

05-192



Environment Canada  
Environnement Canada

Canada



NATIONAL WATER  
RESEARCH INSTITUTE  
INSTITUT NATIONAL DE  
RECHERCHE SUR LES EAUX

TD  
226  
N87  
No.  
05-192

Input Guide for Revisions  
for GW-WETLAND (version 2.0):  
A computer program to simulate  
Groundwater-Wetland Interactions

BY:

S. Shikaze, A. Crowe

NWRI Contribution No: 05-192

05-192

**Input Guide for Revisions for  
GW-WETLAND (Version 2.0):**

**A Computer Program to Simulate  
Groundwater-Wetland Interactions**

S.G. Shikaze and A.S. Crowe

**ABSTRACT**

Computer simulation models offer a cost-effective approach and rapid means of obtaining considerable insight into the groundwater - wetlands environment. Computer models can be used to (1) investigate processes occurring in this environment, (2) quantify fluxes across the groundwater-wetland interface, (3) predict the impact of future hydrologic events on wetlands, and (4) investigate the impacts of various remediation plans, and (5) investigate contaminant transport via groundwater into a wetland. This report describes a numerical GW-WETLAND 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, spatially and temporally varying infiltration and evapotranspiration fluxes along the top of the cross-section, the presence of pumping wells, installation of drainage ditches, a heterogeneous sedimentary sequence, and wetland shorelines that vary with time and location. The model is capable of either calculating the water levels in surface-water bodies in response to fluxes to/from the groundwater system as well as precipitation, or if the history of these water levels is known a priori, the user can set the surface-water-level history as a specified boundary condition that will influence the groundwater regime. The model also has the capability to include contaminant source zones that are associated with surface water boundaries to examine the potential for groundwater contamination as a result of the movement of water from the surface water body into the adjacent aquifer. Although the model was originally written for groundwater-wetland interactions, it is general enough for a variety of groundwater-surface water problems relating to rivers, lakes, and wetlands. Moreover, the model is constantly being updated in order to allow it to be used for new scenarios.

**NWRI RESEARCH SUMMARY**

**Plain language title**

Computer simulation models provide valuable insight into groundwater-wetland interaction.

**What is the problem and what do scientists already know about it?**

During the past two hundred years, most of the coastal wetlands of the Great Lakes have been lost due to human activities. Environment Canada is committed to

protecting existing wetlands and restoring degraded wetlands. Very little is known about the linkages between groundwater and these coastal wetlands.

#### **Why did NWRI do this study?**

This work is being undertaken as part of an Environment Canada program to protect and restore coastal wetlands of the Great Lakes. Field and modelling studies to evaluate the role of groundwater in these coastal environments is undertaken by NWRI scientists who have considerable expertise in hydrogeology and groundwater-surface water interaction. For example, computer modelling studies were undertaken at Point Pelee National Park to assess the potential transport of septic system contaminants via groundwater into the marsh.

#### **What were the results?**

Computer simulation models offer a cost-effective approach and rapid means of obtaining considerable insight into the groundwater - wetlands environment. Computer models can be used to (1) investigate processes occurring in this environment, (2) quantify fluxes across the groundwater-wetland interface, (3) predict the impact of future hydrologic events on wetlands, and (4) investigate the impacts of various remediation plans, and (5) investigate contaminant transport via groundwater into a wetland.

#### **How will these results be used?**

This model will be used by both NWRI scientists and external personnel to evaluate the role of groundwater in the hydrology and water quality of wetlands, lakes and river. The model can be used as both a scientific tool to research the processes affecting wetland quality, and within a regulatory framework to evaluate various groundwater-wetland interaction scenarios related to the protection, restoration, and management of wetlands. Although the model was originally written for groundwater-wetland interactions, it is now general enough for a variety of groundwater-surface water problems relating to rivers, lakes, and wetlands.

## **Guide de saisie pour les révisions du logiciel GW-WETLAND (version 2.0)**

### **Logiciel de simulation des interactions entre les eaux souterraines et les milieux humides**

S. G. Shikaze et A. S. Crowe

#### **RÉSUMÉ**

Les modèles de simulation par ordinateur sont des outils rentables et rapides qui nous permettent d'acquérir une bonne connaissance de l'environnement des eaux souterraines et des milieux humides. On peut en effet les utiliser pour : 1) étudier les processus qui surviennent dans cet environnement; 2) quantifier les débits au niveau de l'interface eaux souterraines-milieux humides; 3) prévoir les effets qu'auront d'éventuels événements hydrologiques sur les milieux humides; 4) évaluer les impacts de divers plans d'assainissement; 5) étudier le rôle des eaux souterraines dans le transport des contaminants vers les milieux humides. Le présent rapport décrit le modèle numérique GW-WETLAND que l'on peut utiliser pour simuler un débit transitoire d'eau souterraine et le transport de contaminants dans une coupe transversale bidimensionnelle selon divers scénarios impliquant des eaux souterraines et des milieux humides, y compris des conditions à l'interface eaux souterraines-milieux humides. Le modèle tient compte des fluctuations de la nappe phréatique, de la formation ou de la disparition des surfaces d'infiltration, de la variation spatiale et temporelle des débits d'infiltration et d'évapotranspiration dans la partie supérieure de la coupe transversale, de la présence de puits de pompage, de fossés ou d'une séquence sédimentaire hétérogène et des changements que subit le périmètre des milieux humides avec le temps et selon l'emplacement. Le modèle est capable de calculer les niveaux d'eau des nappes d'eau de surface selon les débits d'entrée ou de sortie des systèmes d'eaux souterraines et selon les précipitations. De plus, si l'utilisateur connaît l'historique des niveaux d'eau, il peut utiliser les niveaux des eaux de surface connus comme condition limite précise qui aura une incidence sur le régime des eaux souterraines. Le modèle peut aussi inclure des zones où l'on trouve des sources de contaminants associées aux étendues d'eaux de surface et, ainsi, évaluer le potentiel de contamination des eaux souterraines advenant une migration des eaux de surface vers une nappe aquifère adjacente. Bien qu'à l'origine on ait conçu le modèle pour qu'il simule les interactions entre les eaux souterraines et les milieux humides, il offre maintenant un champ d'application suffisamment vaste pour simuler divers problèmes mettant en cause les eaux souterraines et les eaux de cours d'eau, de lacs et de milieux humides. Qui plus est, le modèle est continuellement mis à jour pour l'on puisse l'utiliser avec de nouveaux scénarios.

## **Sommaire des recherches de l'INRE**

### **Titre en langage clair**

Les modèles de simulation par ordinateur nous permettent d'acquérir une bonne connaissance de l'interaction entre les eaux souterraines et les milieux humides.

### **Quel est le problème et que savent les chercheurs à ce sujet?**

Au cours des 200 dernières années, la plupart des milieux humides côtiers de la région des Grands Lacs ont été détruits par les activités humaines. Environnement Canada s'est donc engagé à protéger les milieux humides qui subsistent et à restaurer ceux qui sont dégradés. Or, on connaît très mal les liens qui existent entre les eaux souterraines et les milieux humides côtiers.

### **Pourquoi l'INRE a-t-il effectué cette étude?**

Les travaux entrepris font partie d'un programme d'Environnement Canada visant à protéger et à restaurer les milieux humides côtiers dans la région des Grands Lacs. Les scientifiques de l'INRE mènent des études sur le terrain et par modélisation afin d'évaluer le rôle joué par les eaux souterraines dans ces milieux. Ils possèdent une expertise considérable dans le domaine de l'hydrogéologie et de l'interaction entre les eaux souterraines et les eaux de surface. Par exemple, on a entrepris des études de modélisation par ordinateur au parc national de la Pointe-Pelée en vue d'évaluer le risque de transport de contaminants des installations septiques jusqu'aux marais par l'intermédiaire des eaux souterraines.

### **Quels sont les résultats?**

Les modèles de simulation par ordinateur sont des outils rentables et rapides qui nous permettent d'acquérir une bonne connaissance de l'environnement des eaux souterraines et des milieux humides. On peut en effet les utiliser pour : 1) étudier les processus qui surviennent dans cet environnement; 2) quantifier les débits au niveau de l'interface eaux souterraines-milieux humides; 3) prévoir les effets qu'auront d'éventuels événements hydrologiques sur les milieux humides; 4) évaluer les impacts de divers plans d'assainissement; 5) étudier le rôle des eaux souterraines dans le transport des contaminants vers les milieux humides.

### **Comment ces résultats seront-ils utilisés?**

Les scientifiques de l'INRE ainsi que le personnel externe utiliseront le modèle pour évaluer le rôle joué par les eaux souterraines dans l'hydrologie et la qualité de l'eau des milieux humides, des lacs et des cours d'eau. On peut utiliser le modèle comme outil scientifique afin de déterminer les processus qui influent sur la qualité des milieux humides et, dans un cadre réglementaire, d'évaluer divers scénarios d'interaction entre les eaux souterraines et les milieux humides visant la protection, la restauration et la gestion de ces milieux. Bien qu'à l'origine on ait conçu le modèle pour qu'il simule les interactions entre les eaux souterraines et les milieux humides, il offre maintenant un champ d'application suffisamment vaste pour simuler divers problèmes mettant en cause les eaux souterraines et les eaux de cours d'eau, de lacs et de milieux humides.

**Input Guide and Revisions for  
GW-WETLAND (Version 2.0):**

**A Computer Program to Simulate  
Groundwater-Wetland Interactions**

**Steven G. Shikaze**

**Allan S. Crowe**

National Water Research Institute  
Environment Canada  
Box 5050  
867 Lakeshore Road  
Burlington, ON  
L7R 4A6

March, 2005

## TABLE OF CONTENTS

<b>TABLE OF CONTENTS.....</b>	<b>III</b>
<b>ABSTRACT .....</b>	<b>V</b>
<b>1.0 INTRODUCTION.....</b>	<b>1</b>
<b>2.0 GOVERNING EQUATIONS AND BOUNDARY CONDITIONS .....</b>	<b>6</b>
2.1 Groundwater Flow .....	6
2.2 Wetland/Surface Water Model .....	9
2.3 Contaminant Transport .....	10
<b>3.0 NUMERICAL METHODS .....</b>	<b>12</b>
3.1 Mesh Generation and Modification .....	12
3.1.1 Limitations with the Mesh Generation Method.....	14
3.2 Contaminant Transport Methods .....	15
3.2.1 Deterministic-Probabilistic Mass Transport.....	16
3.2.2 Galerkin Method for Solution to the Advection-Dispersion Equation.....	19
3.2.3 Stability of the Contaminant Transport Methods .....	20
3.3 Internal Calculation of Water Levels in Surface-water Bodies .....	21
3.4 Contaminant Source Associated with a Surface-water Boundary .....	22
3.5 Drainage Ditches.....	22
3.6 Other Features.....	23
3.6.1 Pumping Wells .....	23
3.6.2 Spatially variable specific yield.....	23
3.6.3 Fluid Mass Balance .....	24
3.6.4 Seepage Faces.....	24
3.6.5 Dynamic memory allocation.....	24
<b>4.0 OPERATION OF THE MODEL.....</b>	<b>25</b>
4.1 Computing Requirements .....	25
4.2 Dimensioning of the Arrays.....	25
4.3 Input Operations.....	25
<b>5.0 ILLUSTRATED EXAMPLES .....</b>	<b>35</b>
5.1 Example 1: Contaminant Transport from a River Flood .....	35
5.2 Example 2: Water Levels in Surface-water Bodies that Change in Response to Fluxes.....	37

<b>6.0 REFERENCES .....</b>	<b>41</b>
<b>APPENDIX A - INPUT GUIDE.....</b>	<b>43</b>
A.1 General Notes.....	43
A.2 Input Guide .....	44
A.2.1 Program Title .....	44
A.2.2 Program Options.....	44
A.2.3 Observation Points.....	46
A.2.4 Grid/Boundary Condition/Time Stepping Input .....	46
A.2.5 Column Data.....	47
A.2.6 Drainage Ditches .....	47
A.2.7 Recharge/Evapotranspiration Data .....	48
A.2.8 Constant Head Data .....	48
A.2.9 Surface Water Body Data .....	49
A.2.10 Pumping Well Data .....	50
A.2.11 Stratigraphy Data .....	51
A.2.12 Particle Tracking Data .....	52
A.2.13 Advection/Dispersion Data.....	52
A.3 Output Files.....	53
A.4 Dimensioning of Arrays.....	55
A.5 Units.....	56
<b>APPENDIX B - INPUT DATA FILE FOR EXAMPLE 1.....</b>	<b>57</b>
<b>APPENDIX C - INPUT DATA FILE FOR EXAMPLE 2.....</b>	<b>67</b>



## ABSTRACT

Computer simulation models offer a cost-effective approach and rapid means of obtaining considerable insight into the groundwater - wetlands environment. Computer models can be used to (1) investigate processes occurring in this environment, (2) quantify fluxes across the groundwater-wetland interface, (3) predict the impact of future hydrologic events on wetlands, and (4) investigate the impacts of various remediation plans, and (5) investigate contaminant transport via groundwater into a wetland. This report describes a numerical GW-WETLAND 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, spatially and temporally varying infiltration and evapotranspiration fluxes along the top of the cross-section, the presence of pumping wells, installation of drainage ditches, a heterogeneous sedimentary sequence, and wetland shorelines that vary with time and location. The model is capable of either calculating the water levels in surface-water bodies in response to fluxes to/from the groundwater system as well as precipitation, or if the history of these water levels is known a priori, the user can set the surface-water-level history as a specified boundary condition that will influence the groundwater regime. The model also has the capability to include contaminant source zones that are associated with surface water boundaries to examine the potential for groundwater contamination as a result of the movement of water from the surface water body into the adjacent aquifer. Although the model was originally written for groundwater-wetland interactions, it is general enough for a variety of groundwater-surface water problems relating to rivers, lakes, and wetlands. Moreover, the model is constantly being updated in order to allow it to be used for new scenarios.

