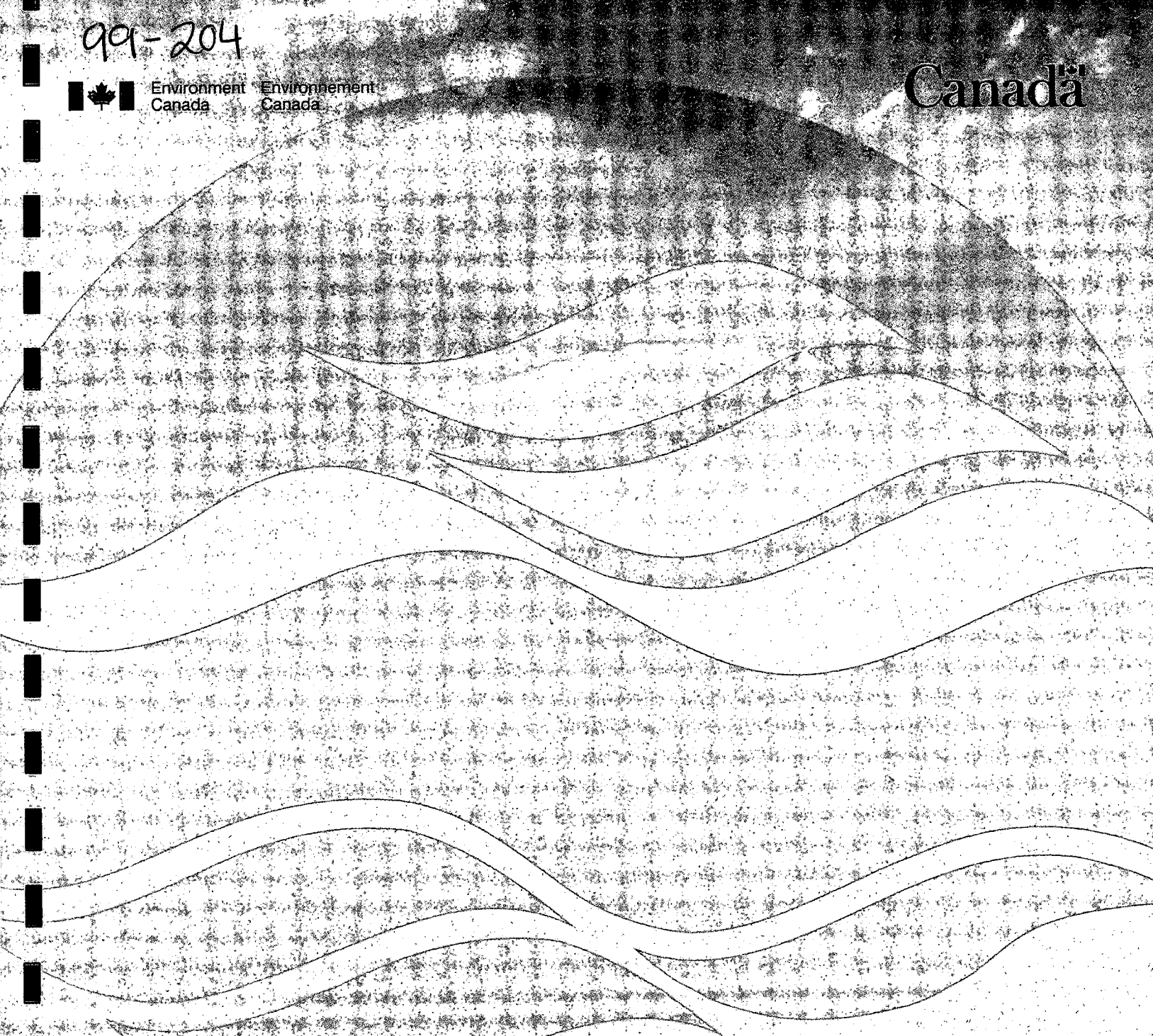
99-204

Environment Canada | Environnement Canada

Canada



User's guide for GW-Wetland: A Computer program to Simulate Groundwater Flow, Particle Tracking, and Solute Transport in a Two-dimensional Cross Section with Transient Boundary Conditions and a Fluctuating Water Table

BY

S. Shikaze, A.S. Crowe

NWRI Contribution No. 99-204

NATIONAL WATER RESEARCH INSTITUTE

INSTITUT NATIONAL DE RECHERCHE SUR LES EAUX

Management perspective

Title: User's Guide for GW-WETLAND: A Computer Program to Simulate Groundwater Flow, Particle Tracking, and Solute Transport in a Two-dimensional Cross Section with Transient Boundary Conditions and a Fluctuating Water Table

Authors and affiliations:
Steven G. Shikaze and Allan S. Crowe
National Water Research Institute, Canada Centre for Inland Waters, Burlington, Ontario L7R 4A6

NWRI Contribution Number: 99-204

Citation: NWRI Contribution

EC Priority/Issue:

The DOE Great Lakes 2000 policy needs reliable information on the potential environmental stress caused by human activities on the marsh ecosystem at Point Pelee National Park. It is suspected that nutrients from the Park's septic-system are entering the Point Pelee marsh via groundwater discharge and may be contributing to a deterioration in the health and natural biodiversity of the marsh. This model contributes to our basic understanding of the hydrogeological environment at Point Pelee and allows us to simulate the transport of nutrients to the marsh. It also has wider implications for the control of contaminant loadings to Point Pelee marsh, and its detrimental impact on this fragile aquatic ecosystem. This program fits under the COA activity1.6: Groundwater.

Current status:

Long-term monitoring of the groundwater flow regime at Point Pelee National Park has shown that hydrogeological environment that it is highly complex, it undergoes seasonal reversals in the direction of flow, and is affected by Lake Erie, the marsh, precipitation, evapotranspiration, and the width of the barrier bar. These factors have caused wide-spread contamination from the Park's septic systems and highly variable input of nutrients to the marsh. The model discussed here will enable Environment Canada to simulate the groundwater flow regime, and hence, understanding the processes and factors affecting the transport and fate of the contaminants to marsh at Point Pelee National Park.

Next steps:

The model is currently being expand and generalized to make it more applicable for other wetlands, not only within the Great Lakes basin, for wetland environments across Canada.

ABSTRACT

To adequately assess the impact of groundwater on a wetland, detailed and long-term field studies will be required. Due to the complexity of the groundwater flow regime and the hydrostratigraphy, the field studies to adequately assess groundwater conditions require considerable instrumentation and several years of monitoring. Thus, a more cost-effective approach and rapid means of obtaining considerable insight into the groundwater flow regime at wetlands and its interaction with the wetland is to use computer models. In addition, computer models can be used to (1) investigate processes occurring in this environment, (2) quantify fluxes across the groundwater-wetland interface, and (3) predict the impact of future hydrologic events on wetlands. This report describes a numerical model that can be used to simulate transient groundwater flow and contaminant transport in a 2-D cross section in a variety of groundwater-wetland scenarios, including conditions at the groundwater-wetland interface The model will account for a fluctuating water table, the formation/disappearance of seepage faces, infiltration and evapotranspiration fluxes along the top of the cross-section, the inclusion of pumping or injection wells, a heterogeneous sedimentary sequence, time-varying shorelines between the groundwater regime and the wetland and marsh. Fluctuations in the water table are considered by adding or deleting nodes and elements as the water table rises or falls, respectively. This method has the advantage of preserving the hydrostratigraphy as the water table rises or falls through different stratigraphic layers. Solute transport is simulated by solving the advection-dispersion equation with either a Galerkin finite element solution, or by particle tracking with or without dispersion.

USER'S GUIDE FOR GW-WETLAND

A computer program to simulate groundwater flow, particle tracking and solute transport in a two-dimensional cross section with transient boundary conditions and a fluctuating water table.

Version 1.0

Steven G. Shikaze
Allan S. Crowe

Groundwater Assessment and Restoration Project
National Water Research Institute
Environment Canada
Box 5050
867 Lakeshore Road
Burlington, ON
L7R 4A6

January 26, 1999

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

To adequately assess the impact of groundwater on a wetland, detailed and long-term field studies will be required. Due to the complexity of the groundwater flow regime and the hydrostratigraphy, the field studies to adequately assess groundwater conditions require considerable instrumentation and several years of monitoring. Thus, a more cost-effective approach and rapid means of obtaining considerable insight into the groundwater flow regime at wetlands and its interaction with the wetland is to use computer models. In addition, computer models can be used to (1) investigate processes occurring in this environment, (2) quantify fluxes across the groundwater-wetland interface, and (3) predict the impact of future hydrologic events on wetlands. This report describes a numerical model that can be used to simulate transient groundwater flow and contaminant transport in a 2-D cross section in a variety of groundwater-wetland scenarios, including conditions at the groundwater-wetland interface. The model accounts for a fluctuating water table, the formation/disappearance of seepage faces, infiltration and evapotranspiration fluxes along the top of the cross-section, the inclusion of pumping or injection wells, a heterogeneous sedimentary sequence, time-varying shorelines between the groundwater regime and the wetland and marsh. Fluctuations in the water table are considered by adding or deleting nodes and elements as the water table rises or falls, respectively. This method has the advantage of preserving the hydrostratigraphy as the water table rises or falls through different stratigraphic layers. Solute transport is simulated by solving the advection-dispersion equation with either a Galerkin finite element solution, or by particle tracking with or without dispersion.

1.0 INTRODUCTION

During the last several years, considerable research has been undertaken to understand the hydrology of wetlands. However, little research has focused on the role of groundwater in wetlands. Field studies investigating the groundwater flow regime associated with a wetland have shown that groundwater flow in the vicinity of wetlands is often complex and can, for example, exhibit reversals in the direction of groundwater flow. To adequately assess the impact of groundwater on a wetland, detailed and long-term field studies will be required. Due to the complexity of the groundwater flow regime and the hydrostratigraphy, the field studies to adequately assess groundwater conditions require considerable instrumentation and several years of monitoring. Thus such field programs may be quite costly. An alternative approach is to use computer models to simulate groundwater flow in the vicinity of the wetland and its interaction with the wetland. Numerical models offer a cost-effective and rapid means of obtaining considerable insight into the groundwater flow regime at wetlands. In addition, computer models can be used to (1) investigate processes occurring in this environment, (2) quantify fluxes across the groundwater-wetland interface, and (3) predict the impact of future hydrologic events on wetlands.

