_{SW}\frac{\partial H^{SW}}{\partial t} = Q_{SW} - Q_{GW}$$
$$Q_{GW} = \sum_i^N (H^{SW} - H_i^{GW}) L_i A_i \qquad (34)$$
$$H_i^{GW} = \begin{cases} H_i^{GW} & \text{if } H_i^{GW} > Elev_{\text{b}} \\ Elev_{\text{b}} & \text{if } H_i^{GW} < Elev_{\text{b}} \end{cases}$$

Due to the fact that S_{SW} may be a function of water elevation, eq(34) has to be solved by using non-linear methods, such as Newton Raphson method.

Interaction of surface water and groundwater will be implemented through constantly updating the coupling term: q_{SW} in Eq(32) and Q_{GW} in Eq(34) in a non-linear way. Numerical schemes to approximate Eq(32) has been described in the previous sections. As to Eq(34), there are several schemes available and will be described below.

(3) Numerical Scheme to Approximate Surface Water Equation

1) Explicit Scheme

Applying backward finite difference scheme to the time derivative and explicit scheme to the coupling terms in RHS of Eq.(34), gives

$$S_{SW}\frac{H^{SW(n+1)} - H^{SW(n)}}{\Delta t} = Q_{SW}^{(n+1)} - \sum_i^N (H_i^{SW(n)} - H_i^{GW(n)}) L_i A_i \qquad (35)$$

Or

$$H^{SW(n+1)} = H^{SW(n)} + \frac{\Delta t}{S_{SW}}\left[Q_{SW}^{(n+1)} - \sum_i^N (H_i^{SW(n)} - H_i^{GW(n)}) L_i A_i \right] \qquad (36)$$

where n and n+1 denote previous and current time level respectively, Δt is the time step.

2) Semi-implicit Scheme:

Assuming that L_i is constant over the lake area, which is equal to L, from Eq(34), gives

$$S_{SW}\frac{\partial H^{SW}}{\partial t} = Q_{SW} - H^{SW} L \sum_i^N A_i + \sum_i^N H_i^{GW} A_i L_i \qquad (37)$$

Or

$$S_{SW}\frac{\partial H^{SW}}{\partial t} + H^{SW} L S_{SW} = Q_{SW} + \sum_i^N H_i^{GW} A_i L_i \qquad (38)$$

Hence,

$$\frac{\partial H^{SW}}{\partial t} + L H^{SW} = \frac{Q_{SW}}{S_{SW}} + \sum_i^N H_i^{GW} L_i \frac{A_i}{S_{SW}} \qquad (39)$$

Usually, H_i^{GW} is function of H^{SW}. Applying time level n+1 to H^{SW} (implicit) and time level n to H_i^{GW} (explicit), gives the semi-implicit scheme of Eq(39):

$$\frac{H^{SW(n+1)} - H^{SW(n)}}{\Delta t} + LH^{SW(n+1)} = \sum_i^N H_i^{GW(n)} L_i \frac{A_i}{S_{SW}} + \frac{Q_{SW}^{(n+1)}}{S_{SW}} \tag{40}$$

Or

$$H^{SW(n+1)} = \frac{1}{1+L\Delta t}\left(H^{SW(n)} + \Delta t \sum_i^N H_i^{GW(n)} L_i \frac{A_i}{S_{SW}} + \Delta t \frac{Q_{SW}^{(n+1)}}{S_{SW}} \right) \tag{41}$$

Combining both Eq(36) and Eq(41), gives

$$H^{SW(n+1)} = \frac{1}{1+\alpha L\Delta t}\left(H^{SW(n)} + \Delta t \sum_i^N H_i^{GW(n)} L_i \frac{A_i}{S_{SW}} + \Delta t \frac{Q_{SW}^{(n+1)}}{S_{SW}} - \Delta t(\alpha - 1) LH^{SW(n)} \right) \tag{42}$$

where if α=0 and L is constant over the surface body, Eq(42) will become Eq(36) and if α=1, then Eq(42) will become Eq(41).