## 1.0 INTRODUCTION

Groundwater is an important component of the hydrology of 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.

Numerical models have been used in several groundwater-wetland studies. A steady-state, 2-D numerical model was used by Woo and Winter (1993) to evaluate the effect of permafrost as a confining layer on groundwater flow associated with wetlands. Winter (1999) used steady-state, 2-D and 3-D models to examine the role of surface topography, and subsurface stratigraphy on groundwater flow in the vicinity of wetlands. Steady-state, 3-D numerical models have been used to investigate the relationship between regional groundwater flow systems and bogs/fens in Northern Minnesota, U.S.A. (Siegel, 1983) and a wetland in East Anglia, U.K. (Gilver *et al.*, 1993). Reeve *et al.* (2000) used a steady-state, 3-D simulation to investigate factors influencing vertical groundwater flow beneath peat domes. Transient groundwater-wetland models have also been developed and applied by Restrepo *et al.* (1998) and Mansell *et al.* (2000). Restrepo *et al.* (1998) coupled 3-D transient groundwater flow, transient sheet flow across a flooded vegetated area (wetland), and flow through a stream channel. Mansell *et al.* (2000) developed and applied a transient, coupled saturated-unsaturated, 2-D model to study the link between groundwater and cypress ponds in the Coastal Plain pine forest region, U.S.A.

These groundwater-wetland modelling studies demonstrate the value of numerical models in understanding groundwater flow in the vicinity of wetlands. However, with the exception of the models of Restrepo *et al.* (1998) and Mansell *et al.* (2000), previous modelling studies have used steady state simulations, which do not permit analysis of responses of the groundwater-wetland system to changing hydrological and meteorological conditions. Existing groundwater models are generally limited in their ability to handle realistic wetland boundary conditions because they were not designed for this purpose.

For example, these existing models do not simulate the increases and decreases the lateral extent of a wetland over the shore as its water level rises and falls, and do not simulate the formation of seepage faces at the wetland-shore interface. In addition, previous modelling studies did not include simulation of contaminant transport through a groundwater flow system to a wetland, or quantify both groundwater and solute flux across a transient groundwater - wetland interface.

Groundwater flow systems in the vicinity of wetlands and groundwater-wetland interactions are complex due to a variety of factors that influence this type of environment. A groundwater-wetland model must account for these factors. The model must be able to simulate transient groundwater flow in a heterogeneous sedimentary environment, in which the distribution of hydraulic head and the position of the water table vary in response to infiltration, evapotranspiration, surface water (wetland) conditions, and external factors (pumping wells, drainage ditches). The model must also account for changes in infiltration and evapotranspiration over time and space. Because the wetland will affect groundwater conditions, the model must simulate transient wetland conditions, including the fluctuating water levels within the wetland, changes in aerial extent due to a fluctuating shoreline (i.e., changing location of shorelines as water levels rise or fall), and formation and disappearance of seepage faces. With respect to contaminants, the model must simulate contaminant transport within the groundwater flow system, contaminant movement across the wetland-aquifer boundary (both into the wetland from the aquifer and vice versa), introduction of contaminants from a variety of source locations, and transient contaminant input conditions.

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 is capable of (1) simulating groundwater flow systems in the vicinity of wetlands; (2) quantifying water and solute fluxes across the groundwater-wetland interface; (3) simulating contaminant transport within the groundwater flow system and release to a wetland; and (4) providing insight into various processes occurring in groundwater-wetland environments. 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 and marsh (fluctuate both vertically and laterally in response to changes in the size and shape of the wetland or lake level cycles).

This model 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 be used to study the groundwater-wetland environment of a wetland in a depression of 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

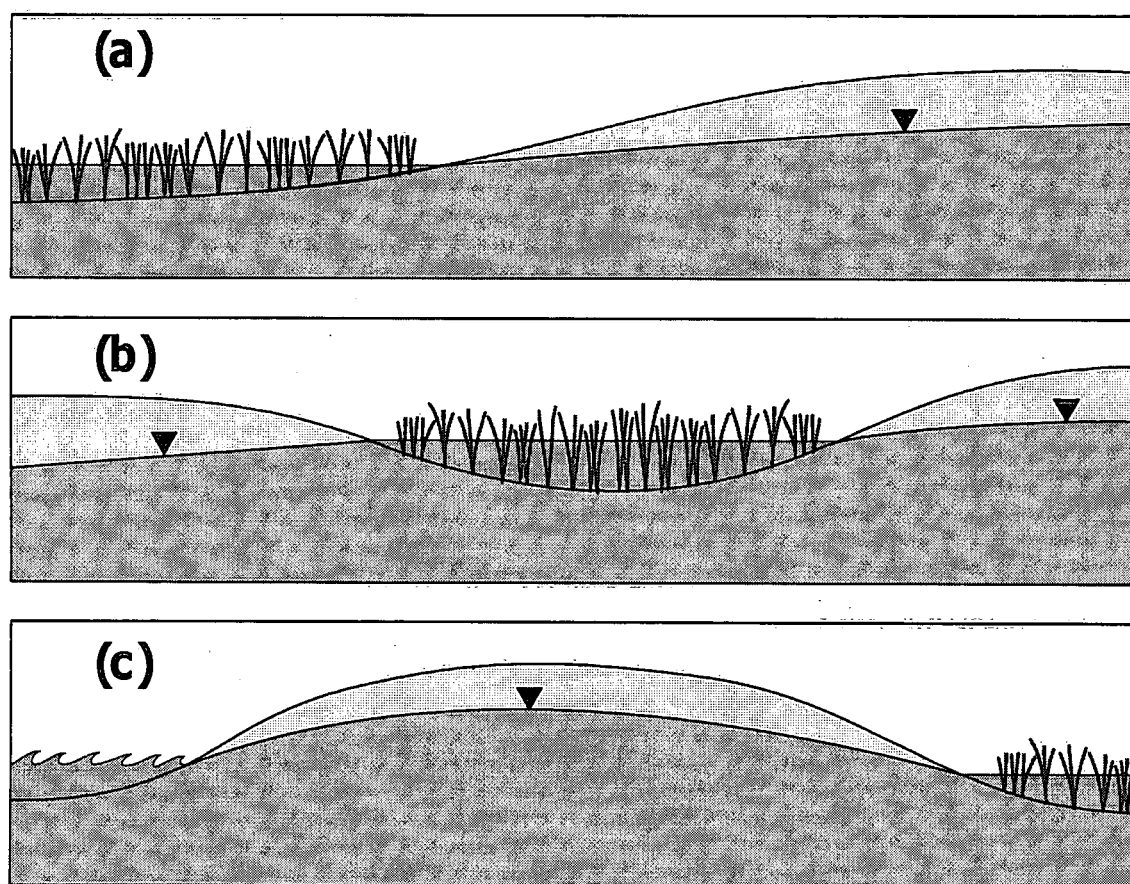


Figure 1.1. Three scenarios that can be simulated with GW-WETLAND, (a) a groundwater-wetland interface, (b) a groundwater-wetland environment where the wetland is in a depression, and (c) groundwater flow between two-wetlands or a wetland-lake environment.

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 which results in 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.

GW-WETLAND is a numerical model that can be used to simulate fully-saturated, transient groundwater flow and contaminant transport in a 2-D cross section for a variety of groundwater-wetland scenarios as well as other problems such as the response of an aquifer to a short-term flood of a river, or the presence of a drainage ditch. The model makes use of the finite-element method to solve the transient groundwater flow equation, and solute transport can be solved either by deterministic-probabilistic particle tracking or by finite-element solution to the advection-dispersion equation. For groundwater flow, several different options are available for boundary conditions, including specified hydraulic head boundaries (e.g., seasonal changes in the water levels in a river), and precipitation and evapotranspiration that can change in time.

One of the unique features of the model is the ability to simulate a water table boundary that can change both spatially and temporally. Although the model does not include flow in the unsaturated zone, the top surface of the finite-element mesh is allowed conform to the position of the water table (Crowe *et al.*, 1999). Specifically, if the water table rises above a given tolerance, new nodes and elements can be added such that the existing stratigraphy is not distorted by stretching elements in the vertical direction. Conversely, if the water table falls below a specified tolerance, nodes and elements are removed from the mesh while avoiding distortion of the finite element mesh, which frequently results from compressing elements (Figure 1.2). Also, because boundary conditions can change at any time during the simulation, the position of the water table can also change in time. As a result, the finite element mesh may change from one time-step to the next.

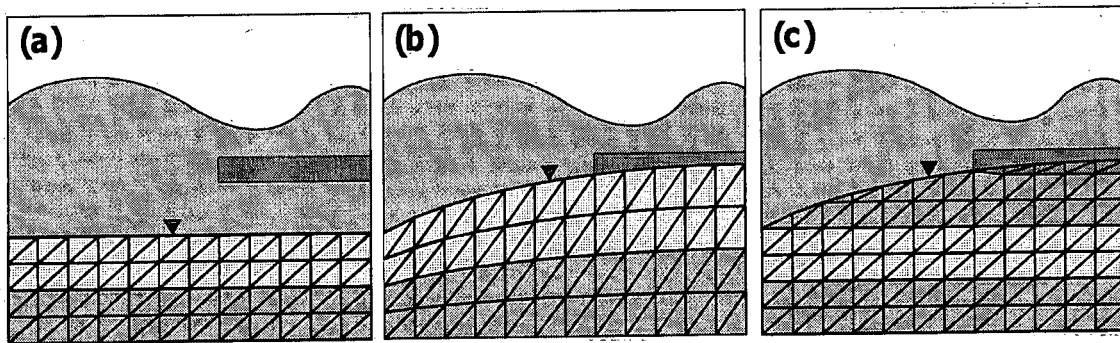


Figure 1.2. The geometry of the hydrostratigraphic units modelled by the regeneration of a finite element mesh as the water table rises. (a) Initial grid and hydrostratigraphy, (b) grid at some later time generated with a method that stretches all rows of elements, and (c) grid at some later time generated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

Since the GW-WETLAND model was introduced, changes have been made to the model in order to make the program more flexible with more options. This report provides a brief description of the changes to the numerical model GW-WETLAND including a detailed chapter on the revised input requirements for the program, a line-by-line input guide for the input data file, and illustrative examples with sample input and output files.

Although this computer program was originally created to simulate groundwater flow and contaminant transport in the vicinity of a wetland, it is now flexible enough to simulate various other scenarios. For example, when a river floods, flood waters may contain contamination that can act as a source for groundwater contamination, even if the duration of the flood is relatively short. The short-term, high water levels in the river can be simulated along with a contaminant source in the river itself (Shikaze and Crowe, 1998). Another scenario is groundwater flow in the vicinity of a drainage ditch, which is typically used to lower the water table. Also, the ditch can have water levels that can vary in time. In this case, the model can be used to predict the response of the water table to the presence of the drainage ditch as well as the water level in the ditch (Crowe *et al.* 2000). The model has also been used to simulate groundwater changes at a shoreline in response to lake-level changes and wave run-up due to storms (Crowe *et al.* 2005). These simulations were undertaken to determine if contaminants in lake water can infiltrate into the groundwater regime below the swash zone during a storm, and if so, how long will the contaminants remain once lake levels have dropped and the groundwater flow system has returned to pre-storm conditions.

This user's guide is set up to provide some relevant background material, a detailed description of the procedure required to set up an input data file, as well as some example test cases and data files. Chapter 2 describes the governing equations for groundwater flow, contaminant transport and the wetland model. Chapter 3 describes the numerical methods used to implement the governing equations. Chapter 4 explains the input procedure to aid in setting up the input data file. Chapter 5 provides two example problems that illustrate some of the features of the model. These include (1) a simulation of contaminant transport from a river under flood conditions, and (2) a demonstration of water levels in a surface water body that changes in response to calculated fluxes. A step-by-step input guide for the input data file is provided in Appendix A.

## 2.0 GOVERNING EQUATIONS AND BOUNDARY CONDITIONS

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 often referred to as a 'free-surface problem', in which the distribution of hydraulic head and the elevation of the water table are unknown.