This report describes a numerical model that can be used to simulate groundwater flow and contaminant transport in a 2-D cross section in a variety of groundwater-wetland scenarios. It can be used to assess conditions at the groundwater-wetland interface (Figure 1.1a). It can also be used to study the groundwater-wetland environment of a wetland formed in a depression within the ground surface (Figure 1.1b), such as prairie potholes. It can assess groundwater flow between two-wetlands or a wetland-lake environment (Figure 1.1c) which may exist where a wetland is separated from a lake by a barrier bar

Although numerous groundwater flow models exist, these models can be limited with respect to numerical and conceptual accuracy in simulating groundwater-wetland interactions. The groundwater-wetlands model presented here overcomes limitations of existing models. The model can have wetlands and/or a lake located anywhere in the cross section. The model will account for a fluctuating water table, the formation/disappearance of seepage faces, a heterogeneous sedimentary sequence, time-varying shorelines between the groundwater regime and the wetland

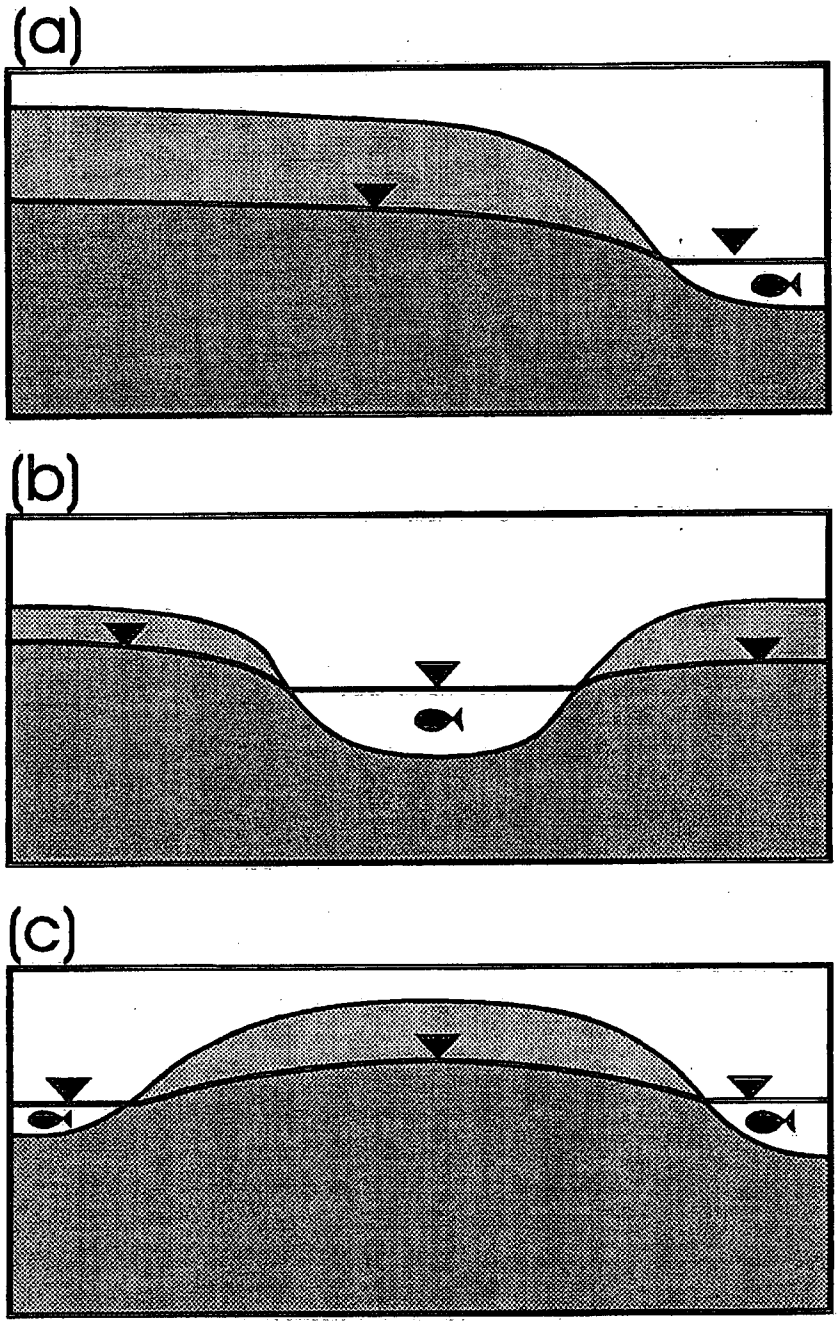


Figure 2.1 – Three scenarios that can be simulated with our numerical model. (a) A groundwater-wetland interface, (b) a groundwater-wetland environment in a wetland formed in a depression within the ground surface, and (c) groundwater flow between two-wetlands or a wetland-lake environment.

and marsh (fluctuate both vertically and laterally in response to changes in the size and shape of the wetland or lake level cycles).

In the types of systems outlined above, the position of the water table is a concern because the water table will fluctuate in response to changes in the lake/wetland levels as well as transient recharge and evapotranspiration. The numerical simulation of transient groundwater flow in an unconfined aquifer with a fluctuating water table can be problematic because as the water table rises or falls, the geometry of the flow regime changes. As a result, both the hydraulic head along the water table and the geometry of the flow system are unknown. The water table may rise or fall as a result of several phenomena. First, seasonal fluctuations in recharge across the water table may result in transient behaviour of the water table. Second, constant head boundaries (e.g., lakes, rivers, and marshes) may also behave in a transient or seasonal manner resulting a water table that changes temporally. As the water table rises, the size of the saturated zone increases and conversely, if the water table drops, the size of the saturated zone decreases. Numerical solutions that do not account for these changes may not accurately simulate the system in some instances.

There are several numerical techniques available to solve this type of problem (for a review of some of the techniques available, see Crowe et al., 1999). The first method is to include the flow of water in the unsaturated zone through the use of Richard's equation. The water table is then defined as the surface along which pressure head is equal to zero. This method, however, requires a more detailed field study to obtain parameters required to simulate flow in the both unsaturated and the saturated zones. This can result in an expensive and time-consuming field program to collect the necessary data. Moreover, the numerical solution of coupled saturated-unsaturated flow can be more time consuming and computationally expensive. A second method is to simulate only the saturated zone by assigning a grid or mesh to the saturated zone and apply a flux at the water table to represent flow through the unsaturated zone to the water table. The two most common approaches are to either maintain a fixed grid throughout the simulation even when the shape of the saturated zone changes, or to deform the mesh as the flow regime expands or contracts. Because numerical inaccuracies can arise in the calculation of the position of the water table when the grid is fixed, the method that deforms the grid is preferred. With this method, the top boundary of the computational domain, which represents the water table, can

freely move up or down in response to boundary conditions. In this method, any number of rows of elements can deform - ranging from only the top row, to all of the rows in the domain. However, if the water table rises or falls through a unit of contrasting permeability, the stratigraphic units may be distorted by the deformation of the elements. In this instance, the finite element mesh may no longer be representative of the stratigraphy (Figures 1.2a,b). Our method makes use of a third technique that overcomes the disadvantages listed above.

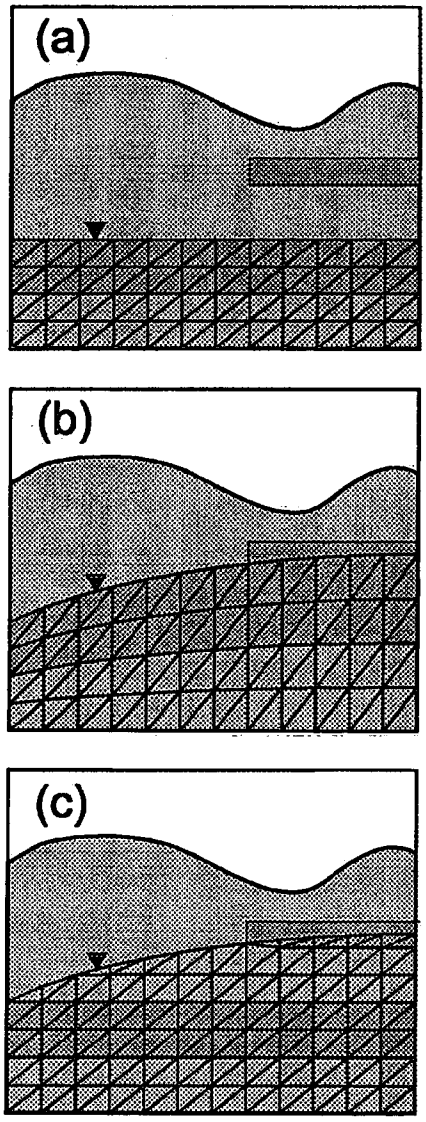


Figure 1.2 – Illustration of the geometry of the hydrostratigraphic units modelled by the regeneration of a finite element mesh as the water table rises. (a) Initial grid and hydrostratigraphy, (b) grid at some later time generated with a method that stretches all rows of elements, and (c) grid at some later time generated by our algorithm where nodes and elements are added.

The model presented here solves the transient groundwater flow equation in a two-dimensional cross section. Fluctuations in the water table are considered by adding or deleting nodes and elements as the water table rises or falls, respectively. This method has the advantage of preserving the hydrostratigraphy as the water table rises or falls through different stratigraphic layers (Figures 1.2a,c). Moreover, explicit definition of parameters used to define the unsaturated zone is not required.