Eq(32) or Eq(42) will be applied when Sw is constant. More general case, for explicit scheme, we have

$$\begin{aligned} &\frac{dV_{SW}}{dt} = Q_{SW} - Q_{GW} = Qs \\ &Q_{GW} = \sum_i^N (H_i^{SW} - H_i^{GW}) L_i A_i \\ &H_i^{GW} = \begin{cases} H_i^{GW} & \text{if } H_i^{GW} > Elev_{\mathrm{b}} \\ Elev_{\mathrm{b}} & \text{if } H_i^{GW} < Elev_{\mathrm{b}} \end{cases} \end{aligned} \tag{43}$$

And for semi-implicit scheme, we have

$$\frac{\partial V_{SW}}{\partial t} = Q_{SW} - (Q1_{GW} H^{SW} + Q2_{GW}) = Qs - Q1H^{SW}$$

Where

$$\begin{aligned} &Q1_{GW} = \sum_i^N L_i A_i \\ &Q2_{GW} = -\sum_i^N H_i^{GW} L_i A_i \\ &H_i^{GW} = \begin{cases} H_i^{GW} & \text{if } H_i^{GW} > Elev_{\mathrm{b}} \\ Elev_{\mathrm{b}} & \text{if } H_i^{GW} < Elev_{\mathrm{b}} \end{cases} \end{aligned} \tag{44}$$

Or

$$V_{SW} = V_0 + \Delta tQs - \Delta tQ1_{GW} H^{SW} \tag{45}$$

$$F(H^{SW}) = V_{SW} + \Delta tQ1_{GW} H^{SW} - (V_0 + \Delta tQs) = 0 \tag{46}$$

This gives

$$\frac{dF}{dH^{SW}} = \frac{dV_{SW}}{dH^{SW}} + \frac{\Delta tQ1_{GW} H^{SW}}{dH^{SW}} = A^{SW} + \Delta tQ1_{GW} \tag{47}$$

Therefore, the Newton-Raphson formula for solving H^{GW} can be written as

$$H_{new}^{SW} = H_{old}^{SW} - \frac{F(H_{old}^{SW})}{A^{SW} + \Delta tQ1_{GW}} = H_{old}^{SW} - \frac{V_{SW}(H_{old}^{SW}) + \Delta tQ1_{GW} H_{old}^{SW} - (V_0 + \Delta tQs)}{A^{SW} + \Delta tQ1_{GW}} \tag{48}$$

For explicit scheme $Q1_{GW}$=0.0 and Q_{GW} will be calculated by using water level at old time step; for semi-implicit scheme, Q_{GW} will be separated into two terms: $Q1_{GW}$ and $Q2_{GW}$ where $Q_{GW=}$ $H^{SW}Q1_{GW}$+ $Q2_{GW}$ and H^{SW} is water level at current time step.

3) Nonlinear Coupling Scheme

In this scheme, H^{GW} and H^{SW} will be solved separately and a differ correction (DC) technique will be applied to couple these two heads in such a way that the coupling terms in both surface water and groundwater equations tend to be identical. During each time step:

(a) Given $H^{SW(n)}$ and H_i^{GW} at time level n, solving Eq(48) yields $H^{SW(n+1)}$

(b) Once $H^{SW(n+1)}$ is known, solving Eq.(32) gives the new H_i^{GW}

(c) If $\left|newH_i^{GW} - oldH_i^{GW}\right| \le \varepsilon$ then go to next time step; or H_i^{GW} = New H_i^{GW}, repeats (a) through (c)

This iterative loop is shown in Fig.23.

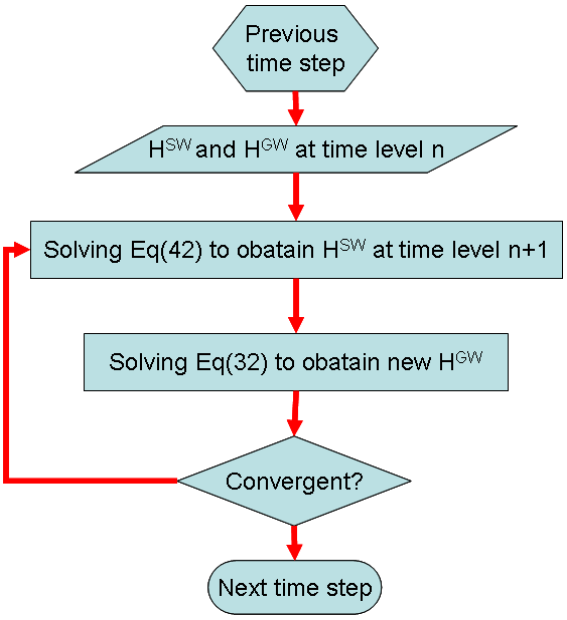


Figure 23 SW/GW coupling iterative loop

In order to incorporate the interaction of surface water and groundwater into hierarchical modeling system, the coupling iterative process will be added in the model solver, which will be illustrated in Fig.24.