### 2.1 Groundwater Flow

The model presented here is actually an extension of an existing code which was developed to simulate groundwater flow system changes in response to dewatering at strip mines and subsequent flooding of the abandoned pits (Schwartz and Crowe, 1985, 1987). The governing equations and boundary conditions upon which the code is based have been described in detail by Neuman and Witherspoon (1971), and are briefly discussed here.

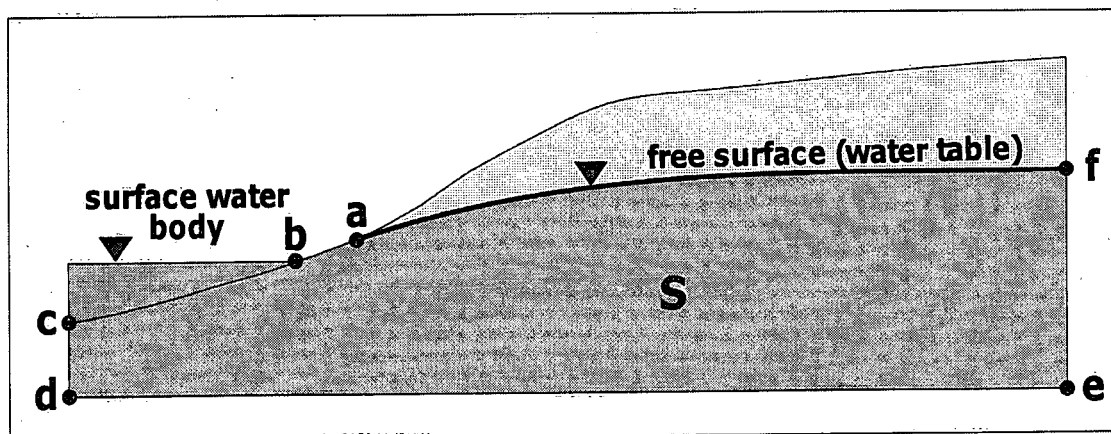


Figure 2.1. Schematic cross section of a typical computational domain.

Groundwater flow is simulated in only the saturated zone within a two-dimensional, heterogeneous cross section (S in Figure 2.1). The distribution of hydraulic head within the cross section is solved with the transient groundwater flow equation:

$$\frac{\partial}{\partial x} \left( K_x \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial z} \left( K_z \frac{\partial h}{\partial z} \right) = S_s \frac{\partial h}{\partial t} \quad (1)$$

where:  $K_x$  = horizontal saturated hydraulic conductivity (L/T)

$K_z$  = vertical saturated hydraulic conductivity (L/T)  
 $h$  = hydraulic head (L)  
 $S_s$  = specific storage coefficient (L<sup>-1</sup>)  
 $x, z$  = coordinates (L)  
 $t$  = time (T)

Boundary conditions along the left and right ends and the base of the cross section (c-d, d-e, e-f, respectively in Figure 2.1) are no-flow boundaries, respectively:

$$\left. \frac{\partial h}{\partial x} \right|_{x=0} = 0 \quad \left. \frac{\partial h}{\partial x} \right|_{x=s} = 0 \quad \left. \frac{\partial h}{\partial z} \right|_{z=0} = 0 \quad (2)$$

where:  $x=0$  = left boundary of the cross section  
 $x=s$  = right boundary of the cross section  
 $z=0$  = base of the cross section

The upper boundary of the saturated zone (i.e., top of the computational domain) is a water table or free surface (a-f in Figure 2.1), and is allowed to rise and fall in response to the distribution of hydraulic head within the saturated domain and boundary conditions. This ensures that at all times during a simulation, the elevation of the water table matches the head of the water table:

$$\phi(x, z, t) = h(x, z, t) \quad (3)$$

where:  $\phi$  = elevation of the node at the free surface (L)

The surface of the groundwater flow system can have two types of boundary conditions; a specified head (1<sup>st</sup> type or Dirichlet) boundary and a specified flux (2<sup>nd</sup> type or Neumann) boundary. A specified head boundary condition at the water table sets the value of hydraulic head at the water to a predetermine value, which is equal to the elevation of the water table at that point.

$$h(x, z, t) = H(x, z, t) = z(x) \quad (4)$$

where:  $H$  = the prescribed value of hydraulic head at the water table (L)  
 $z$  = elevation of the node at the water table (L)

To maintain the position of the water table at a given elevation, the amount of flux across a specified head boundary will be automatically determined according to the gradient between specific value of hydraulic head at the water table and the value of hydraulic head at the adjacent node. A specified flux boundary at the water table will allow the groundwater flow system to gain water through infiltration and lose water through evapotranspiration. Both infiltration and evapotranspiration are variable with



time and location along the water table, including times when there is no infiltration or evapotranspiration:

$$K_x \frac{\partial h}{\partial x} + K_z \frac{\partial h}{\partial z} = \left[ I - ET - S_y \frac{\partial h}{\partial t} \right] \quad (5)$$

where:  $I$  = rate of infiltration across the free surface (L/T)  
 $ET$  = rate of evapotranspiration across the free surface (L/T)  
 $S_y$  = specific yield

The value of the hydraulic head at the water table, and hence the elevation of the water table will be automatically determined by the specified flux crossing the water table.

The model also simulates the formation and removal of seepage-face boundaries at the edge of the wetland (a-b in Figure 2.1). Seepage faces form when the water table rises to the ground surface, and there is an outward flux from the saturated domain across this boundary. The value of hydraulic head is set equal to the elevation of the ground surface:

$$h(x, z, t) = Z(x) \quad (6)$$

where:  $Z$  = elevation of the ground surface (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:

$$h(x, z, t = 0) = h_0(x, z) \quad (7)$$

$$\phi(x, z, 0) = \phi_0(x, z) \quad (8)$$

where:  $\phi$  = the elevation of the free surface (a-f in Figure 2.1) above a datum (L),  
 $\phi_0$  = the initial elevation of the free surface (L),  
 $h_0$  = the initial hydraulic head throughout the domain (L).

Because both the distribution of hydraulic head within the saturated zone and the geometry of the computational domain (i.e., position of the water table) are unknown at each time step, the solution for is non-linear and requires an iterative approach. During each time step, hydraulic heads within the saturated domain and along the water table are calculated and the heads along the water table are compared to the elevation of the nodes. The elevation of each node along the water table is then adjusted. A solution is achieved when, for all nodes along the water table, the residual between the

value of hydraulic head and the elevation of the node is less than a preset convergence criterion (Crowe *et al.*, 1999):

$$residual = \sum |\phi(x, z, t) - h(x, z, t)| \quad (9)$$

In the following simulations a value convergence criterion of 0.01 m was selected. Although a smaller value will produce a numerical solution that is slightly more accurate, it will increase the number of iterations and the time required for convergence during a time step.

## 2.2 Wetland/Surface Water Model

The model can simulate a variety of single or multiple surface water bodies along the surface of the groundwater flow system (e.g., a wetland, lake, or river) within a 2-D cross sectional domain. The surface water bodies may expand and contract with respect to both water levels and lateral extent along the shore line due to changing meteorological, hydrological, and groundwater conditions.

Surface water bodies are represented by a transient specified head boundary along the top of the computational domain where it intersects the ground surface (b-c in Figure 2.1). The values of hydraulic head are equal to the water level of the surface water body and may change at each time step:

$$H(x, z, t) = L(t) \quad (10)$$

where:  $L$  = elevation of the surface of water in the surface water body ( $L$ )

As the surface water body rises, the nodes along the surface of the computational domain are switched from free nodes to specified head nodes to represent the lateral extent of the surface water body. Conversely, as water levels in the surface water body fall, nodes at an elevation above the new surface of the surface water body are converted from specified head to free nodes. If the water level of the surface water body falls below the ground surface (i.e., the surface water body becomes dry), all specified head nodes along the ground surface become free nodes (equation (3)).

Fluctuations in the surface area and water levels in the surface water body can be simulated in two ways, depending upon whether (a) the user wishes to use a long-term and known record of water levels of the surface water body to control groundwater flow conditions, or (b) the user wishes to use the groundwater flow conditions to simulate surface water body water level fluctuations (no available long-term record). In the first case, the known history of water level fluctuations can be entered, thus defining the water level of the surface water body at each time step throughout the simulation as a transient specified head boundary. In this case the user-defined water levels of the surface water body control groundwater fluxes across the wetland-aquifer boundary.

In the second case, the water level of the surface water body at each time step is not defined by the user but calculated by the model based on groundwater fluxes across the surface water body-aquifer boundary. Under these conditions, the change in the volume of water in the surface water body per unit width is calculated from precipitation and evaporation occurring at the surface of the surface water body and groundwater flux across the sediment-surface water body interface (i.e., into and out of the groundwater flow system):

$$\frac{dV}{dt} = \sum_{i=1}^{nelm} q_i \cdot \Gamma_i^e + P_t \cdot A_t - E_t \cdot A_t \quad (11)$$

where:  $V$  = volume of the surface water body per unit width during at time step ( $L^3/L$ )  
 $q_i$  = groundwater flux across element at the surface water-sediment interface ( $L/T$ )  
 $\Gamma_i^e$  = area per unit width of an element at the surface water sediment interface ( $L^2/L$ )  
 $nelm$  = number of elements along the surface water-groundwater domain interface  
 $P_t$  = precipitation falling directly on the surface water body at time  $t$  ( $L/T$ )  
 $A_t$  = surface area of the surface water body per unit width at time  $t$  ( $L^2/L$ )  
 $E_t$  = evaporation directly from the surface of the surface water body at time  $t$  ( $L/T$ )

Based on the user-defined geometry of the base of the surface water body and its maximum possible water level, the model calculates a water-level-to-volume curve. This curve is then used to adjust the elevation of the surface of the water body and its surface area given the change in volume of water within the surface water body. Specified head nodes are then added or removed as described above.

### 2.3 Contaminant Transport

Both advective and dispersive contaminant transport are simulated by a numerical solution to the two-dimensional advection-dispersion equation:

$$\frac{\partial c}{\partial t} = -\frac{\partial}{\partial x}(v_x c) - \frac{\partial}{\partial z}(v_z c) + \frac{\partial}{\partial x}\left(D_{xx} \frac{\partial c}{\partial x} + D_{xz} \frac{\partial c}{\partial z}\right) + \frac{\partial}{\partial z}\left(D_{zx} \frac{\partial c}{\partial x} + D_{zz} \frac{\partial c}{\partial z}\right) - \sum k_n c \quad (12)$$

where:  $c$  = concentration of the contaminant ( $M/L^3$ )  
 $v_x, v_z$  = the average linear groundwater velocity in the x and z directions ( $L/T$ )  
 $D$  = hydrodynamic dispersion tensor ( $L^2/T$ )  
 $k_n$  = the chemical rate constant for reaction  $n$ .

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

$$v_x = \frac{K_x}{\theta} \left[ \frac{\partial h}{\partial x} \right] \quad v_z = \frac{K_z}{\theta} \left[ \frac{\partial h}{\partial z} \right] \quad (13)$$

where:  $\theta$  = porosity

The amount of mass in the system can be reduced over time through the non-reversible first-order degradation reaction:

$$\frac{dc}{dt} = -k \cdot c \quad (14)$$

where:  $k$  = degradation rate ( $T^{-1}$ )

The degradation rate is defined as:

$$k = \frac{0.693}{t_{1/2}} \quad (15)$$

where:  $t_{1/2}$  = the contaminant half-life (T)

### 3.0 NUMERICAL METHODS

#### 3.1 Mesh Generation and Modification

The two-dimensional form of the transient groundwater flow equation (equation (1)), subject to initial and boundary conditions (equations (2) – (8)), is solved in a vertical cross section using a standard finite-element technique. The finite element equations are formulated using the standard Galerkin method (Pinder and Gray, 1977) utilizing triangular finite elements. The algorithm for generating the finite element grid is governed by:

- 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  $\phi(x,t) = h(x,t)$ ;
- hydrostratigraphic unit boundaries 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).

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 3.1a, 3.1g). 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 3.1b, 3.1h). All remaining elements below the uppermost row of elements remain at a constant vertical spacing of  $\Delta z$ . 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 (9)). 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 3.1c, 3.1e, 3.1i, and 3.1k).