Also included in this code are the following features:

- The ability to simulate scenarios where the value of hydraulic head at constant head nodes can exceed the elevation of the ground surface. This can be useful when simulating lakes, rivers or wetlands where the hydraulic head is known a priori.
- The values that are specified at constant-head nodes are allowed to change in time. In addition, constant head nodes can be turned 'off' and 'on'. This includes nodes that represent surface-water bodies described above.
- Infiltration and evapotranspiration fluxes along the top of the cross-section can also change with time and position.
- Solute transport can be simulated by solving the advection-dispersion equation.
- Particle tracking can be simulated with or without dispersion. Particles can be added anywhere in the domain, and at any time during the solution. Because boundary conditions can change in time, the particle tracking routine is based on a transient velocity field.
- Seepage faces can form and disappear along the water table-ground surface interface.
- Fluid fluxes and a fluid mass balance are calculated.
- Pumping or injection wells can be included and the pumping/injection rates can change in time, or be turned off and on.
- The horizontal size (Δx) of the elements can be specified to be either constant or variable.

This user's guide is set up to provide some relevant background material as well as some example test cases. Chapter 2 presents the background theory including governing equations, boundary conditions for transient groundwater flow, particle tracking and solute transport and a description

of the grid-modification procedure. Chapter 3 provides two example problems that illustrate transient groundwater flow through barrier bars of different widths. An input guide is provided in Appendix A, and Appendices B and C provide samples of a data file and an output file, respectively.

2.0 BACKGROUND THEORY

The computer program presented here is a two-dimensional, cross-sectional numerical model that can be used to solve the transient groundwater flow equation with options to include transient particle tracking (with or without dispersion) and/or solution of the advection/dispersion equation. Section 2.1 presents the theory for the groundwater flow problem. This type of problem is typically referred to as a 'free-surface problem', in which the distribution of hydraulic head and the elevation of the water table are unknown. Included in this section is a discussion of our methodology, which allows the mesh to be modified in response to changes in the water table. In Section 2.2, the particle tracking routine is presented, and in Section 2.3, the theory of solute transport via the advection/dispersion equation is discussed.

2.1 GROUNDWATER FLOW

2.1.1 Governing Equations and Boundary Conditions

As mentioned previously, the approach presented here considers only the saturated zone (i.e., the water table is the top boundary of the computational domain). This top boundary is allowed to rise or fall in response to boundary and initial conditions. The model solves the groundwater flow equation in a two-dimensional cross section. The governing equation for transient groundwater flow in the saturated zone (S in Figure 2.1) is:

$$\frac{\partial}{\partial x_i}\left[K_{ij}\frac{\partial h}{\partial x_j}\right] = S_s\frac{\partial h}{\partial t} \tag{1}$$

where K_{ij} is the hydraulic conductivity tensor [L/T], h is hydraulic head [L], S_s is the specific storage coefficient [L^{-1}], t is time [T], x_i is the coordinate vector and $i, j = x,z$ [L].

Because equation (1) is defined on a transient basis, initial hydraulic heads must be defined for the entire domain, and the initial elevation of the water table must also be specified. The initial conditions are:

$$h(x,z,t=0) = h_o(x,z) \tag{2}$$

$$\delta(x,z,0) = \delta_o(x,z) \tag{3}$$

where δ is the elevation of the free surface (F in Figure 2.1) above a datum [L], δ_o is the initial elevation of the free surface [L], h_o is the initial hydraulic head throughout the domain [L].

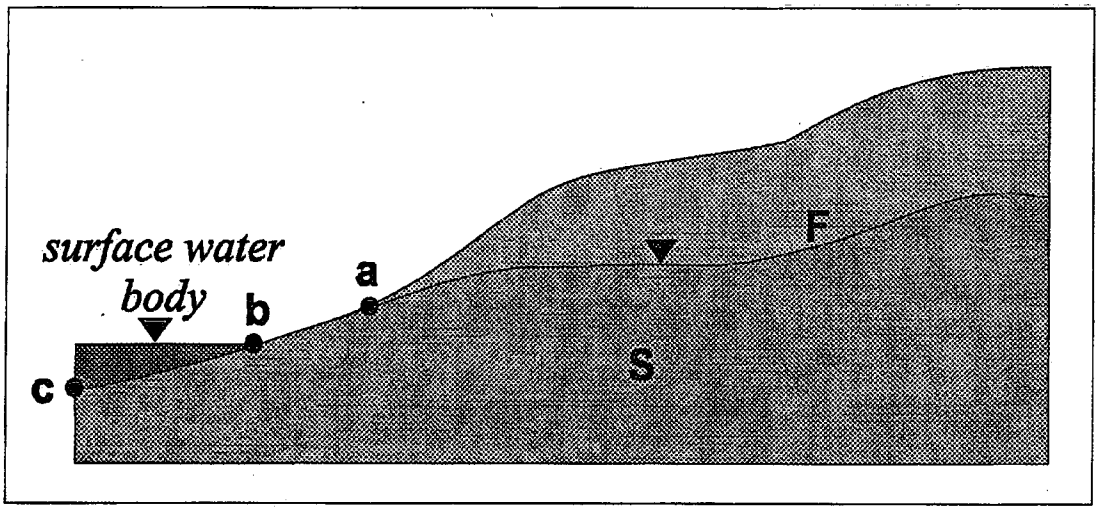


Figure 2.1 – Schematic cross section of a typical computational domain.

Boundary conditions must also be specified. Options include constant or specified hydraulic head (first-type or Dirichlet) boundaries, and specified fluid flux (second-type or Neumann) boundaries. The boundary conditions for equation (1) are:

$$h(x,z,t) = H(x,z,t) \qquad \text{on b-c (Figure 2.1)} \tag{4}$$

$$K_{ij}\frac{\partial h}{\partial x_i}n_i = -Q(x,z,t) \tag{5}$$

$$\delta(x,z,t) = h(x,z,\delta,t) \qquad \text{on F (Figure 2.1)} \tag{6}$$

$$K_{ij}\frac{\partial h}{\partial x_i}n_i = \left[R - S_y\frac{\partial \delta}{\partial t}\right]n_3 \qquad \text{on F (Figure 2.1)} \tag{7}$$

$$h(x,z,t) = z \qquad \text{on a-b (Figure 2.1)} \tag{8}$$

where H is the hydraulic head on a constant head boundary [L], R is the rate of vertical recharge along the free surface [L/T], n_i is the unit outward normal vector, S_y is the specific yield and Q is the flux along a specified-flux boundary [L/T].

Equation (4) represents a first-type or Dirichlet boundary condition where specified values of hydraulic head are assigned along a boundary. The value of the specified head at these boundaries can change in time, and constant head nodes can be turned off and/or on at any time during a simulation. Equation (5) represents a second-type, or a Neumann boundary condition, where a specified flux across a boundary is assigned. The value of flux at these second-type nodes can also change in time or be turned off/on at any time during a simulation (e.g. recharge or pumping rates that change in time). Equations (6) and (7) represent the boundary conditions along the free surface, depending on whether or not recharge fluxes are present. Equation (8) represents a seepage-face boundary where the hydraulic head is equal to elevation of the ground surface. Neuman and Witherspoon (1971) describe the boundary value problem defined by equations (1) through (8) in further detail.

Once the hydraulic heads have been solved, groundwater velocities can be determined at each time step based on:

$$v_i = \frac{K_i}{\theta}\left[\frac{\partial h}{\partial x_i}\right] \tag{9}$$

where v_i is the average linear groundwater velocity in direction i [L/T] and θ is the porosity.

The solution presented here is non-linear because both the hydraulic heads at the water table and the geometry of the domain are unknown. Thus, an iterative solution is required at each time step. At the end of each iteration, numerical convergence is tested by calculating a residual that is defined as the sum of the absolute difference between the head at the nodes along the water table ($h(x,t)$) and the elevation ($\delta(x,t)$) of this node:

$$residual = \sum\left|\delta(x,t) - h(x,t)\right| \tag{10}$$

The solution has converged when the residual is less than a user-defined tolerance or closure criterion.

2.1.2 Other Features

The program allows pumping/injection wells to be added by specifying a node or group of nodes as second-type nodes with the pumping/injection rate specified by the user. These nodes are chosen by specifying the x- and z-coordinates of a rectangle inside which all nodes will be assigned the specified flux. These pumping/injection nodes can be turned off and/or on at any time during the simulation.

Specified-head nodes where the hydraulic head is greater than the elevation of the ground surface can also be specified. These are referred to as 'surface water bodies' and include nodes that represent lakes, rivers, marshes or other surface water bodies. The values of head assigned to these nodes can be changed in time. Moreover, these constant head nodes can be switched to free nodes and vice-versa at any point during the simulation. This feature allows for inclusion of surface water bodies that have water levels that change in time.

A fluid mass balance can be calculated at each time step. Fluid fluxes can be calculated at all first-type (i.e., constant-head) nodes. The sum of these fluxes in addition to the sum of the fluxes at second-type nodes (i.e., recharge and pumping wells) is then compared to the change in fluid mass storage for the time step in question. For fluid mass to be conserved, the change in fluid mass storage for a given time step should equal the sum of the fluid fluxes into and out of the computational domain.

Seepage faces can also form if the water table reaches the ground surface. If a seepage node forms along the water table, the node becomes a constant head node, with the value of head set as equal to the elevation of the ground surface. If boundary conditions are such that the water table is falling in time, the seepage face may disappear. The program checks for this by examining the fluid flux at the seepage node. If the flux is negative (i.e., water is exiting the domain at this point), the node remains a constant-head node. If, however, the flux is positive, the seepage node is changed from a constant-head node back to a free node.

2.1.3 Mesh Generation/Modification

The two-dimensional form of equation (1), subject to initial and boundary conditions in equations (2) – (8), is solved in a vertical, two-dimensional cross section using a standard finite-element technique. The finite element equations are formulated using the standard Galerkin method (see for e.g., Pinder and Gray, 1977). The algorithm for generating the finite element grid includes the following:

- The position of the water table can rise or fall over time as a result of boundary conditions that can change in time.
- All nodes along the water table are located at $\delta(x,t) = h(x,t)$.
- The interfaces between hydrostratigraphic units within the saturated zone are always located at nodes.

To account for changes in the shape of the flow domain, the method presented here involves a combination of a limited stretching of elements and/or the addition or removal of nodes and elements along the water table. If the change in the position of the water table is small with respect to the vertical grid spacing, the elements along the water table are stretched or compressed. If the change in position is large with respect to the vertical grid spacing, new elements and nodes are added or removed. The procedure for adding or removing nodes or elements during a transient simulation is summarized below. For a more detailed explanation of the procedure, the user is referred to Crowe et al. (1999).