From Fig.24, the coupling iterative loop will be embedded in each down and up scaling model solver and this will keep the whole hierarchical framework unchanged.

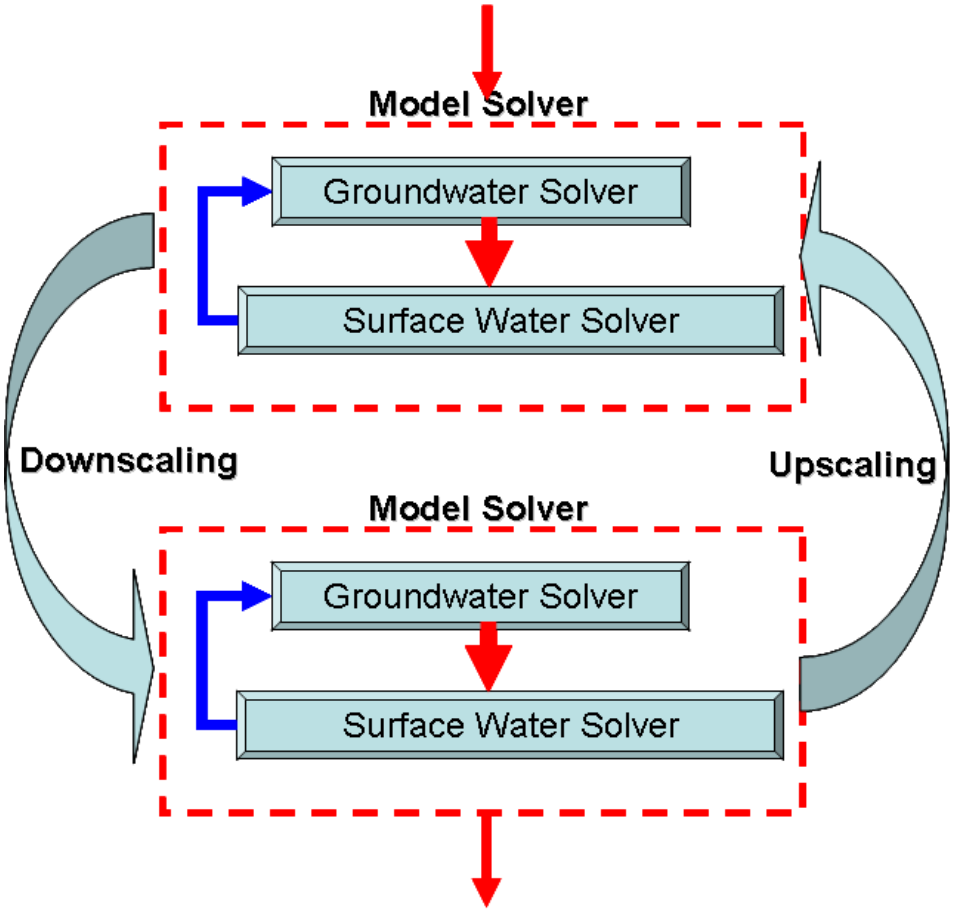


Figure 24 SW/GW coupling iterative loop embedded in model solver

(4) Scheme Comparison

The following simple test case illustrates the interaction of two lakes and one pumping well system (Fig.25). Q_{SW} in lake 1 and lake 2 are 500 m^3/day and 0 m^3/day respectively and the pumping rate at the well is -500 m^3/day. Coupling scheme and explicit scheme are used to do the comparison with different time steps.