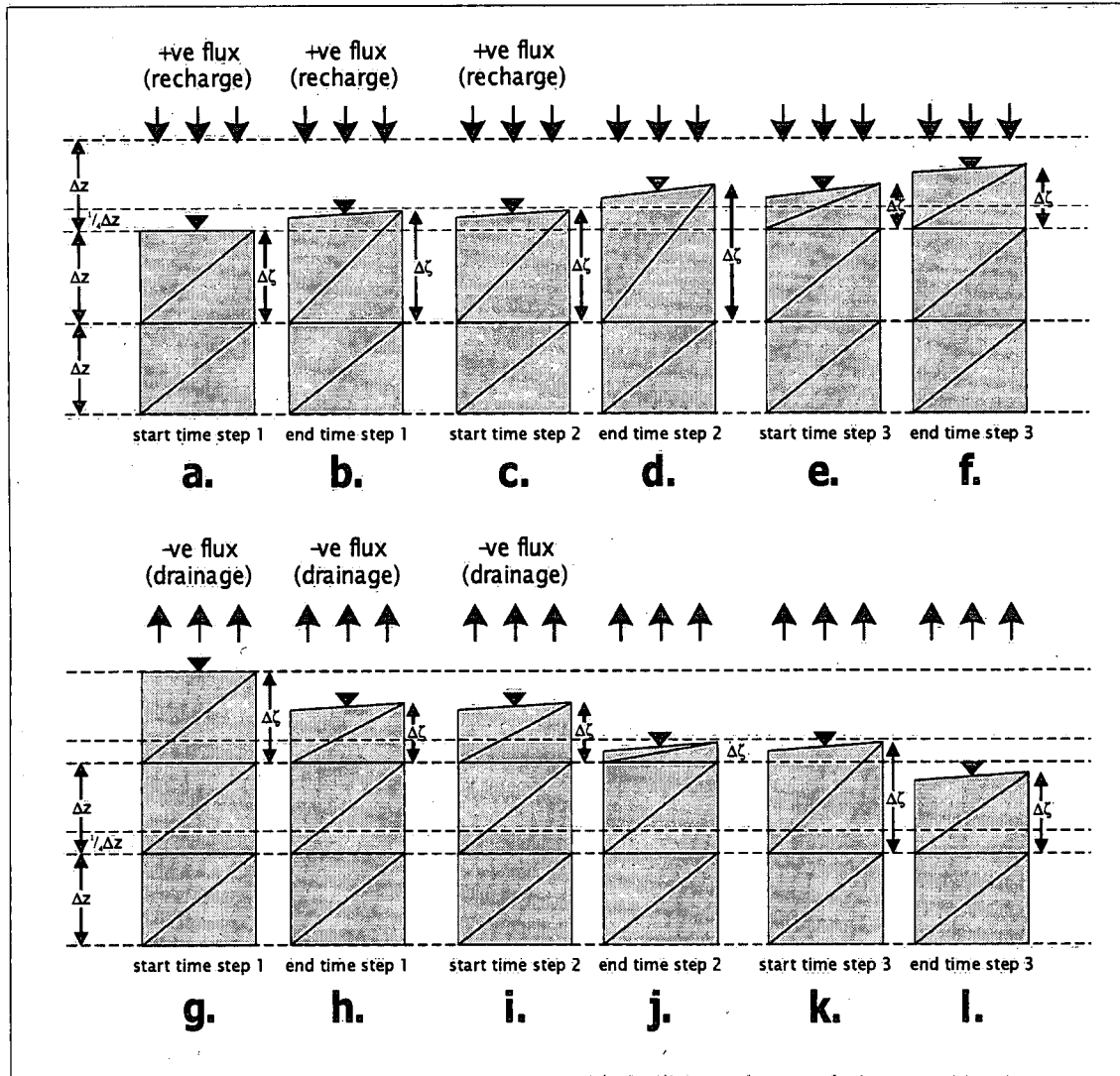


Figure 3.1. Schematic diagram of the development of a new mesh. a-f shows the addition of new nodes and elements; g-l shows the removal of nodes and elements.

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  $\frac{1}{4}\Delta z$  beyond the regular vertical grid spacing ( $\Delta\zeta > \frac{5}{4}\Delta z$ ), we form a new node and a new element. The new node is inserted at the regular  $\Delta z$  spacing, and the new element is inserted along the water table with a vertical element spacing of  $\Delta\zeta_{\text{new}} = \Delta\zeta_{\text{old}} - \Delta z$  (Figure 3.1e). If an element stretches less than  $\frac{1}{4}\Delta z$  beyond the regular grid spacing ( $\Delta\zeta < \frac{5}{4}\Delta z$ ), only the top two elements are stretched, and a new node is not inserted (Figure 3.1b).

Similarly, if a node at the water table declines by more than  $\frac{3}{4}\Delta z$  of the regular grid spacing ( $\Delta \zeta < \frac{1}{4}\Delta z$ ), the node immediately below this water table node is removed (Figure 3.1k). If the decline of a water table node is less than  $\frac{3}{4}\Delta z$ , 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 3.1h).

Because all elements, except those at the water table, are maintained at the original vertical grid spacing of  $\Delta z$ , unit boundaries remain unchanged (see Figure 3.1c). 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  $\frac{1}{4}\Delta z$ . 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 accurately represented. The error resulting from this slight misrepresentation is small. Once the new elements form, element boundaries will be placed at the proper  $\Delta z$  spacing. Hence, the error will likely be insignificant.

### 3.1.1 Limitations with the Mesh Generation Method

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  $\frac{1}{4}\Delta z$ . 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  $\frac{3}{4}\Delta z$ , 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.

Two solutions are provided 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  $\frac{1}{4}\Delta z$  to  $\frac{1}{10}\Delta z$ , and the criterion for removing an element is increased from  $\frac{3}{4}\Delta z$  to  $\frac{9}{10}\Delta z$  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 occurring. 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.

### 3.2 Contaminant Transport Methods

The program solves the solute transport equation with two different methods. The first method is based on a particle tracking method or also known as a deterministic-probabilistic solute transport model (Ahlstrom *et al.*, 1977; Schwartz, 1978; Schwartz and Crowe, 1980, Prickett *et al.*, 1981). 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 (Pinder and Gray, 1977) provides a direct solution to the advection-dispersion equation, whereby advection and dispersion are solved simultaneously.

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 will provide a more stable solution because it is not subject to constraints that restrict the Galerkin method such as time step or element shape (i.e., the Peclet and Courant stability criteria). This routine can also be used in cases where dispersion is completely neglected (i.e., particle tracking). In addition the particle tracking method can produce the path of advective transport of the contaminant via the particle locations at each time step, whereas the Galerkin method produces only a distribution of solute concentrations within all of the elements. The advantage of the Galerkin method is that dispersion is calculated directly using both dispersivity of the porous media and solute diffusion as separate input parameters, whereas dispersion in the particle tracking method is calculated as a 'random' component utilizing only dispersion coefficients that could incorporate both dispersivity and diffusion. Hence, dispersion is more accurately represented with the Galerkin method than with the particle tracking method because it is not subject to a random component. This in turn means the Galerkin method produces a better estimate of the distribution of solute concentrations within a cross section. Also, in our code, the Galerkin method can include simple retardation processes.



### 3.2.1 Deterministic-Probabilistic Mass Transport

The particle tracking method does not directly solve the mass transport equation, but simulates the spread of a dissolved contaminant in groundwater. In this method, contaminants are represented as particles, each with an x- and z-coordinate and each with an assigned mass. The total mass of all the particles represents the total mass of the contaminant. The particles are added to the system at a source location within the saturated domain.

The first step with this method is to solve for the groundwater velocities in each of the triangular finite elements using equation (13) (Figure 3.2a). Based on the horizontal and vertical groundwater velocities, a net groundwater velocity in the direction of groundwater flow is calculated by:

$$u = \sqrt{v_x^2 + v_z^2} \quad (16)$$

where:  $u$  = the net groundwater velocity in the direction of groundwater flow (L/T)

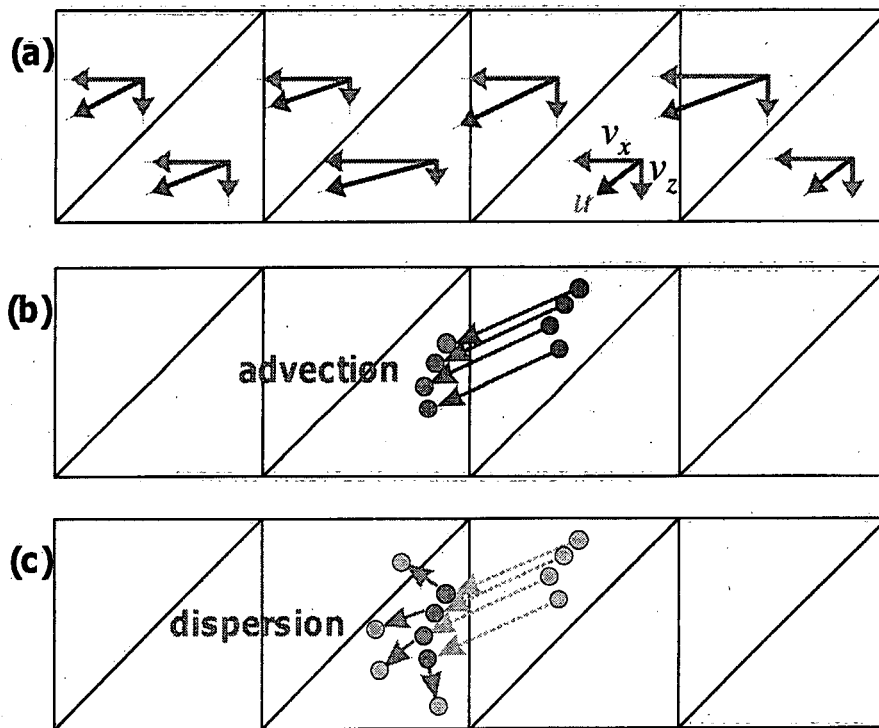


Figure 3.2. Schematic diagram illustrating the movement of particles through (a) the groundwater velocity field, by (b) advective transport and (c) dispersive transport.

Next, advection transport is simulated through the movement of the particles through time within the transient groundwater flow field. Each particle moves in the direction of groundwater flow and a distance as defined by the magnitudes of horizontal ( $v_x$ ) and vertical ( $v_z$ ) velocities, and the

direction of the hydraulic gradient as calculated for each of the triangular finite elements (equation (13)), as well as the size of the time step (Figure 3.2b). This advective displacement represents the deterministic component of contaminant transport and is calculated by:

$$x_t = x_{t-1} + v_x \Delta t \quad (17)$$

$$z_t = z_{t-1} + v_z \Delta t$$

where:  $x_t$  = the new  $x$  (horizontal) position of the particle due to advection (L)  
 $x_{t-1}$  = the  $x$  position of the particle during the previous time step (L)  
 $z_t$  = the new  $z$  (vertical) position of the particle due to advection (L)  
 $z_{t-1}$  = the  $z$  position of the particle during the previous time step (L)  
 $\Delta t$  = the size of the time step (T)

After the particles have moved due to advective transport, the particles are moved by dispersion. Hydrodynamic dispersion is incorporated into the particle motion by adding a random component to the advection particle motion (Figure 3.2c). This random motion is related to the dispersive characteristics of the porous medium which represents the random tortuosity of the pore space. The new particle locations arising from dispersion are calculated by (Ahlstrom *et al.*, 1977):

$$x_t = x_t^* + x' \cdot \frac{v_x}{u} + z' \cdot \frac{v_z}{u} \quad (18)$$

$$z_t = z_t^* + x' \cdot \frac{v_z}{u} + z' \cdot \frac{v_x}{u}$$

where:  $x_t$  = new particle position in the  $x$  direction due to dispersion (L)  
 $x_t^*$  = the horizontal particle position due to advection, defined as above (L)  
 $z_t$  = new particle position in the  $z$  direction due to dispersion (L)  
 $z_t^*$  = the vertical particle position due to advection, defined as above (L)  
 $x'$  = particle displacement in the  $x_t$  direction due to dispersion as defined below (L)  
 $z'$  = particle displacement in the  $x_t$  direction due to dispersion as defined below (L)

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

$$x' = \sqrt{24D_l \cdot \Delta t} \cdot (0.5 - \Omega) \quad (19)$$

$$z' = \sqrt{24D_t \cdot \Delta t} \cdot (0.5 - \Omega)$$

where:  $D_l$  = the longitudinal dispersion coefficient ( $L^2/T$ )  
 $D_t$  = the transverse dispersion coefficient ( $L^2/T$ )  
 $\Omega$  = a random number between 0 and 1

The longitudinal and transverse dispersion coefficients are defined as:

$$D_l = \alpha_l |u| \quad (20)$$

$$D_t = \alpha_t |u|$$

where:  $\alpha_l$  = the longitudinal dispersivity (L)  
 $\alpha_t$  = the transverse dispersivity (L)

When only advection and dispersion act on the contaminants the quantity of mass associated with each particle remains constant over time. However, when degradation occurs it is necessary to reduce the quantity of mass assigned to each reference particle at the end of each time step:

$$m_{i,t} = m_{i,t-1} \cdot e^{-k\Delta t} \quad (21)$$

( where:  $m_{i,t}$  = the mass of particle i during the present time step (M)  
 $m_{i,t-1}$  = the mass of particle i during the previous time step (M)  
 $k$  = the degradation rate ( $T^{-1}$ )  
 $\Delta t$  = size of the time step (T)

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:

$$C_c = (\sum p_{i,0} m_{i,0} + \sum p_{j,in} m_{j,in} - \sum p_{k,out} m_{k,out}) / V\theta \quad (22)$$

where:  $C_c$  = the concentration of a contaminant in the cell ( $M/L^3$ )  
 $p_{i,0}$  = the number of particles in a cell at the start of the time step  
 $p_{i,in}$  = the number of particles entering the cell during the time step  
 $p_{k,out}$  = the number of particles leaving the cell during the time step  
 $m$  = the mass of a particle (M)  
 $V$  = the volume of the cell ( $L^3$ )  
 $\theta$  = the porosity

The contaminant source may be placed at a variety of locations, such as at the water table (e.g., may represent a spill on the surface), along the entire wetland-groundwater interface (e.g., represents contaminated surface water), and within the saturated domain (e.g., at a well). However, only a fixed-source concentration (1<sup>st</sup> type or Dirichlet) boundary condition can be used, but the source may be turned on or off during a simulation. When the wetland-groundwater interface is specified as the

contaminant source, the size of the source zone will change in proportion to fluctuations in the size (elevation, shoreline) of the wetland.