The first step is to discretize the computational domain into triangular finite elements. This is done internally and the initial mesh is dependent on the geometry of the domain, the boundary conditions and the initial elevation of the water table. The grid generator assigns an elevation to each node along the uppermost row of the mesh that is equal to the initial value of hydraulic head of the water table at that node. Below the top row of elements, the mesh is comprised of uniform, triangular finite elements with constant Δz values as specified by the user (Figure 1.2c).

Because both the elevation of the water table and hydraulic heads are unknown, an iterative solution is required to locate the position of the water table. At the beginning of each time step, the elevation of the nodes along the water table is fixed and the hydraulic heads within the flow domain are calculated (Figures 2.2a, 2.2g). The difference between the elevation and the calculated head for each node along the water table is then examined. If the sum of the

differences at all nodes is greater than a user-specified convergence tolerance, the nodes along the water table are repositioned vertically to a location corresponding to the calculated value of hydraulic head (the x-position remains constant). Because only the nodes along the water table are allowed to move, only the top row of elements is stretched or compressed. Changing the vertical dimension of an element produces a new vertical spacing of $\Delta\zeta$ for the top layer of elements (Figures 2.2b, 2.2h). All remaining elements below the uppermost row of elements remain at a constant vertical spacing of Δz. At the end of each iteration, numerical convergence is tested by calculating a residual based on the difference between the head and the elevation of the nodes along the water table (see equation (10)). Convergence is attained when the residual is less than a user-defined tolerance. Moving to the next time step, the process is repeated and the mesh is allowed to change again (Figures 2.2c, 2.2e, 2.2i, and 2.2k).

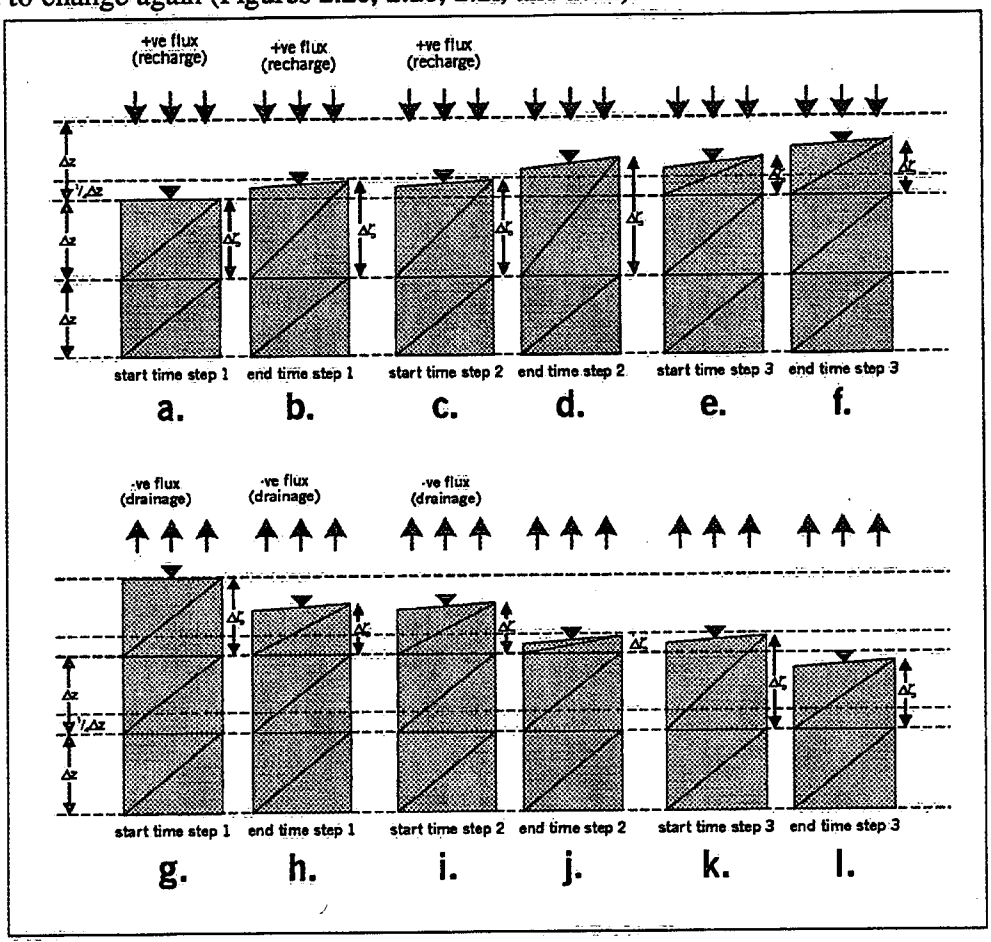


Figure 2.2 – Schematic diagrams of the development of a new mesh. a-f show the addition of new nodes and elements, g-l show the removal of nodes and elements.

The procedure outlined above is used with many finite-element codes that allow the grid to deform as the shape of the flow domain changes. However, in our method, at the beginning of each new time step, if an element has been stretched more than $^1/_4\Delta z$ beyond the regular vertical grid spacing ($\Delta\zeta > {}^5/_4\,\Delta z$), we form a new node and a new element. The new node is inserted at the regular Δz spacing, and the new element is inserted along the water table with a vertical element spacing of $\Delta\zeta_{new} = \Delta\zeta_{old} - \Delta z$ (Figure 2.2e). If an element stretches less than $^1/_4\Delta z$ beyond the regular grid spacing ($\Delta\zeta < {}^5/_4\Delta z$), only the top two elements are stretched, and a new node is not inserted (Figure 2.2b). Similarly, if a node at the water table declines by more than $^3/_4\Delta z$ of the regular grid spacing ($\Delta\zeta < {}^1/_4\Delta z$), the node immediately below this water table node is removed (Figure 2.2k). If the decline of a water table node is less than $^3/_4\Delta z$, the z-position of this node is simply lowered to the current value, thereby compressing the finite element, with no removal of nodes or elements (Figure 2.2h).

Because all elements, except those at the water table, are maintained at the original vertical grid spacing of Δz, unit boundaries remain unchanged (see Figure 1.2c). The only instance where the mesh may not coincide with the unit boundaries occurs when the water table passes into a new geologic unit. Initially, the changes in the water table elevation may result in water-table elements stretching less than $^1/_4\Delta z$. If the stretched node exists at the interface between two units, the stratigraphy will not be preserved because new elements are not formed and the new stratigraphic unit will not exist in the model. With time, the water table may continue to rise to a point where a new element will form, at which time the stratigraphy will be once again be accurately represented. The error resulting from this slight misrepresentation is small. Once the new elements form, element boundaries will be placed at the proper Δz spacing. Hence, the error will likely be insignificant.

2.1.4 Limitations

Convergence problems can develop when the water table moves through a boundary between geologic units that have a large contrast in hydraulic conductivity. For example, if the water table rises from a unit of low to one of high hydraulic conductivity, the elements in the lower-K unit will stretch until the hydraulic heads increase beyond $^1/_4\Delta z$. After this, new water-table elements

will form, but these new elements will have the higher K value assigned to them, and the low K elements will shrink to the regular element size. Because these new elements have a higher K, the hydraulic head along the water table can decrease resulting in a drop in the water table. If these high-K elements shrink to below $^3/_4\Delta z$, they will be removed and the low-K elements will be stretched upwards. This can result in an increase in the hydraulic head and consequently, the formation of a new high-K element again. This entire sequence may repeat itself and convergence may not be achieved. As a result, the water table will oscillate at the node(s) in question.

We provide two solutions to rectify this problem. First, an algorithm is included within the code that identifies this type of oscillation. With the first solution, the criterion for forming a new element is decreased from $^1/_4\Delta z$ to $^1/_{10}\Delta z$, and the criterion for removing an element is increased from $^3/_4\Delta z$ to $^9/_{10}\Delta z$ for the current time step. After the water table rises across the boundary between low-K elements and high-K elements, a new element will form. However, when the water table subsequently falls due to the new high-K element, the element will not be removed. This solution results in convergence in most cases.

In some cases where the contrast in hydraulic conductivity of the two units is very large, the above solution may still not result in convergence. In this case, we recommend that the user change the hydraulic conductivity of the cell containing the element where oscillation is occurying. We recommend that the user start with a value of hydraulic conductivity that is halfway between those of the two adjacent units. This second solution will always result in convergence.

2.2 CONTAMINANT TRANSPORT

The program solves for solute transport with two different methods. The first method is based on a deterministic-probabilistic solute transport model (Ahlstrom et al., 1977; Schwartz, 1978; Schwartz and Crowe, 1980). In this method, particles, which are assigned a contaminant mass, are added to the system. Advection is simulated by tracking the particles through time based on the transient groundwater flow field. Dispersion is added to the particle motion by adding a random component to the deterministic particle track.

In the second method, a standard Galerkin finite element scheme (see for e.g., Pinder and Gray, 1977) is used to solve the advection-dispersion equation in the two-dimensional cross section.

Two methods are provided here because each method has its own advantages. For cases where dispersion is very low (i.e., systems that are dominated by advection), the deterministic-probabilistic routine (Section 2.2.1 below) will behave better. This routine can also be used in cases where dispersion is completely neglected (i.e., particle tracking). Moreover, this method is not subject to constraints that restrict the second method, such as the Peclet and Courant stability criteria. The Galerkin method for solution of the advection-dispersion equation (Section 2.2.2) allows a more direct method for solving solute transport, whereby advection and dispersion are solved simultaneously. Dispersion is accounted for directly, rather than as a 'random' component, as is the case with the method outlined in Section 2.2.1. Also, in our code, the second method can include simple retardation and degradation processes.