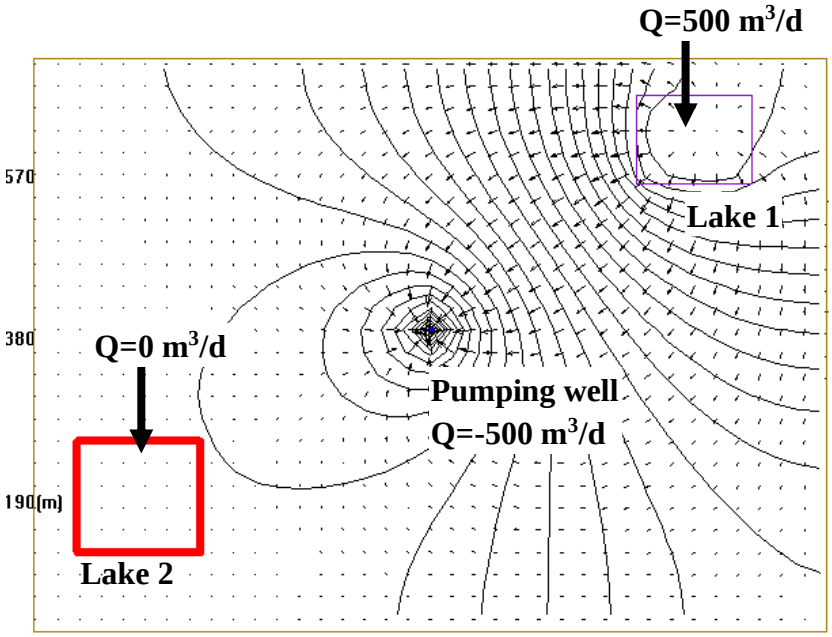


Figure 25 Interaction of two lakes and one pumping well system

Fig.26 to Fig.28 are showing the results of the time-varying lake water level obtained from these two scheme by using time step of 1 day, 3 days and 5 days. It is seen that coupling scheme predicts the same results as the explicit scheme does using small time step. Large time step will cause oscillation in explicit scheme while give the same results in coupling scheme.

Comparison of lake levels from explicit scheme using different time steps is shown in Fig.29. That from coupling scheme is shown in Fig.30. Again, it is clear that coupling scheme is more reliable than other schemes, especially large time step is involved.

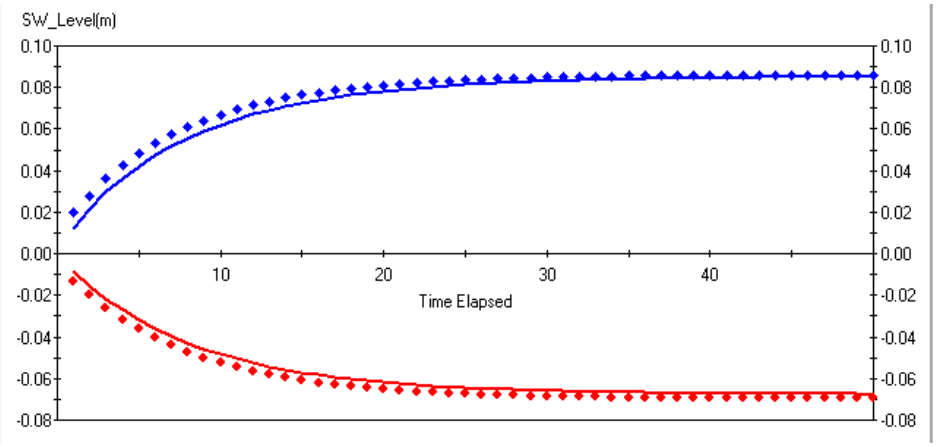


Figure 26 Coupling Scheme (solid lines) VS Explicit Scheme(circle) at Δt=1 days (Blue color represents water level in Lake 1, red color represents water level in Lake 2)

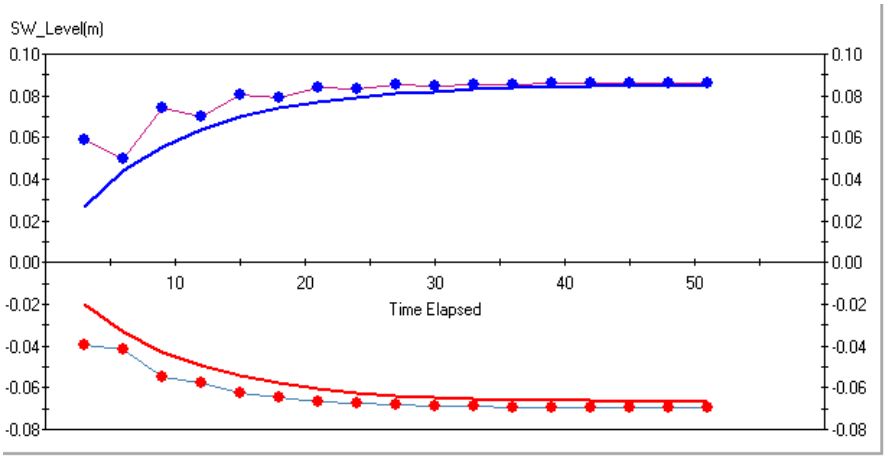


Figure 27 Coupling (solid lines) VS Explicit (solid line with circle) Scheme at Δt=3 days