Output from the model, using this method, consists of both contaminant concentrations within each element, and the location of the particles. Because the x- and z-position of each individual particle is known within the domain, the distribution of contaminant mass (e.g., the distribution of contaminant particles) can also be determined. This can be used to track the advective and dispersive transport through the groundwater regime over time.

### 3.2.2 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:

$$\frac{\partial c}{\partial t} = - \frac{\partial}{\partial x_i} \left( \frac{v_i}{R_i} c \right) + \frac{\partial}{\partial x_i} \left( \frac{D_{ij}}{R} \frac{\partial c}{\partial x_j} \right) - \lambda c \quad (23)$$

where:  $\lambda$  = the degradation constant ( $T^{-1}$ )

In this solution, the hydrodynamic dispersion tensor is given by:

$$\begin{aligned} D_{xx} &= D^* \tau + \frac{\alpha_l v_x^2}{|u|} + \frac{\alpha_t v_z^2}{|u|} \\ D_{zz} &= D^* \tau + \frac{\alpha_l v_z^2}{|u|} + \frac{\alpha_t v_x^2}{|u|} \\ D_{xz} &= D_{zx} = (\alpha_l - \alpha_t) \frac{v_x v_z}{|u|} \end{aligned} \quad (24)$$

where:  $D^*$  = the free-solution diffusion coefficient ( $L^2/T$ )  
 $\tau$  = the tortuosity of the porous medium

Currently, only first-type, or Dirichlet boundary condition can be specified in the program. This allows 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.

### 3.2.3 Stability of the Contaminant Transport Methods

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:

$$\begin{aligned} P_x &= \frac{v_x \Delta x}{D_{xx}} \\ P_z &= \frac{v_z \Delta z}{D_{zz}} \end{aligned} \tag{25}$$

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:

$$\begin{aligned} C_x &= \frac{v_x \Delta t}{\Delta x} \\ C_z &= \frac{v_z \Delta t}{\Delta z} \end{aligned} \tag{26}$$

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

$$A = \frac{D_{xx} \Delta z^2}{D_{zz} \Delta x^2} \tag{27}$$

The Peclet and Courant 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. Although no strict guidelines for the aspect ratio exist, experience has shown that the aspect ratio should fall within the range of 0.01 to 100.

### 3.3 Internal Calculation of Water Levels in Surface-water Bodies

The user has two options relating to water levels in surface-water bodies: (1) if the water-level history of a surface-water boundary is known, the user can specify this history as a known, time-dependent boundary condition for the water body or (2) the user can let the model calculate the transient water levels in the surface-water body, in response to fluxes at the groundwater-surface water interface as well as precipitation and evaporation fluxes. In the first case, the user must enter values for these water levels, as either constant or variable in time. Thus, the user must always know the water level in the surface water body before the simulation commenced.

Alternatively, the user can allow the model to calculate, at each time step, the water level of the surface-water body. This calculated water level will rise or fall depending on water fluxes along the groundwater-surface water interface as well as precipitation, all of which can vary in time. If the user chooses to allow the program to calculate water levels, and once the user specifies the location of a surface water body, the program will calculate a function for the volume-to-water-level of the surface water body. This function is based on the geometry of the base of the surface-water body, and volumes are calculated for vertical increments in water level of  $0.25\Delta x$ . This function is only calculated once, at the start of the simulation, and is used at the end of each time step to calculate the new water level, as based on the volumetric fluxes to/from the groundwater flow system as well as precipitation.

At the end of each time step (once hydraulic heads are calculated), the program calculates water fluxes along the interface between the groundwater flow system and the surface water boundary. Given the time-step size, a volume of water added to (or removed from) the surface water body (from/to groundwater) is determined. In addition to this, a volume of water is added to the surface water body from precipitation. These new volumes are added to (or subtracted from) the existing volume of water in the surface water body, and from this new volume, a new water level is determined from the volume-to-water-level function for the surface water boundary. This new water level is then used as the new boundary condition for the next time step, with constant head nodes being added or removed depending on the new value of head in the surface water body. The program then proceeds to the next time step.

The user must specify an initial and maximum water level for the surface water body. In the case of a rising water level, once this maximum is reached, the water level can rise no further. The water level can, however, decline from this maximum level if conditions are such to allow a water-level drop. The water level in the surface water body may also decline to the point where the water body dries completely. If this occurs, there will be no constant head nodes associated with this surface water body. If conditions change, this water body can resaturate, and constant head nodes will be added.

Also, the user enters values for recharge for the simulation and these values can change in both space and time. These recharge values are also used as precipitation fluxes to the surface-water bodies.

### **3.4 Contaminant Source Associated with a Surface-water Boundary**

The user can specify whether or not a surface-water boundary is a contaminant source zone. For example, this could be used to evaluate the potential for a river to contaminate an adjacent aquifer (Shikaze and Crowe, 1998). The user has two options by which to simulate solute transport. Advantages and disadvantages each option are the same as presented in section 2.1.4.

The first solute transport option is particle tracking. If this option is chosen, particles are added to each element that is in contact with the surface-water boundary for a length of time that is user-specified. As the water level in the surface-water body rises, the number of elements that represent the contaminant source zone may increase (the model will detect this automatically). Moreover, when the water level in the surface-water body falls, the number of contaminant-source elements may be reduced. The user specifies a single value for the number of particles that are to be added to each cell (Note: each cell consists of two triangular finite elements).

The second solute-transport option is the Galerkin solution to the advection-dispersion equation. In this formulation, the same mesh that was used for the groundwater flow solution is used for the advection-dispersion solution at each time step (and this mesh, as mentioned previously, can change in time). If the user selects this option, nodes associated with the surface-water body are set as first-type concentration nodes for a period of time that is specified by the user. Also, as the water level in the surface-water body rises or falls, the number of first-type concentration nodes associated with the surface-water body will increase or decrease, respectively. The user can specify the start and end times at which the contaminant source zone will be turned on and off, respectively.

The user can specify either method for solute transport from a surface-water boundary. Results (i.e., cell concentrations for particle tracking; nodal concentrations for finite-element solution) can be viewed in a manner as for the particle-tracking or advection-dispersion results.

### **3.5 Drainage Ditches**

GW-WETLAND has the ability to include drainage ditches along the ground surface of the computational domain. This option is included to simulate the effectiveness of ditches at draining adjacent zones of peat (Crowe *et al.*, 2000). In this example, the purpose of the drainage ditches was to lower the water table in the peat unit between two ditches such that the peat could be mined.

For this new option, the ground surface elevation for each column of nodes is entered as before. However, the user can then specify a surface ditch with vertical walls anywhere in the domain. This input overrides the previously entered ground surface data. In the previous version of the code (Shikaze and Crowe, 1999), if the user desired a drainage ditch, the elevation and location of the ditch would have to be entered using the ground-surface elevation input. In the case of a deep ditch, a portion of the finite-element mesh is actually removed from the domain to accurately represent the ditch.

Also, the ditch can have water levels that can change in time. This feature can be useful for simulating conditions that change annually or seasonally. The water-level history for the ditch is entered in a manner similar to that for surface water bodies. As the water level in the ditch rises, additional nodes along the walls of the ditch may be turned 'on' as constant head nodes. Similarly, as the water level in the ditch falls, some nodes along the walls of the ditch may be turned 'off' as constant head nodes (this is done automatically by the program).

The ditch must exist during the entire simulation; a ditch cannot be added in the middle of a transient simulation, and cannot be filled at any later time. A future addition to the program will be to allow a re-filling of the ditch at some time during the simulation. This could be used to simulate, for example, the restoration of water levels in adjacent peat units by refilling the drainage ditches.

### **3.6 Other Features**

#### **3.6.1 Pumping Wells**

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 nodes can be turned off and/or on at any time during the simulation.

#### **3.6.2 Spatially variable specific yield**

Specific yield can be entered to vary for each different hydrostratigraphic unit, in the same way that parameters such as hydraulic conductivity, porosity and specific storage coefficient can vary. This modification to the program offers more flexibility, and was specifically implemented for use in an application that involved the study of water table fluctuations in a peat material where hydraulic properties, including specific yield, were known vary spatially.



### **3.6.3 Fluid Mass Balance**

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.

### **3.6.4 Seepage Faces**

Seepage faces can form if the water table reaches ground surface. If a seepage node forms at the water table, the node becomes a constant head node, with the value of hydraulic head set equal to the elevation of the ground surface. If 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 to a free node.

### **3.6.5 Dynamic memory allocation**

GW-WETLAND automatically detects the size of most arrays that will be required for each specific simulation. As a result, the user does not need to worry about re-dimensioning most arrays prior to each new grid. Moreover, this dynamic memory allocation allows the program to be more efficient in terms of memory use, as arrays are only dimensioned as large as the need to be (in comparison with the older version, where some arrays may be much larger than required, thereby wasting significant memory, and thereby slowing the program and increasing simulation times). The user may still be required to re-dimension some arrays because in some cases, the required size of these arrays will not be known until the simulation is nearly complete. For these few arrays, we have specified values that should be large enough for most simulations (Appendix A4). However, in some cases, the user may be required to re-dimension the array and recompile the FORTRAN program.

## 4.0 OPERATION OF THE MODEL

This chapter describes the requirements and procedure for running the model. A detailed User's Guide, including step-by-step instructions for setting up the input data files, is shown in Appendix A.

### 4.1 Computing Requirements

The model has been programmed in FORTRAN, and compiled with Microsoft PowerStation, version 4.0. Some of the utilities in the code make use of functions that are specific to Microsoft PowerStation, and thus if the user wants to compile the code with another FORTRAN compiler, they should contact the authors 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. It is recommended that the user open a DOS-window from MS Windows (95, 98, NT, 2000, XP) to run the program. Output files are in ASCII format, and specific outputs are designed to be used by *GridBuilder* (McLaren, 1998) for visualization.

### 4.2 Dimensioning of the Arrays

Most arrays are dimensioned internally with each simulation; the user should not have to re-dimension arrays for larger simulations. However, some arrays are still included in the 'PELEE.INC' file, and may need re-dimensioning. Appendix A4 lists parameters that may require re-dimensioning. These parameters have been set to values that should be large enough to handle most simulations.

### 4.3 Input Operations

This section describes the data requirements for input to the model. The complete input guide, which includes a detailed line-by-line description of the input data files, is shown in Appendix A.

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 in milligrams. We recommend units of metres/days/milligrams, and the examples provided in this report use these units. Appendix A.5 summarizes input parameters that have units as well as acceptable combinations of units.

Although the input data file is in ASCII format, there are several features that make creating and editing this file easier. For example, any line within the data file that begins with a "!" is taken to be a comment line, and is ignored by the program. Also, on any given line, comments can be added after the

input parameters by typing a ";". The program will ignore anything that follows the semi-colon. This feature can be useful when the user wishes to add comments throughout the input file to aid in finding their way through the file.

The first information that the user must enter is a one-line title or problem identifier (**title**). This can be used to provide 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 **".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 (including 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 the finite-element Galerkin method

The finite element grid and hydraulic head distribution are automatically calculated at the beginning of the simulation - based on initial information supplied by the user - as well as at each subsequent time step. Information about the mesh and domain may be output to a file at each time step:

- option(1)** : node coordinates
- option(2)** : finite element incidences
- option(3)** : hydraulic conductivity and storativity of the element
- option(6)** : nodal hydraulic heads
- option(13)** : hydraulic gradients including direction
- option(18)** : particle and concentration distributions
- option(23)** : coordinates of particles and associated mass

It is worth noting that if these options are turned on, the resulting output files can be quite large.

The following information may be output to files that could be used for plotting (e.g. with *GridBuilder*):

- option(4)** : nodal 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
- option(16)** : velocity field (vertical and horizontal)
- option(19)** : concentrations and distribution of reference particles
- option(20)** : distribution of reference particles

These options allow the user to control aspects of a 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 is 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 horizontal and vertical grid spacings are used to create a mesh of cells, where each cell consists of 2 triangular finite elements.