2.2.1 Deterministic-Probabilistic Mass Transport

The two-dimensional advection-dispersion equation is given as:

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x_i}(v_i c) - \frac{\partial}{\partial x_i}\left(D_{ij} \frac{\partial c}{\partial x_j} \right) = \Sigma k_n c \qquad (11)$$

where c = the concentration of the contaminant [M/L^3], v_i is the average linear groundwater velocity [L/T], D_{ij} is the hydrodynamic dispersion tensor [L^2/T] and k_n is the chemical rate constant for reaction n.

The deterministic-probabilistic method presented here uses particles as the basic units. Each particle represents a given mass of contaminant and is allowed to enter and be transported through the flow system. Particles can be added to a source location within the saturated flow domain. Because the x- and z-position of each individual particle is known within the domain, the distribution of contaminant mass can also be determined.

The first step involved in this method is to solve for advective transport. This solution requires both horizontal and vertical velocities from the groundwater flow solution (equation (9)). Each particle moves along its respective vectors a distance that is determined by the size of the time step, the direction of the hydraulic gradient and the magnitude of the velocities. This advective displacement represents the deterministic component of contaminant transport and is calculated by:

$$x_{i,t} = x_{i,t-1} + v_i \Delta t \tag{12}$$

where $x_{i,t}$ is the new x_i position of the particle, $x_{i,t-1}$ is the x_i position of the particle during the previous time step and Δt is the size of the time step.

Hydrodynamic dispersion is incorporated into the particle motion by adding a random component to the advection particle motion. This random motion is related to the dispersive characteristics of the porous medium. The new particle locations arising from dispersion are calculated by (Ahlstrom et al., 1977):

$$\begin{aligned} x_{1,t} &= x_{1,t-1} + x_1' \frac{v_1}{u} x_2' \frac{v_2}{u} \\ x_{2,t} &= x_{2,t-1} + x_1' \frac{v_2}{u} x_2' \frac{v_1}{u} \end{aligned} \tag{13}$$

where $x_{i,t}$ is the displacement of a particle in the x_i direction due to dispersion (defined below), and u is the groundwater velocity in the direction of flow, defined as:

$$u = \sqrt{v_1^2 + v_2^2} \tag{14}$$

The distance that a particle moves due to dispersive motion is related to the dispersivity characteristics of the porous medium. A random number is included to account for the tortuosity of the porous medium:

$$x_1' = \sqrt{24 D_l \Delta t} * (0.5 - RND) \tag{15}$$

$$x_2' = \sqrt{24 D_t \Delta t} * (0.5 - RND) \tag{16}$$

where D_l and D_t are the longitudinal and transverse dispersion coefficients [L^2/T], and *RND* is a random number between 0 and 1. The longitudinal and transverse dispersion coefficients are defined as:

$$D_l = \alpha_l |u| \tag{17}$$

$$D_t = \alpha_t |u| \tag{18}$$

where α_l and α_t are the longitudinal and transverse dispersivities, respectively [L].

Calculation of actual contaminant concentrations requires that the flow domain be divided into 'cells'. Each cell is rectangular in shape and is composed of two adjacent triangular finite elements. During each time step, the number of particles in a given cell is changed by having particles either enter or leave the cell. By summing the mass carried by all particles in a given cell and by determining the volume of water in the cell, the contaminant concentration in the cell can be calculated by:

$$C_c = \left(\sum p_{i,0} m_{i,0} + \sum p_{j,in} m_{j,in} - \sum p_{k,out} m_{k,out} \right) / V\theta \tag{19}$$

where C_c is the concentration of a contaminant in the cell, $p_{i,0}$ is the number of particles in a cell at the start of the time step, $p_{i,in}$ is the number of particles entering the cell during the time step, $p_{k,out}$ is the number of particles leaving the cell during thes time step, m is the mass of a particle, V is the volume of the cell, and θ is the porosity.

2.2.2 Galerkin Method for Solution to the Advection-Dispersion Equation

Contaminant transport can be simulated by use of the standard Galerkin finite element solution (see for e.g., Pinder and Gray, 1977). In this case, at each time step, the same finite element mesh that was used for the groundwater flow solution is used for the advection-dispersion equation. In this method, both the advective and the dispersive components are solved simultaneously, rather than separately, as they are in the deterministic-probabilistic method presented in the previous section. The method presented here also includes simple, first-order degradation, as well as retardation.

The advection-dispersion equation used for this solution is given as:

$$\frac{\partial c}{\partial t} = -\frac{\partial}{\partial x_i}\left(\frac{v_i}{R_i} c\right) + \frac{\partial}{\partial x_i}\left(\frac{D_{ij}}{R}\frac{\partial c}{\partial x_j}\right) - \lambda c \tag{20}$$

where λ is the degradation constant [T^{-1}]. In this solution, the hydrodynamic dispersion tensor is given by:

$$\begin{aligned} D_{xx} &= D^*\tau + \frac{\alpha_l v_x^2}{|u|} + \frac{\alpha_t v_z^2}{|u|} \\ D_{xx} &= D^*\tau + \frac{\alpha_t v_x^2}{|u|} + \frac{\alpha_l v_z^2}{|u|} \\ D_{xz} &= D_{zx} = (\alpha_l - \alpha_t)\frac{v_x v_z}{|u|} \end{aligned} \tag{21}$$

where D^* is the free-solution diffusion coefficient [L^2/T] and τ is the tortuosity of the porous medium. Other parameters are defined in the previous section.

Currently, only first-type, or Dirichlet boundary conditions can be specified in the program. These allow the user to specify a fixed-source concentration at any location within the computational domain. In the current version of the code, the solute source must be present throughout the duration of the simulation and the location and rate of application of contaminant of the source must remain constant.

2.2.2.1 Stability

Numerical stability of the solution of the advection-dispersion equation can be controlled by examination of the well-known Courant and Peclet criteria. Although these criteria were originally developed for the simpler one-dimensional solution, they may be used as a guideline for the stability of the two-dimensional solution provided here. These criteria do not control the outcome of the solution; however, the maximum values for each criterion as well as the element number where the maximum is occurring are printed to the output file to aid the user in determining why a particular solution has failed or resulted in a questionable solution.

The Peclet Number (P) can be calculated for each element and for each of the horizontal and vertical directions as follows:

$$P_x = \frac{v_x \Delta x}{D_{xx}}$$
$$P_z = \frac{v_z \Delta z}{D_{zz}} \qquad (22)$$

As a guideline, these values should be less than about 2. The Courant number (C) can also be determined for each element and for each direction:

$$C_x = \frac{v_x \Delta t}{\Delta x}$$
$$C_z = \frac{v_z \Delta t}{\Delta z} \qquad (23)$$

These parameters ensure that the distance the solute moves within a given time step does not exceed the size of the element itself. As a guideline, these two parameters should not exceed 1. It should be noted that minor violation of the Peclet and Courant criteria may not result in numerical problems in some instances.

A final criterion used in the program is the aspect ratio (A), which is defined as:

$$A = \frac{D_{xx}\Delta z^2}{D_{zz}\Delta x^2} \tag{24}$$

Although no strict guidelines for this parameter exist, experience has shown that the aspect ratio should fall within the range of 0.01 to 100.

3.0 OPERATION OF THE MODEL

This chapter describes the requirements as well as the procedure for running the program.

3.1 COMPUTING REQUIREMENTS

The program has been programmed in FORTRAN, and compiled with Microsoft PowerStation, version 4.0. Some of the utilities in the code use functions that are specific to Microsoft PowerStation, so if the user wants to compile the code with another FORTRAN compiler, they should contact the authors of the code before doing so. The minimum system requirements are:

- Pentium processor (166 MHz is recommended as the minimum)
- 32 MB RAM (actual memory requirements will depend on the size of the problem)
- A hard disk with at least 40 MB free.

The executable program runs in a DOS environment, and it is recommended that the user open a DOS-window from Windows95 (or NT) to run the program. Output files are in ASCII format, and specific output has been designed to be used in GridBuilder (McLaren, 1998) for visualization.

3.2 DIMENSIONING OF THE ARRAYS

Dimensioning arrays is done through an INCLUDE file, listed in APPENDIX A.4. Thus, if the user requires larger dimensions for a specific problem, changes only need to be made to this INCLUDE file.

3.3 INPUT OPERATIONS

This section describes the data requirements for input to the model. An input guide containing a list of input parameters and variables, units for all appropriate input parameters, and the input file setup, is presented in Appendix A. User is referred to Appendix A for more information on the input parameters given below in bold letters. Appendix B contains a sample input file.

The user has the option of undertaking the simulations using either English or metric units for length. However units of time must be in days and units of mass must be input in milligrams. Appendix A.5 summarizes input parameters that need units and acceptable combinations of units.

The user can insert comments within the input data set as either a line of comments, when the column one of the line contains a "!"; or after the parameter values on a line by first entering a ";". It highly recommended that the user take advantage of this option to place comments within the data file to aid in finding and changing variables.

The first information that the use must enter is a one-line title or problem identifier (**title**). This provides a description of both the input data set and the output. Twenty-four program control parameters (**option(n)**) are used to control the nature of the simulation and the type of output generated. If the user wishes to choose one of the particular functions, **option** should be set to the logical value "**.TRUE.**" (a value of "**.FALSE.**" indicates that the option or output is not required). The user has the option of running various routines, such as:

option(5) : calculate heads

option(12) : generate hydraulic gradients and their direction for each cell

option(15) : calculate the velocity in each element

option(17) : run mass transport simulation using particle tracking

option(24) : run mass transport simulation using Galerkin method

The finite element grid and head distribution are automatically calculated at the beginning of the simulation, based on initial information supplied by the user, and at each subsequent time step. Information about the system may be output to a printer at each time step:

option(1) : node coordinates

option(2) : finite element incidences

option(3) : hydraulic conductivity and storativity of the element

option(6) : hydraulic head values

option(13) : hydraulic gradients and direction of gradient

option(18) : particle and concentration distributions

option(23) : coordinates of particles and associated mass

Also, the following information may be output to files that could be used for plotting, for example with GridBuilder:

option(4) : node coordinates and element incidences

option(7) : head distribution at each time step

option(8) : head distribution at the final time step only

option(14) : hydraulic gradients and direction of gradient

option(16) : velocity field

option(19) : concentrations and distribution of reference particles

option(20) : distribution of reference particles

option(14) : particle concentration distribution

Addition options allow the user to control various aspects of the simulation, and check its convergence:

option(9) : print water table convergence information

option(10) : input final heads from a previous run to continue a simulation

option(11) : output final heads to continue the simulation

option(21) : output the final distribution of particles to continue a simulation

option(22) : input the final distribution of particles to continue a simulation

The grid for the model is constructed of an array of nodes and elements. This grid is constructed within the program according to the user's input parameters. The user must specify the number of rows (**nrow**) and columns (**ncol**) of nodes and the vertical (**delz**) and horizontal (**xe**) spacing between nodes. The vertical node spacing must be the same for each row of cells, but the horizontal node spacing may be constant or variable. If the horizontal grid spacing is constant, set **var_x** to **.TRUE.** and enter one value for **xe**, else set **var_x** to **.FALSE.** and enter the different values of **xe**.