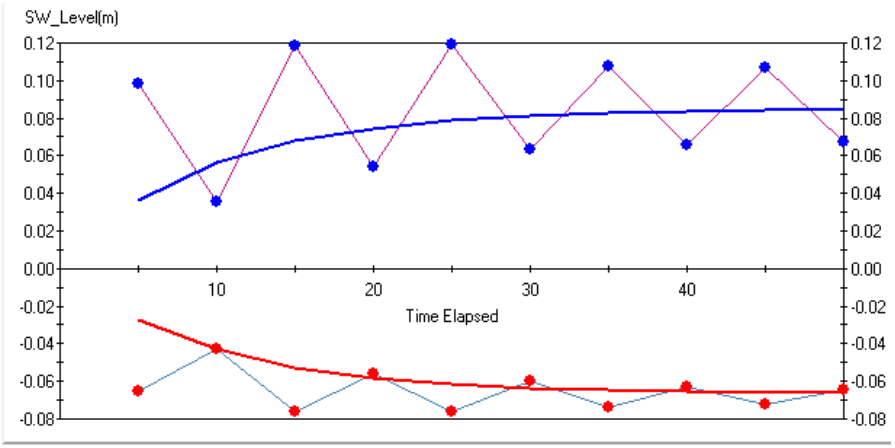


Figure 28 Coupling (solid lines) VS Explicit (solid line with circle) Scheme at Δt=5 days

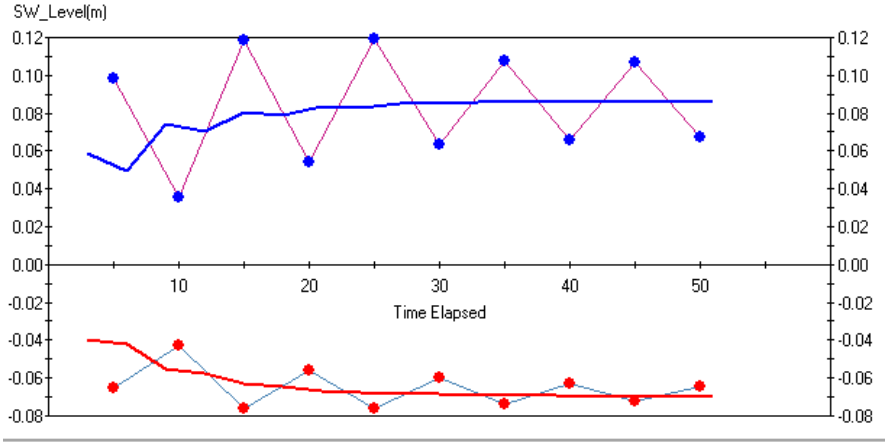


Figure 29 Explicit Scheme: Δt=3 days(solid lines) and Δt=5 days(solid line with circle)

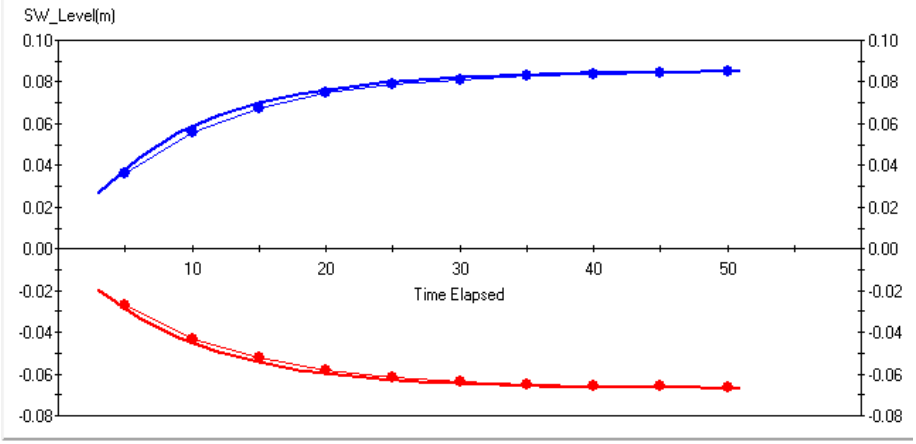


Figure 30 Coupling Scheme: Δt=3 days(solid lines) and Δt=5 days(solid line with circle)

4 Iteration Levels within the Algorithm

There are five nestled iterative procedures executed during each time step in a multiscale modeling system. Innermost among these is the matrix solver to solve the system of equations subject to the FVM scheme; the second nested one is for the head dependent source/sink terms; nonlinear unconfined head determination forms the intermediate iteration level; the coupling iteration of surface water and groundwater is embedded as the fourth iteration level and down and upscaling loop is the most outer iteration level of the system. The overall iteration loop structure in a multiscale modeling system is shown in Fig.21.