The cross section may contain up to 9 different hydrostratigraphic units (**ngeol**), each with unique values of horizontal and vertical hydraulic conductivity (**khorz**, **kvert**), storativity (**stor**), specific yield (**specyd**), porosity (**por**), longitudinal dispersivity (**disperl**), and transverse dispersivity (**dispert**). These hydrostratigraphic units are used to represent the stratigraphy of the subsurface from the base of the cross section to the ground surface, whether or not the entire cross section will be saturated. The hydrostratigraphic units are defined as an array (**mapgeo**) by assigning a numbered code, from 1 to 9, which represents the 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, with 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 parameters assigned to each hydrostratigraphic unit (**khorz**, **kvert**, **stor**, **specyd**, **por**, **disperl**, **dispert**) are input with the hydrostratigraphic unit identifier (**l**) corresponding to the hydrostratigraphic unit in the cross section (**mapgeo**). Also, entered on this line are the codes to indicate whether or not a particle is allowed to move (**code**) within the specified hydrostratigraphic 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.

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 (**elvbases**) are input for each column of nodes. Because these values need not be constant, the cross section may have an irregular shape.

A single drainage ditch can be added by setting variable **ditch** to **.TRUE.**. If this is selected, the user must enter **dleft**, **dright**, and **dbottom** to define the left, right and bottom boundaries (in grid coordinates – usually metres). The user must specify whether include their own water-level history over time for the ditch (**watlev\_ditch = .TRUE.**) or have the program calculate transient water levels in the ditch (**watlev\_ditch = .FALSE.**) in response to groundwater fluxes and precipitation. If the user includes their own water-level history for the ditch, the water levels in the ditch are entered via a separate data file (**ditchheadfile**). This file should be set up in ASCII format, and the first line in this file should contain number of entries in the file (**n\_ditchhead\_chg**), followed by a list where each successive line contains a time (**t\_ditch\_head**) and a new water level (**ditch\_head**). If the user wants the program to calculate water levels in a ditch, (**watlev\_ditch = .FALSE.**), a maximum water level (**headmax\_ditch**) and minimum water levels (**headinit\_ditch**) for the ditch must be specified. For a rising water level, when this maximum level is reached, the water level in the ditch will not rise further; however, the water level can decline if conditions require such a drop.

A ditch is considered to be a surface-water body in the program. If the user sets up a simulation to have the program calculate the water levels in the ditch, the water level and volume of water in the ditch (and any other surface-water bodies) are calculated at each time step and printed to an output file with a \*.wbm file extension (see Appendix A.3).

A single initial value of hydraulic head for the entire saturated domain is required (**head1**); to help the solution converge. This initial value should be set equal to the average value of the hydraulic head along the water table. The initial values of hydraulic head along the water table, (i.e., the elevation of the water table) (**headwt**) are specified for each column of nodes. The user must also indicate at which nodes along the water table the hydraulic heads will be allowed to fluctuate 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 recharge/evapotranspiration boundaries (second-type) and constant head boundaries (first-type). Recharge and evapotranspiration are input separately to accurately simulate field conditions. Both recharge and evapotranspiration can vary over time and spatially across the cross section. Recharge is entered by first indicating the number of different recharge periods (**nrestp**) that will occur during a simulation, and the time at which each 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. For each recharge period, the spatial distribution along the cross section is indicated by specifying the number of zones of constant recharge across the cross section (**nrzones**), the starting and ending column number (**nxfrom**, **nxto**) bounding each recharge zone, and the value of recharge within each zone (**rval**). Recharge values of 0.0 should be entered 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 time periods in which different values of evapotranspiration are desired (**netper**), the times at which these evapotranspiration periods begin (**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 simulated, then the user simply enters a value of **netper** = 0, and no values for **t\_et**, **netzone**, **nxfrom\_et**, **nxto\_et**, and **etval** are required.

Constant head nodes are used to represent surface water bodies (lakes, rivers, wetlands), drains, or any location in the domain where the hydraulic head is known. Surface water bodies can be represented by selecting and inputting information using the 'Surface Water Body Data' option (see Appendix A.2.9). The user must specify whether they want to include their own water-level history for the surface water body (**watlev\_swb** = **.TRUE.**) or have the program calculate transient water levels in the water body (**watlev\_swb** = **.FALSE.**) in response to groundwater fluxes and precipitation.

If the user includes their own water-level history for the surface water body (**watlev\_swb** = **.TRUE.**), they must create a separate file containing the water-level history (**file\_wb**). 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 corresponding new water level. 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 which represent the location of the surface water bodies (**icolumns**). The columns chosen should include every nodal column where the maximum hydraulic 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 only a few columns. For example, if the surface of the water body specified in **file\_wb** range between 100.0 and 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 wants the program to calculate water levels in a surface water boundary, (**watlev\_swb = .FALSE.**), an initial water level for the surface water boundary (**init\_head\_swb**) and a maximum water level (**max\_head\_swb**) for the surface water boundary must be specified. For a rising water level, when this maximum level is reached, the level will not rise any further; however, the water level can decline if conditions require such a drop. As mentioned above, within the data set the user first specifies the number of distinct surface water bodies (**nwb**) within the cross section. The user must specify the number of columns of nodes associated with a surface water body (**nchnodes\_wb**), and the corresponding columns of nodes which represent the location of the surface water bodies (**icolumns**). These columns must represent columns which have a ground surface elevation that is less than or equal to the maximum water level (**max\_head\_swb**) for the surface water boundary.

If the user sets up a simulation to have the program calculate the water levels in the surface-water body, the water level and volume of water surface-water body are calculated at each time step and printed to an output file with a \*.wb~~n~~ file extension (see Appendix A.3). Volume-to-stage relationships are calculated automatically within GW-WETLAND based on the elevation of the bottom of the surface-water body as defined by the elevation of the surface of the cross sectional domain (**elvgrd**).

If the user does not want to enter surface water bodies or constant head nodes in this manner, then set **nwb = 0** and do not enter values for **watlev\_swb**, **nchnodes\_wb**, **file\_wb**, **max\_head\_swb**, **init\_head\_swb**, **icolumns**, as well as **nwbpart**, for particle transport.

Surface-water boundaries can be simulated as contaminant sources in one of two ways. First, these water boundaries can be set up as source zones for particle transport. In this case, **option(18)** must be **.TRUE.**. If the user wishes to simulate particle tracking from a surface-water boundary, **nwbpart** should be set to **.TRUE.**, and values should be entered for the number of particles for each source cell (**npart\_wb**) and the start and end times for the source zone (**ton\_part\_wb**, **toff\_part\_wb**). The second method available is solution to the advection-dispersion equation by the Galerkin finite element technique. If this method is desired, **option(18)** must be set to **.TRUE.**, and **nwb\_ad** should also be set to **.TRUE.**. Also, the user must specify values for the source concentration (**conc\_wb**) and the start and end times between which the source is applied to the water boundary (**ton\_ad\_wb**, **toff\_ad\_wb**). The remaining parameters for advective-dispersive flow are to be added later in the data file (see Appendix A, section A.2.13).

Regular constant head nodes (non-surface water bodies) can also be specified (e.g., drains). 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 **nthead**. 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  $-1.0 \times 10^{-35}$  to **chval** to designate that it will become a constant head node later during the simulation. The user may read the changing hydraulic head values for the constant head node 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  $-1.0 \times 10^{-35}$  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 node 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 head nodes in this manner, then simply set **nthead** = 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 has 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 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 occurs (**ton\_pw**), and the new pumping rates (**prate**). The pumping rate specified is that which is assigned to each node within the well screen box. If only one 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 at 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 hydraulic 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 10-15 iterations, but if convergence is for some reason not achieved, **nitmax** will prevent the model from iterating indefinitely.

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 is to be printed (**n\_out\_time**) and the desired times (**t\_out**) are input by the user. The user should specify at least one output time. 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 time step calculated by the program will adjust such that the user-specified output times will be reached. Secondly, the hydraulic head and the elevation of the water table at any location along the cross section can be printed, producing a time-series record of hydraulic heads and 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. Up to 9 observation points can be selected by entering the number of observation points (**n\_obs**) and the row (**nobs\_row**) and column (**nobs\_col**) that correspond to the desired observation point location. When the user specifies a row and column as an observation point, the hydraulic head AND elevation of the water table at that location is output at each time step. For example, if you specify Row 25 and column 20 as an observation well, the head and will be output for that exact node, and the water table elevation will be output for column 20.

The user can also simulate solute transport and basic reactions using one of two methods: (1) particle tracking and (2) direct solution of the advection-dispersion with Galerkin method (see version 1.0 of the Users Manual for more information (Shikaze and Crowe 1999).

When the particle tracking method is selected (**option(17) = .TRUE.**), only mass transport by advection and dispersion occurs. However, the mass of each particle can decrease over time according to a specified half-life (**half-life**) that represent mass loss due to degradation. The user defines the initial mass of each particle (**xmassi**), and then 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 (**picol, prcol**), and bottom and top rows (**pbrow, ptrow**), bounding the source zone. The user specifies the number of solute particles 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 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 listed as follows:

- 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 to 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
    - recharge/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) recharge and evapotranspiration rates (3) pumping rates.
  - 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.

## 5.0 ILLUSTRATED EXAMPLES

This chapter demonstrates how the model can be used to simulate typical scenarios. Also, this chapter aids in understanding and interpreting results from the model. Two examples are presented.

### 5.1 Example 1: Contaminant Transport from a River Flood

The first example involves transient groundwater flow in the vicinity of a short-term river flood. In this case, the ambient groundwater flow direction is towards the river; however, as the water level in the river rises during a flood, the groundwater flow direction may reverse such that water flows from the river into the groundwater system. If the flood water in the river is contaminated, the groundwater may also become contaminated. The input data file for this example is listed in Appendix B, and the output file is listed in Appendix C.

Boundary conditions have been set up such that the nodes along the river are specified head nodes that have values which vary in time. The remaining nodes along the water table are specified-flux nodes to represent recharge. The remaining boundary nodes have been set to no-flow boundaries (left and right sides, and bottom). Contaminants entering the groundwater regime from the river is examined by particle tracking whereby contaminant particles are added to elements adjacent to the river during the flooding event. The cross-section is 200 metres wide and the river exists on the right-hand side of the domain (Figure 5.1).

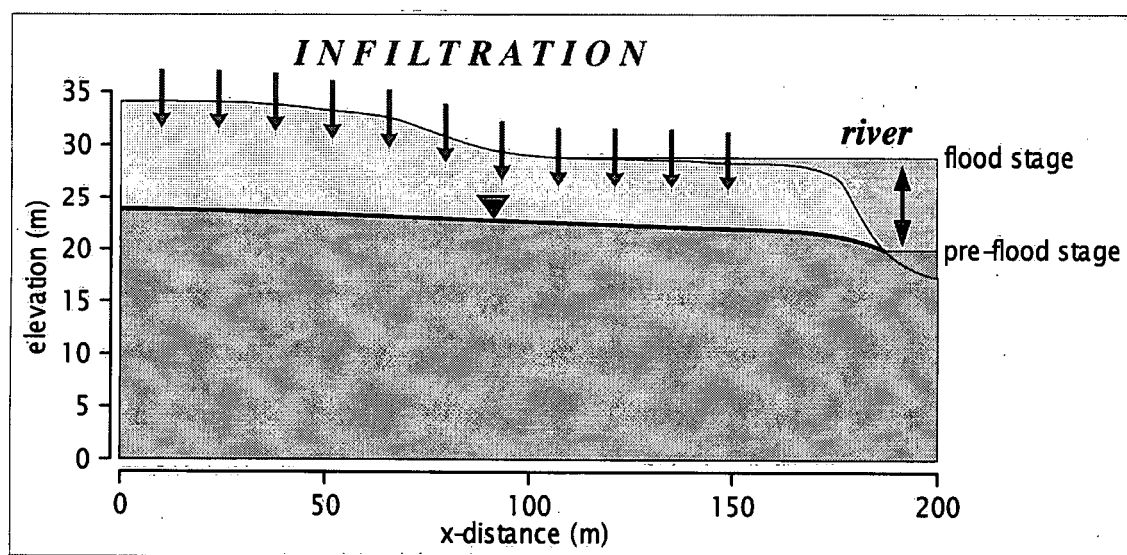


Figure 5.1. Schematic illustration of the cross section used to simulate groundwater-river interaction and contaminant transport into an aquifer during a river flood.

The water level in a river is steady for one year. During this time, groundwater flows from the aquifer into the river due to recharge that is applied to the aquifer. At the beginning of the second year, the water level in the river rises gradually for one-half day, then remains at the flood stage for 4 days, and then drops to its previous low level over a period of about one-half day. Once the water level in river returns to its pre-flood levels after 5 days, the water level remains at this level for the remainder of the simulation (730 days). The water level history for the river is shown in Figure 5.2.