The cross section may contain up to 9 different hydrostratigraphic units (**ngeol**), with different values of horizontal and vertical hydraulic conductivity (**khorz**, **kvert**), storativity (**stor**), porosity (**por**), longitudinal dispersivity (**disperl**), and transverse dispersivity (**dispert**). These units represent the subsurface from the base of the cross section to the ground surface, whether or

not the entire cross section will be saturated. The units are defined as an array (**mapgeo**) by assigning a numbered code, from 1 to 9, to represent the presence of a specific hydrostratigraphic unit at each cell within the domain. The codes are entered one row at a time starting with the uppermost row of cells, and having one code value assigned to each cell on that row, and each subsequent lower row in the cross section starts a new line. Thus, once the values for **mapgeo** are entered, it should appear exactly the same as illustrated in the drawn cross section. The values of the parameter assigned to a hydrostratigraphic unit (**khorz, kvert, stor, por, disperl, dispert**) are input with the hydrostratigraphic unit identifier (**I**) corresponding to the unit in the cross section (**mapgeo**). One line is entered for each hydrostratigraphic unit. Also, entered on this line are the codes to indicate whether or not a particle is allowed to move (**code**) within the specified unit. As illustrated in the example in Appendix B, a simple graphical representation of the cross section for the particular variable is formed. This style of data input facilitates convenient entry of data for complex settings, rapid alterations, and easy checking of input data. Only one value of specific yield (**specyd**) for the unsaturated zone is entered.

Additional parameters and values are used to define the shape of the cross section and its initial hydrogeological conditions. The elevation of the ground surface (**elvgrd**) and elevation of the base of the cross section (**elvbas**) are input for each column of nodes. Because these values need not be constant, the cross section may have an irregular shape. A single initial value of hydraulic head assigned to the saturated domain is required (**headi**); to help the solution converge, this initial value should be set equal to the average value of the head along the water table. The initial values of head along the water table, or elevation of the water table (**headwt**) is also required as one value for each column of nodes. The user must also indicate which heads along the water table may fluctuate vertically in response to changes to the flow domain (set **wtmove** = **.TRUE.**) and which may not move (set **wtmove** = **.FALSE.**). If surface water nodes or drains are specified, **wtmove** should be specified as **.FALSE.** for these columns of nodes.

The user can specify various boundary conditions for the flow domain, including infiltration/evapotranspiration values (type two boundary) and constant head values (type one boundary). Infiltration and evapotranspiration are input separately to accurately simulate field conditions. Recharge and evapotranspiration can vary both over time (including periods of recharge/evapotranspiration and no recharge/evapotranspiration) and spatially across the cross

section (including areas of recharge/evapotranspiration and no recharge/evapotranspiration). Recharge is entered by first indicating the number of recharge periods (**nrestp**) that will occur during a simulation, and the time at which the recharge period starts (**t_rech**) (e.g., if recharge changes twice, **nrestp** = 3). If recharge occurs from the beginning of the simulation, then the first value of **t_rech** = 0.0, else enter the starting time. For each recharge period, the spatial distribution along the cross section is indicated by indicating the number of zone of constant recharge across the cross section (**nrzones**), the starting and ending column number (**nxfrom, nxto**) bounding a zone, and the value of recharge within each zone (**rval**). The first value of **nxfrom** must always be 1, the last value of **nxto** must always be equal to **ncol**. Values of **rval** must be positive, including a values of 0.0 for **rval** to indicate that no recharge occurs in a given zone and at a specific time. If no recharge occurs throughout the simulation, then the user simply enters a value of 0 for **nrestp**, and values for **t_rech**, **nrzones**, **nxfrom**, **nxto** and **rval** are not entered. Evapotranspiration is handled in the same way. The user enters the number of periods of time in which different evapotranspiration occurs (**netper**), the times at which these evapotranspiration periods occur (**t_et**), the number of spatial zones along the cross section (**netzone**), and starting and ending columns for these zones (**nxfrom_et, nxto_et**), and the values of evapotranspiration within each zone (**etval**). If evapotranspiration is not to be simulation, then the user simply enters a value of **netper** = 0, and does not enter values for **t_et, netzone, nxfrom_et, nxto_et,** and **etval.**

Constant head nodes are typically used to represent surface water bodies (lakes, rivers, wetlands), drains, or anywhere where the head is known. Surface water bodies can be represented as constant head nodes by two different methods within the model. However, it is recommended that, if the water level in the surface water body changes frequently over time, the user select and input information using the 'Surface Water Body Data' means of entering data (see Appendix A.2.8). If the user selects this method, the user must create a separate file containing the water-level history of the surface water body. This file should be set up in ASCII format, and the first line in this file should contain number of entries in the file, and each successive line should contain a value of time, followed by a new head value. Within the input data set, the user first specifies the number of distinct surface water bodies (**nwb**) within the cross section. For each surface water body, the user must specify the number of columns of nodes associated with a surface water body

(**nchnodes_wb**), and the corresponding columns of nodes at which represent the location of the surface water bodies (**icolumns**), as well as the file name that contains the water-level history (**file_wb**). The columns chosen should include every nodal column where the maximum head in the water-level history of the surface water body exceeds the elevation of the ground surface, even if during parts or most of the simulation, the elevation of the surface water body is above ground surface at much few column of nodes. For example, if the surface of the water body specified in **file_wb** range between 100.0 an 101.0 m during the simulation period and has an initial elevation of 100.2 m, then all columns at the location of the surface water body where the ground surface is less than or equal to 101.0 m should be listed in **icolumns**. If the user does not want to enter surface water bodies or constant heads in this manner, then set **nwb** = 0 and do not enter values for **nchnodes_wb**, **icolumns**, and **file_wb**.

Regular constant head nodes (non-surface water bodies) can also be specified (e.g., drains where water levels do not change with time or changes considerably less often than above). With this method, the user specifies the number of constant head nodes (**nchead**). If a node is to be designated as a constant head node during a later time in the simulation (e.g., it is originally not a constant head node), then it should still be included in **nchead**. For each constant head node, the following must be specified. The location of each of these constant head nodes (or future constant head nodes) is specified by its the nodal column (**ich**) and row (**jch**). The original head value (**chval**) assigned to the node must also be entered, as well as the number of times the head value for the current constant head node will change during the simulation (**nchg_head**). If the node is originally not a constant head node, then assign a value of -1.0×10^{35} to **chval** to designate that it will become a constant head node late during the simulation. The user may read the values of the changing values for the constant head node read from separate file (set **chead_file** = **.TRUE.**), or from the current input data file (set **chead_file** = **.FALSE.**). If **chead_file** is **.TRUE.**, then the name of the file that contains the changing head data (**chead_file_name**) must be specified. If this is the case, a list containing the time at which the value of head at the specific node changes (**t_new**) and the new value of head (**chg_h_val**) should be entered in **chead_file_name** and no additional information is required in the input data set. If **chead_file** is **.FALSE.**, a list of **t_new** and **chg_h_val** should be specified within the input data set and an additional file is not required. If the constant head node changes from a constant head node to a

regular node during the simulation, then assign a value of -1.0×10^{35} to **chg_h_val** to designate that it will become a regular node at time **t_new**. In either case, this list should contain **nchg_head** values. If the value of the constant head does not change during the simulation, then set **nchg_head** = 0, and do not enter values for **chead_file, chead_file_name, t_new,** and **chg_h_val**. If the user does not wish to enter constant heads this way, then simply set **nchead** = 0 and do not enter any values for **ich, jch, chval, nchg_head, chead_file, chead_file_name, t_new,** and **chg_h_val**.

The model also as the flexibility to include pumping wells within the cross section. The location within the cross section from which the pumping actually occurs (i.e., screened interval of the well, is defined by identifying the nodes from which the pumping occurs. The user first enters the number of pumping wells (**nwells**). For each well, the "well screen" is located by entering the left and right x-coordinates (**x1, x2**) and the upper and lower z-coordinates (**z1, z2**) of a box representing the location of the well screen. All nodes within this box are identified as the well screen, and a pumping rate is applied to each of these nodes. The user can change the pumping rate during the simulation, by entering the number of pumping periods (**n_pump_per**), the time at which the change or change occurs (**ton_pw**), and the new pumping rates (**prate**). The pumping rate specified is the pumping rate assigned to each node within the well screen box. If only 1 period of pumping occurs, enter **n_pump_per** = 1. If there are no pumping wells, the user simply enters a value of **nwells** = 0, and does not enter values for **x1, x2, z1, z2, n_pump_per, ton_pw,** and **prate**.

The length of the simulation, size of the time steps, and times at which information is output are controlled by the user. Rather than running the model to a specific number of time steps, the user specifies the actual time in which the simulation should stop (**ftime**). This is done because the size of the time step is allowed to vary and thus, desired times can be attained regardless of the size or number of time steps. The user selects an initial time step (**delt**), and the rate at which the time step increases in size (**deltin**). If a constant time step is desired, the user enters **deltin** = 1.0, else the time step will increase in size until the specific maximum time step size (**dtmax**) is reached, at which point the time step (**delt**) will remain constant for the rest of the simulation. Because of the nonlinear nature of the solution, the user must also enter the convergence tolerance (**tolrnc**) for convergence within an iteration. A value of 0.001 is recommended. This

value represents 1 mm of head, if the units of length are metres. The user must also specify the maximum number of iterations for the nonlinear portion of the code (**nitmax**). Typically, the code will converge in less than iterations, but in some cases, more iterations may be required.