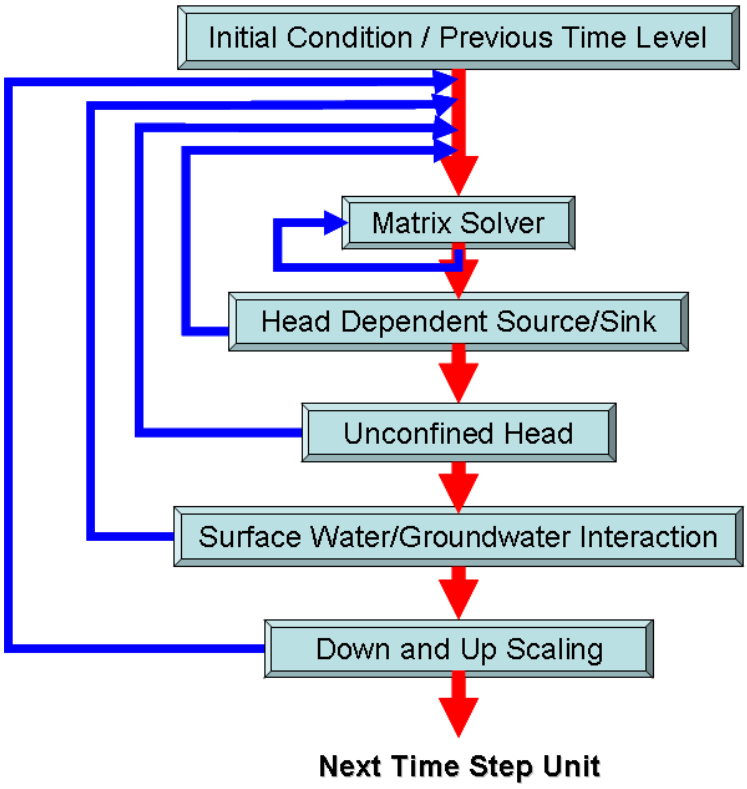
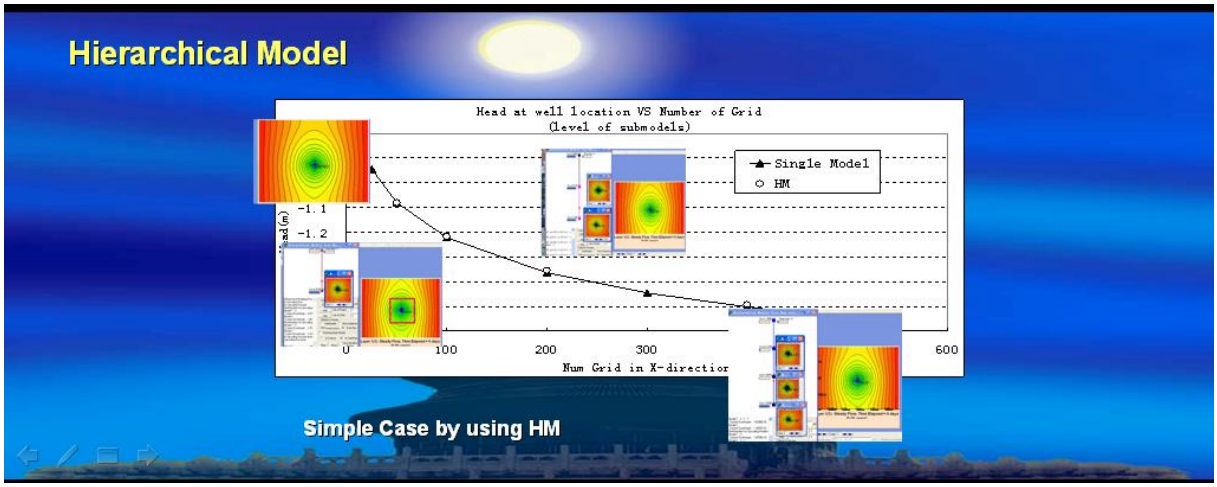


Figure 22 Overall Iterative Loop Structure in a Multiscale Modeling

5 Verification and Examples

Discription:
Parameter:
Results:
Remarks:



More readings please go to:

HydroProcess_fens_offprint.pdf
SisterLakesPaper_Vol20_Issue 11_2015.pdf
Zhephy.pdf