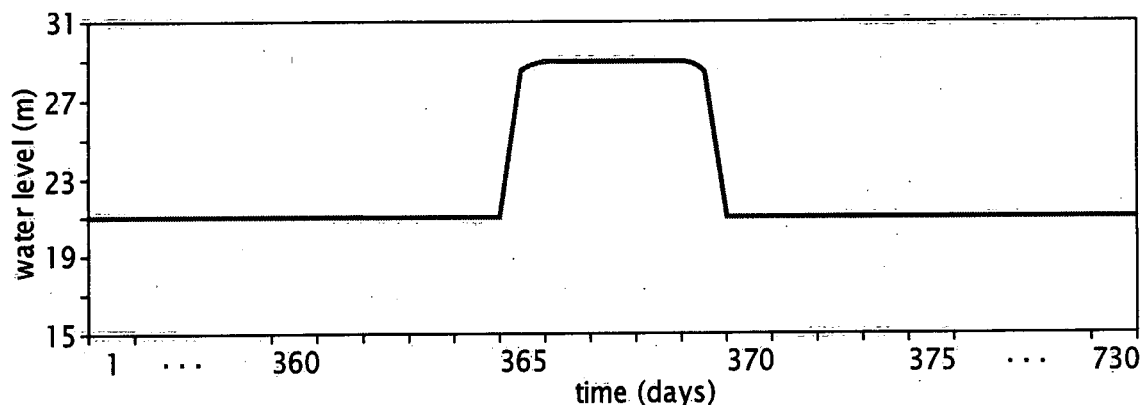


Figure 5.2. Water-level history in the river, used to simulate a flood between 365 and 370 simulation days.

The initial water level in the river is 21 metres and during the flood, the maximum water level is 29 metres, which represents the elevation of the flood plain, as shown in the schematic in Figure 5.1. The flood lasts five days at its peak water level before falling to its initial value of 21 metres. For solute transport, particle tracking is used, with particles added to cells adjacent to the river. Particles are only added during the time of the flood (approximately 365 to 370 days into the simulation).

Figure 5.3 shows results from this simulation at four different times. At 365 days, the flood has just started, and the river level is still low. Groundwater still flows from the aquifer into the river, and particles remain along the interface between the river and the aquifer. At a time of 369 days, the water level in the river is at its maximum, and at this time, the number of source cells associated with particle tracking has increased because the river has reached the flood plain. Consequently, the distribution of particles is much more extensive. Also, near the river, groundwater appears to be flowing from the river into the aquifer. At 370 days, the water level in the river has begun to subside and immediately adjacent to the river, groundwater is now flowing from the aquifer into the river. However, there is still a noticeable groundwater mound that results in the particles flowing away from the river. By 500 days, the river level has reached its original level and all remaining particles are migrating towards the river. Assuming conditions remain constant, all of these remaining particles will eventually exit into the river.

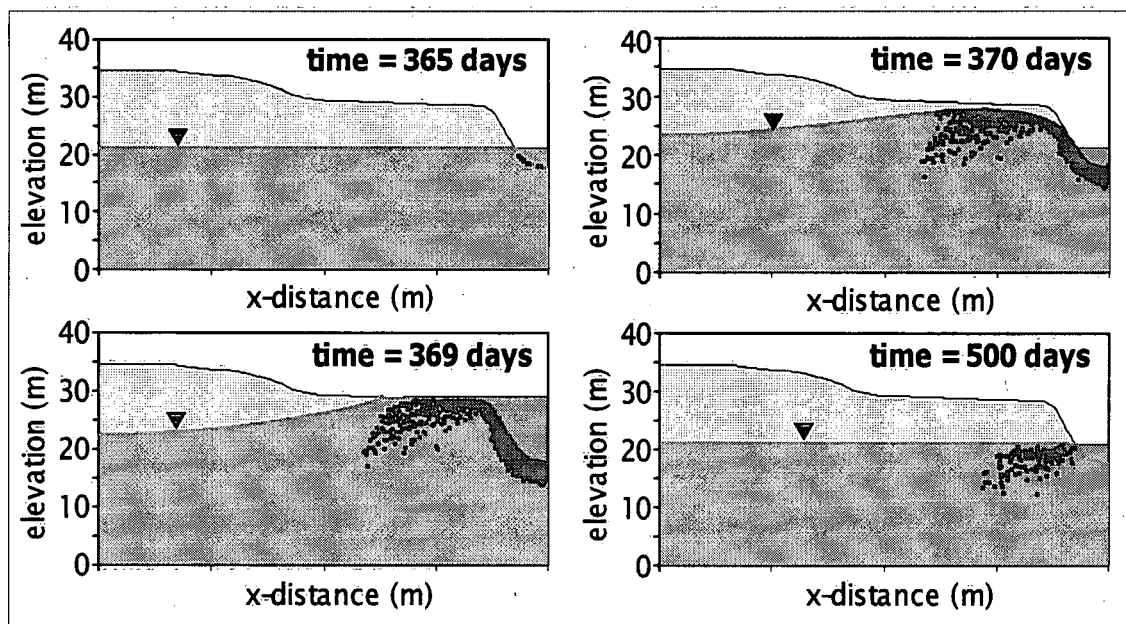


Figure 5.3. Results of the simulated flood illustrated changes to the elevation of the water table and contaminant transport into the aquifer during the flood and out of the aquifer after the flood.

## 5.2 Example 2: Water Levels in Surface-water Bodies that Change in Response to Fluxes

The second example illustrates the ability of the model to calculate water levels in surface water bodies on the basis of fluxes (groundwater, precipitation, evapotranspiration) to and from these bodies. The data file for this example is listed in Appendix B. A cross-section is simulated that includes two wetlands. A schematic of this example is shown in Figure 5.4. The length of the cross-section is 400 metres, and the thickness of the domain is variable, to a maximum of 9 metres. The first wetland (Wetland #1) is a small depression in the ground surface located near the centre of the domain, while the second wetland (Wetland #2) is a larger wetland located at the right of the domain.

At the beginning of the simulation, the program calculates the volume-to-water-level functions for both wetland boundaries as based on the geometry of the elevation of the bottom of each wetland (entered into the input data set as "ground surface elevation"). These functions are plotted in Figure 5.5. The maximum water level for Wetland #1 has been set to 8.15 m, and this value is not reached during the two-year simulation. For Wetland #2, the maximum water level is set to 8 m, and this value is reached during the second year of the simulation, in the spring and the fall. The initial water level in both wetlands and the initial water table are all set to 7.5 metres.

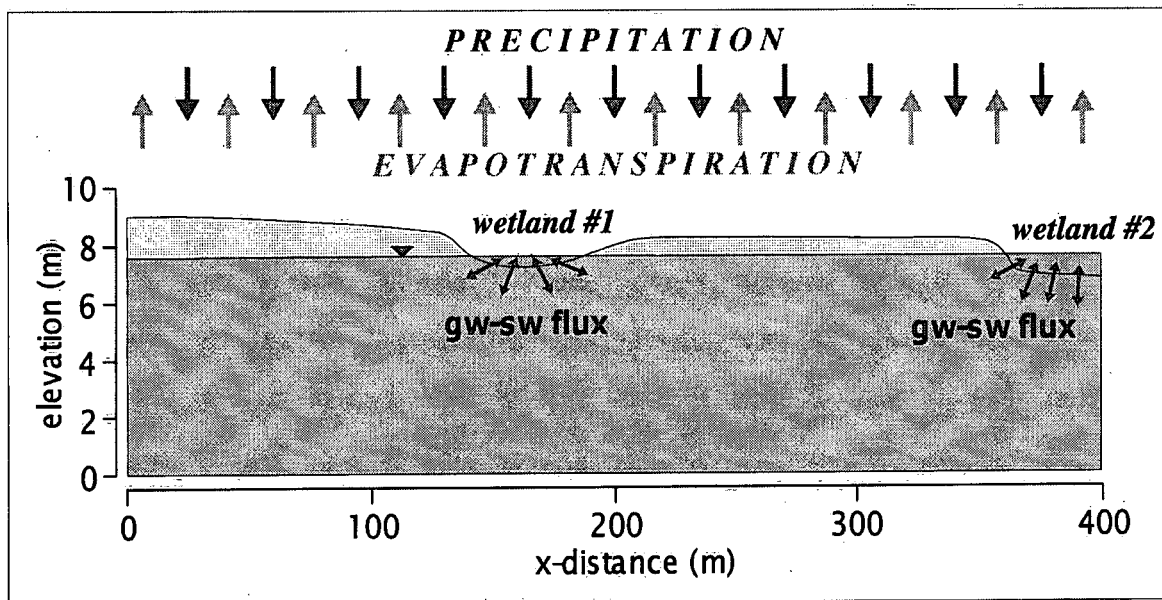


Figure 5.4. Schematic diagram of the cross section used to groundwater-surface fluxes and changes to the size of the wetland due to these fluxes.

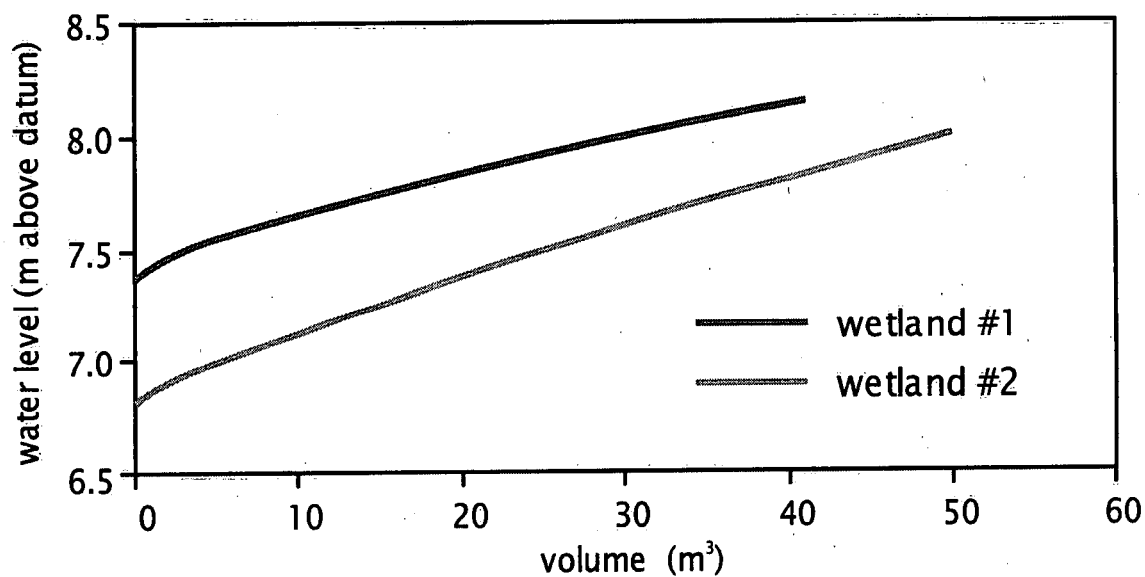


Figure 5.5. Water-level-to-volume functions for the two wetlands.

The simulation was run for two years (730 days), and recharge and evapotranspiration were varied over the duration of the two years. Recharge (precipitation) is highest in the spring months, followed by a decrease during the summer months, a slight increase in fall, and a drop again in winter, as the snow pack is assumed to prevent recharge. Evapotranspiration is zero through the first few

months of each year, and slowly increases to a peak in summer, when trees/plants have fully bloomed and are active. This is followed by a decrease towards the fall. Figure 5.6 shows plots of the recharge and recharge - evapotranspiration for the duration of the simulation. Values of recharge are applied directly to the wetlands, while recharge - evapotranspiration is applied to the portion of the domain that does not include any wetland.

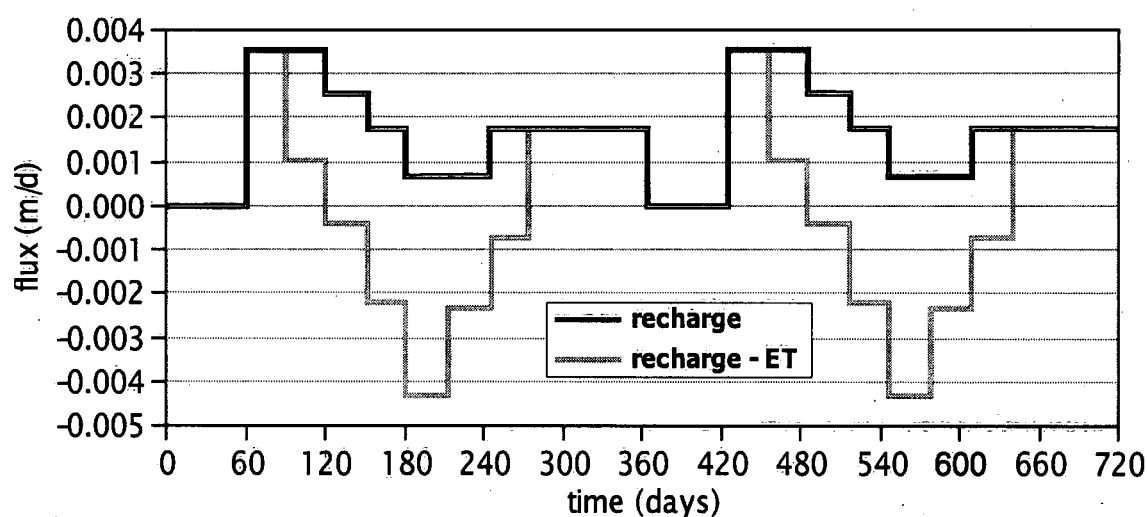


Figure 5.6. Values of recharge and recharge-evapotranspiration used in the simulation.