Output is generated at times specified by the user in two ways. First, a variety of information about the grid, hydrogeology, and contaminant distribution of the cross section (see **options** above) is output as either printed output or as data files which can be used as input to other software (see **options** above and Appendix A.3). The number of times at which information should be printed (**n_out_time**) and the desired times (**t_out**) is input by the user. At one time should be specified by the user. Because the simulation may proceed with a varying time step, it may be difficult to know the exact simulate times throughout the simulation. However, the times specified by the user do not have to correspond exactly with the simulation time. The model will output the information at the simulation time closest to the user-specified time. Secondly, the elevation of the water table at any location along the cross section can be printed, producing a time series record of water table elevations for all time steps. This could be used to compare the model's prediction of water table elevations to a water table well located along the cross section. Any number of observation points can be selected by entering the number of observation points along the water table (**n_obs**) and the column correspond to the desired location of output of water table elevations (**nobs_col**).

The user can also simulate solute transport and basic reactions using one of two methods. When the particle tracking using the deterministic-probabilistic mass transport method is selected (**option(17)** = **.TRUE.**), only mass transport by advection and dispersion occurs. The user defines the initial mass of each particle (**xmassi**). Next the user enters the number of source zones within the cross section (**npart_zones**). Then for each source zone, the user defines the location and size of the source zone by entering the left and right columns (**plcol, prcol**), and bottom and top rows (**pbrow, ptrow**), bounding the source zone. The user specifies the number of solute particles are added to each source zone (**npper**), the time at which particles are introduced into the source zone (**t_start_part**), and the time at which particles stop entering the source zone (**t_stop_part**). If the user wishes to enter particles for only one time step, they should enter the same time for both **t_start_part** and **t_stop_part**. Each source zone can have a different size, number of

particles entered, and different times at which particles are placed in the source zone. If mass transport by this method is not selected (**option(17)** = **.FALSE.**), none of these data are entered.

When solute transport using the Galerkin method for the solution of the advection-dispersion equation is selected (**option(24)** = **.TRUE.**), the user can simulation both advective and dispersive transport as well as limited reactions that retard (**retfact**) or decay (**decay**) the contaminant. **retfact** represents the dimensionless retardation factor, and **decay** represents the decay coefficient or degradation rate constant. Also required are the free-solution diffusion coefficient (**dstar**) and the tortuosity of the medium (**tort**). There are also options for the time-weighting formulation for the Galerkin solution for the advection-dispersion equation. **epsi** can be set to 0.0 (explicit formulation), 0.5 (Crank-Nicolson weighting) or 1.0 (implicit). It is recommended that 0.5 be used because this value results in the most stable solution. If **consist** = **.TRUE.**, consistent formulation is used, whereas if **consist** = **.FALSE.**, lumped formulation is used. This refers to the way in which concentrations are obtained from the Galerkin method. It has been shown that consistent formulation results in less error in the solution, and is thus recommended. Once this data is entered, the user must enter the number of source zones inside which the concentration must be specified (**n_conc_zones**). For each concentration source zone, the user must specify the x- (**x1conc, x2conc**) and z-coordinates (**z1conc, z2conc**) that define a box inside which all nodes will be assigned a fixed concentration value. **x1conc** and **x2conc** represent the minimum and maximum x-values of the 'box', respectively, while **z1conc** and **z2conc** represent the minimum and maximum z-values of the box. The final variable for this section is **cinit**, which represents the concentration value assigned to the current zone. The units are in relative concentrations, so all concentrations calculated in the domain will be relative to **cinit**.

Other programming hints are shown below:

- The values specified at constant head nodes are allowed to change in time. In addition, constant head nodes can be turned 'off' by specifying the value to be '$-1.0e^{+35}$'.
- When making use of the restart option (see **option(10), option(11), option(22)**), there are a few things to keep in mind.

- If you wish the CREATE a restart file, **option (10)** should be **.TRUE.**. This will create a file called **`prefix.hou`**. Output will be written to this file at the most recent output time that is specified in the data file (**`prefix.dat`**). To input from a restart file, it is recommended that you rename the **`prefix`** to, for example, **`prefix2`**. Also, the **`prefix.hou`** file will have to be renamed **`prefix2.hin`**. The new data file (**`prefix2.dat`**) should be modified as discussed below. To restart **`prefix2.dat`**, **option(11)** should be set to **.TRUE.**.
- If you restart at time $t = t_{rs}$ you must change to following such that they do not include any data changes at $t < t_{rs}$.
    - constant head nodes
    - infiltration/evapotranspiration
    - pumping rate
    - output times.
- Make sure that at the first solution time, there are no changes in (1) the values of heads at constant head nodes, (2) infiltration and evapotranspiration rates (3) pumping rates.
- As of December 1998, the restart option only works for the flow solution, and particle tracking; it does not currently work for concentrations generated by the solution of the advection-dispersion equation.
- If particle tracking is simulated, the final locations of the particles are written to the **`prefix.hou`** file. So, if the **`prefix2.dat`** file is restarted as described above, you should remove the particle data, or you'll be starting particles from their original locations in addition to the restarted particles.

4.0 ILLUSTRATIVE EXAMPLES

The objectives of this chapter are to demonstrate how the model can be used to simulate typical scenarios. Also, this chapter aids in understanding and interpreting results from the model. In this chapter, two illustrative examples are presented.

4.1 NARROW BARRIER BAR – 50 M

The first example is designed to show the transient behaviour of groundwater flow in barrier bars that lie between a marsh and a lake (e.g., Point Pelee National Park, Ontario, Crowe and Ptacek, 1999). The water levels in the lake and the marsh vary in time. These variations result in a complex groundwater flow field where the direction of flow can change, depending on the boundary conditions. The width of the barrier bar is 50 m. Figures 4.1 shows a schematic of the cross sections for the 50-m barrier bars, respectively. Also, in this figure are the initial locations of the water table. The lower boundary is seen to slope upwards to the right because at Point Pelee, the aquifer is underlain by a low-permeability clay unit that slopes in this manner.

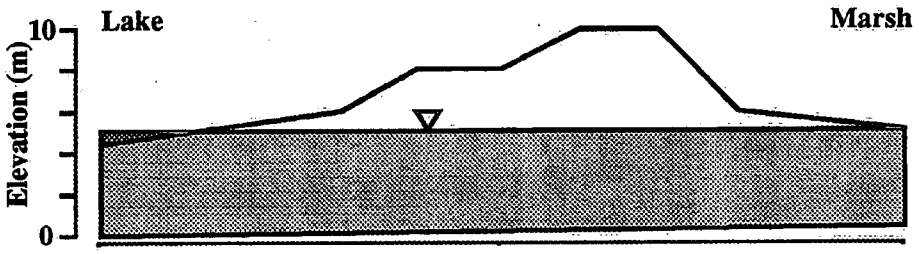


Figure 4.1 – Schematic diagrams of the cross sections for the first illustrative example.

The simulation was run to 730 days (2 years), and the water-level history in both the lake and the marsh is repeated for each of the 2 years. In general, water levels are higher in the spring and summer months, and lower in the winter months. Moreover, boundary conditions have been set up such that the water level in the lake varies more significantly than that in the marsh (Figure 4.2). Because of this, during part of the year, the water level in the marsh is higher than that in the lake, and the rest of the year, the reverse is true. Thus, the flow of groundwater can undergo a reversal in direction - with flow from the marsh to the lake during part of the year, and from the

lake to the marsh during the remainder of the year. The extent of this reversal, however, will depend on the width of the barrier bar as well as the boundary conditions.

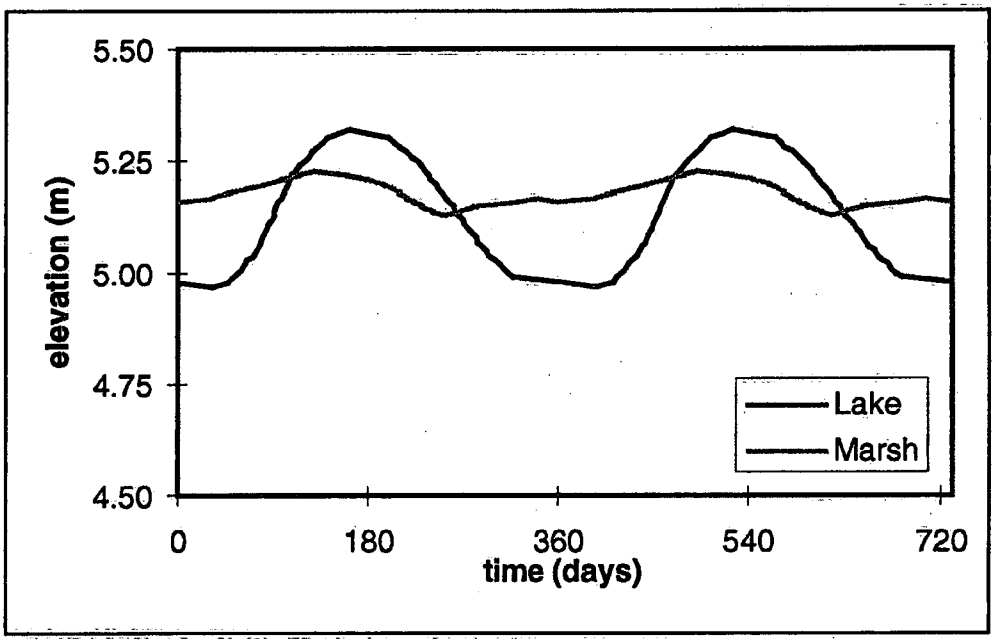


Figure 4.2 – Water levels in the lake and marsh used in the first illustrative example.

No recharge has been added to the water table, and this will allow us to isolate the effects of the changing lake and marsh water levels. Also, particles are added at the beginning of each simulation. These particles represent particles or slugs of water, and examination of the transient behaviour of these particles will aid in the understanding of the flow systems.

The first illustrative example involves transient groundwater flow in a barrier bar that is 50 metres wide. The water-level histories for both the lake and marsh are shown in Figure 4.2. The aquifer is assumed to be homogeneous, and the hydraulic conductivity and porosity are 8.64 m/day and 0.30, respectively. The specific storage coefficient is 0.0005 m^{-1}. As mentioned previously, recharge is not added to this system, and three particles are added at the beginning of the simulation in order to obtain a better understanding of the flow system.

The complete data file for this example is listed in Appendix B1. The section has been set up with 51 columns of nodes (**nrow**) and 101 rows of nodes (**ncol**). This results in 50 columns of cells and 100 rows of cells with a grid spacing of 0.1 metres in the vertical (z) direction (**delz**) and 1.0 metres in the horizontal (x) direction (**delx**). There is one single hydrostratigraphic unit

(parameters listed above), and the initial water table is at an elevation of 5.0 metres. The ground surface varies across the section as show in Figure 4.1. The water table nodes across the top boundary of the domain are allowed to move (**wtmove=T**) except for the nodes that are specified as surface-water bodies (**wtmove=F**).

In this example, the lake and marsh are represented as separate surface-water bodies, each with its own water-level history. The lake boundary (left-hand side) consists of nine nodes along the water table, and the water-level history for these nodes (Figure 4.2) is shown in Appendix B2 (file: lake.hed). The marsh boundary (right-hand side) consists of three nodes along the water table. The water-level history for the marsh (Figure 4.2) is shown in Appendix B3 (file: marsh.hed).

The particles have been added using the deterministic-probabilistic method described in section 2.2.1. For this example, dispersion has been neglected and thus the three particles migrate by advection only and thus represent water particles. All three particles have been added at the beginning of the simulation and their migration is tracked throughout the duration of the two-year simulation. The left particle (near the lake) has been placed in the cell that lies between nodal columns 8 and 9, and nodal rows 26 and 27. The centre particle has been placed in the cell between nodal columns 26 and 27 and nodal rows 26 and 27, while the right particle (near the marsh) has been placed in the cell between nodal columns 46 and 47, and nodal rows 26 and 27. Figure 4.3a shows the hydraulic head distribution and the velocity vectors at a time of 515 days. At this time, the water level in the lake is greater than that in the marsh (Figure 4.2). As a result, the direction of groundwater flow is from left to right (i.e., from the lake to the marsh). Figure 4.3b shows the heads and velocities at a time of 730 days. At this time, the water level in the marsh exceeds that in the lake, and consequently, the direction of groundwater flow is right to left (i.e., from the marsh towards the lake). Thus, a complete reversal of the flow direction can be seen within the 50-metre barrier bar.

Figure 4.4 shows the migration of three water particles over the duration of the simulation. The three particles were placed near the lake, near the marsh, and in the centre of the domain, initially, and their migration was tracked at each time step. Because the initial flow direction is towards the lake, the particle that is closest to the lake exits the groundwater flow system and enters the lake at a time of approximately 50 days. The remaining two particles exhibit an

oscillatory flow pattern, with a net movement from the marsh towards the lake. The net marsh-to-lake flow is due to the fact that the water level in the marsh is greater than that in the lake for a longer period of time each year. If the simulation were to be run for a longer period of time, these two remaining particles would eventually exit the groundwater flow system and enter the lake.

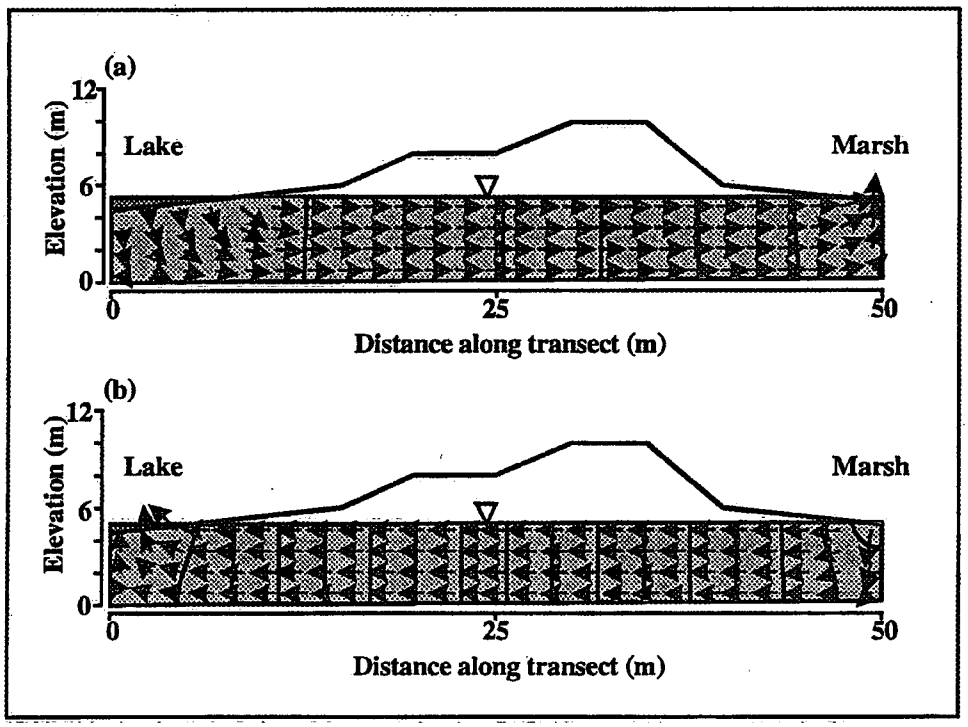


Figure 4.3 – Hydraulic heads and velocity vectors for the example with the 50-metre barrier bar, at (a) 515 days and (b) 730 days.

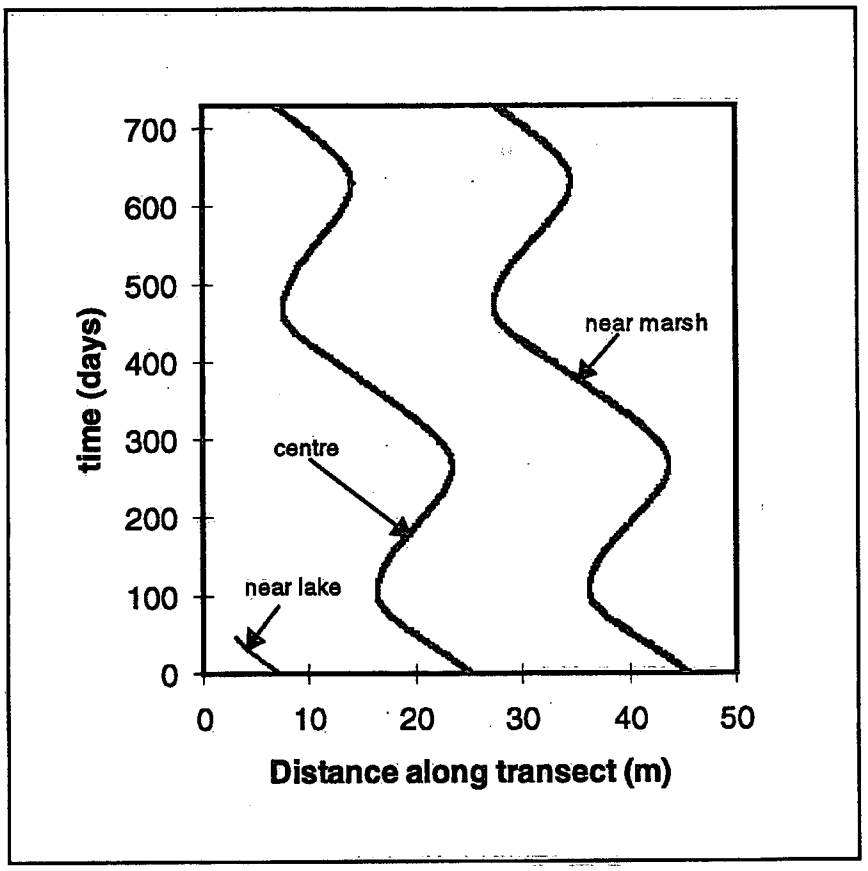


Figure 4.4 – Horizontal particle movement for three particles within the 50-metre barrier bar.

4.2 PRAIRIE SLOUGH

The second example involves a central topographic depression such as might be representative of a prairie slough. The purpose of this simulation is to show a scenario where the time-dependent surface-water boundary condition is located in the centre of the computational domain. When simulating this type of domain, the user may typically take advantage of symmetry if the properties on both the right and left sides of the surface-water body are identical. In the simulation presented here, we have added a unit of contrasting hydraulic conductivity and thus symmetry cannot be used. Figure 4.5 shows a schematic of this simulation.

For this simulation, the hydraulic conductivity of the main aquifer (K_1 in Figure 4.5) is $1x10^{-5}$ m/s. A layer of lower hydraulic conductivity (K_2 in Figure 4.5) has been added along the upper right-hand side of the domain ($K_2 = 1x10^{-7}$ m/s). The porosity and storage coefficient for each zone have been set to 0.30 and 0.0005 m^{-1}, respectively.

PRINTED IN CANADA
IMPRIMÉ AU CANADA
ON RECYCLED PAPER
SUR DU PAPIER RECYCLÉ

National Water Research Institute
Environment Canada
Canada Centre for Inland Waters
P.O. Box 5050
867 Lakeshore Road
Burlington, Ontario
L7R 4A6 Canada

National Hydrology Research Centre
11 Innovation Boulevard
Saskatoon, Saskatchewan
S7N 3H5 Canada

NATIONAL WATER RESEARCH INSTITUTE
INSTITUT NATIONAL DE RECHERCHE SUR LES EAUX

Institut national de recherche sur les eaux
Environnement Canada
Centre canadien des eaux intérieures
Case postale 5050
867, chemin Lakeshore
Burlington, Ontario
L7R 4A6 Canada

Centre national de recherche en hydrologie
11, boul. Innovation
Saskatoon, Saskatchewan
S7N 3H5 Canada

Environment Canada Environnement Canada

Canada