Figure 5.7 shows the upper portion of the domain, and the water table location and the water levels in the two wetlands at four different times during the simulation. In spring, the water table begins to rise due to the higher rates of recharge. As time proceeds, groundwater eventually flows towards the wetland, and the water levels in the wetlands increase. Figure 5.8 shows the water-level history for both wetlands, and the increase in water levels in the wetlands can be traced directly to the transient nature of groundwater discharge to the wetland, surface water flow into the aquifer from the wetland, and precipitation and evapotranspiration at the wetland.



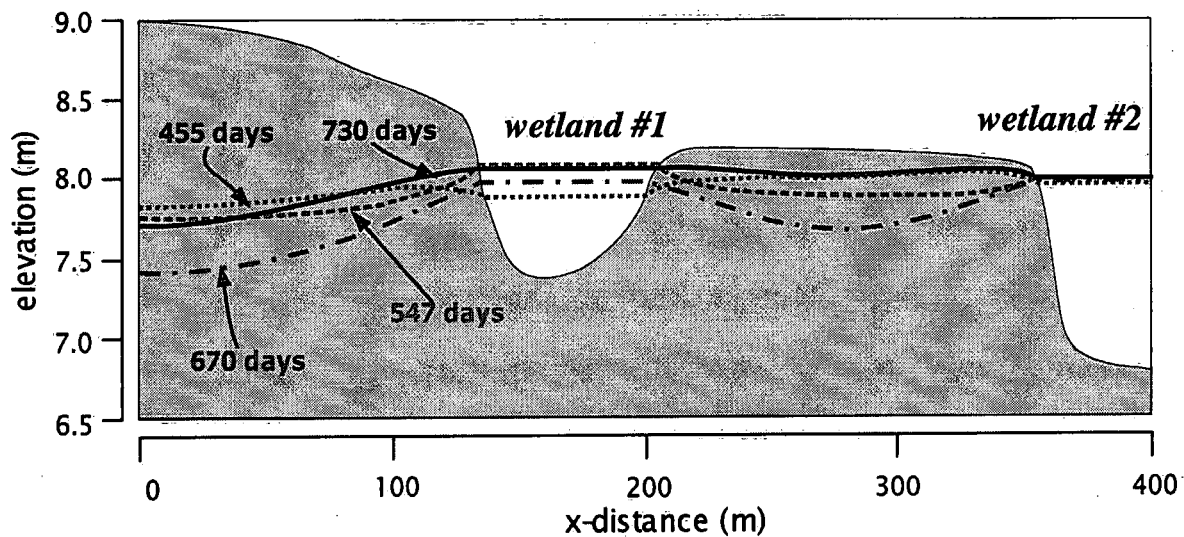


Figure 5.7. Simulation results showing changes in the position of the water table and water levels of the two wetland at 455, 547, 670 and 730 days after the start of the simulation.

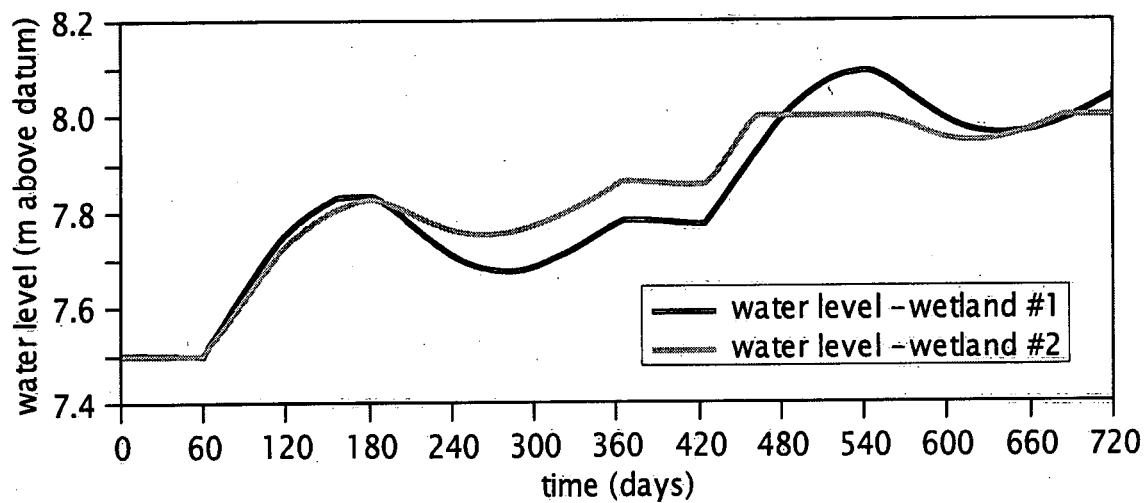


Figure 5.8. Simulated water levels in the two wetlands over the two-year simulation.

## 6.0 REFERENCES

- Ahlstrom, S.W., H.P. Foote, R.C. Arnett, C.R. Cole and R.J. Serne. 1977. Multicomponent mass transport model: Theory and numerical implementation (discrete-parcel-random-walk version); Batelle P.N.L. Report 2127
- Crowe, A.S., S.G. Shikaze and F.W. Schwartz. 1999. A grid-generating algorithm for simulating a fluctuating water table boundary in heterogeneous unconfined aquifers. *Advances in Water Resources*, 22(7):43-51.
- Crowe, A.S., S.G. Shikaze, J.E. Milne and M.G. Skafel. 2005. Groundwater as a source of *E. coli* along the S.E. shoreline of Lake Huron, in Program and Abstracts of the 40<sup>th</sup> Central Canadian Symposium on Water Quality Research. Burlington, Ontario, February 14-15, 2005, page 42.
- Crowe, A.S., S.G. Shikaze and C.J. Ptacek. 2004. Numerical modelling of groundwater flow and contaminant transport to the Point Pelee marsh, Ontario, Canada. *Hydrological Processes*, 18:293-314.
- Crowe, A.S., S.G. Shikaze and J.E. Smith. 2000. Hydrogeological studies in support of the restoration of the Wainfleet Bog: Numerical Modelling. Report submitted to the Niagara Peninsula Conservation Authority, National Water Research Institute Contribution 00-63, 86 pp.
- Gilver, D.J., R. Andrews, J.H. Tellham, J.W. Lloyd and D.N. Lerner. 1993. Quantification of the water balance and hydrogeological processes in the vicinity of a small groundwater-fed wetland, East Anglia, U.K. *Journal of Hydrology*, 144:311-334.
- Mansell, R.S., S.A. Bloom and G. Sun. 2000. A model for wetland hydrology: description and validation. *Soil Science*, 165:384-397.
- McLaren, R.G. 1998. GRID BUILDER 5.21 User's Guide. A generator for 2-D triangular finite element meshes and mesh properties. Institute for Groundwater Research, University of Waterloo, Waterloo, Ontario.
- Neuman, S.P. and P.A. Witherspoon. 1971. Analysis of nonsteady flow with a free surface using the finite element method. *Water Resources Research*, 7(3):611-623.
- Pinder, G.F. and W.G. Gray. 1977. Finite element simulation in surface and subsurface hydrology. Academic Press, Inc., 295 pp.
- Prickett, T.A., T.G. Naymik and C.G. Lonquist. 1981. A random-walk solute transport model for selected groundwater quality evaluations. *Illinois State Water Survey Bulletin* 65, Champaign, Illinois, 103pp.
- Reeve, A.S., D.I. Siegel and P.H. Glaser. 2000. Simulating vertical flow in large peatlands. *Journal of Hydrology*, 227:207-217.
- Restrepo, J.I., A.M. Montoya and J. Obeyesekere. 1998. A wetland simulation module for the MODFLOW ground water model. *Ground Water*, 36:764-770.
- Schwartz, F.W. 1978. Application of probabilistic-deterministic modeling to problems of mass transport in groundwater systems. *Third International Hydrology Symposium*, Ft. Collins, pp. 281-296.

- Schwartz, F.W. and A.S. Crowe. 1980. A deterministic-probabilistic model for contaminant transport: User's manual; Report by CGS Inc., prepared for the U.S. Nuclear Regulatory Commission, report NUREG/CR-1609 CGS/NR85U060, 158 pp.
- Schwartz F.W. and A.S. Crowe. 1985. Simulation of changes in ground-water levels associated with strip mining. *Geological Society of America Bulletin*, 96:253-262.
- Schwartz F.W. and A.S. Crowe. 1987. Model study of some factors influencing the resaturation of spoil following mining and reclamation. *Journal of Hydrology*, 92:121-147.
- Shikaze, S.G., and A.S. Crowe. 1998. Impact of flooding and flood mitigation on groundwater quality. In *Proceedings of the Conference: Groundwater in a Watershed Context*, Burlington, Ontario, pp. 389-396.
- Shikaze, S.G. and A.S. Crowe. 1999. 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. National Water Research Institute Contribution 99-204, 65 pp.
- Shikaze, S.G. and A.S. Crowe. 2000. UPDATE for User's Guide for GW-WETLAND (Version 1.1): 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. National Water Research Institute Contribution 00-65, 65 pp.
- Shikaze, S.G. and A.S. Crowe. 2001. Input Guide and Revisions for GW-WETLAND (Version 1.2): A Computer Program to Simulate Groundwater-Wetland Interactions. National Water Research Institute Technical Note AEI-TN-01-001, 86 pp.
- Siegel, D.I. 1983. Ground water and the evolution of patterned mires: Glacial Lake Agassiz peatlands, northern Minnesota. *Journal of Ecology*, 71:913-921.
- Winter, T.C. 1999. Relation of streams, lakes and wetlands to groundwater flow systems. *Hydrogeology Journal*, 7:28-45.
- Woo, M-K and T.C. Winter. 1993. The role of permafrost and seasonal frost in the hydrology of northern wetlands in North America. *Journal of Hydrology*, 141: 5-31.

## APPENDIX A - INPUT GUIDE

This appendix is setup as follows: Section A.1 describes some general points with regards to input instructions. Section A.2 describes the specific input requirements for the data file that are used to run the code. Section A.3 lists files that are created by a successful simulation and Section A.4 lists the arrays and dimensions. Section A.5 lists some tips in ensuring that a simulation is completed successfully.

### A.1 General Notes

- The executable file should be run through a DOS window.
- Once the code is run, the user will be asked to input the **prefix**. The prefix represents the filename without the three-letter extension. The prefix must have a **'.dat'** extension and be 7 characters or less in length. If the data file is not found on the current directory, an error message will appear and the program will stop.
- Any line in the **prefix.dat** file that begins with a **'!'** is a comment line and will be ignored by the program. This option is useful for placing statements within the data file to aid the user in finding exactly where specific parameters are located.
- All data is free-format. In other words, the user does not have to be concerned with aligning the input parameters in specific columns. A space or a comma must separate variables that occur on the same line.

## A.2 Input Guide

### A.2.1 Program Title

1. **Title** - Any character string (up to 80 characters in length).

**NOTE:** If the user needs more descriptive title lines, use the '!' option discussed above; however, you must have one legitimate title line that does not begin with '!'

### A.2.2 Program Options

**NOTE:** For program options listed below, all output is written in ASCII format to a file called **prefix.out** unless otherwise specified. Also, setting options (1), (2), (3), (9), (13), (18), (23), to **True** can result in a very large **prefix.out** file

2. **option(1)** - LOGICAL

**True:** Print nodal coordinates

**False:** Do not print nodal coordinates.

3. **option(2)** - LOGICAL

**True:** Print elemental incidences

**False:** Do not print elemental incidences.

4. **option(3)** - LOGICAL

**True:** Print elemental K, S

**False:** Do not print elemental K, S.

5. **option(4)** - LOGICAL

**True:** Print node/element data to **prefix.ele** file

**False:** Do not print node/element data to **prefix.ele** file.

6. **option(5)** - LOGICAL

**True:** Calculate hydraulic heads

**False:** Do not calculate hydraulic heads.

7. **option(6)** - LOGICAL

**True:** Print heads to output file

**False:** Do not print heads to output file.

**NOTE:** **option(6)** must be set to **True** if *GridBuilder* files are desired as output

8. **option(7)** - LOGICAL

**True:** Print heads to **prefix.hds** file

**False:** Do not print heads to **prefix.hds** file.

9. **option(8)** - LOGICAL

**True:** Print only final heads to **prefix.hds** file

**False:** Do not print only final heads to **prefix.hds** file.

10. **option(9)** - LOGICAL

**True:** Print water table convergence data

False: Do not print water table convergence data.

11. **option(10)** – LOGICAL

True: Read old heads (from file **prefix.hin**) to continue a run

False: Do not read old heads to continue a run.

12. **option(11)** – LOGICAL

True: Output last heads for a later run (to file **prefix.hou**)

False: Do not output last heads for a later run.

13. **option(12)** – LOGICAL

True: Generate gradients/directions

False: Do not generate gradients/directions.

14. **option(13)** – LOGICAL

True: Print gradients to output file

False: Do not print gradients to output file.

15. **option(14)** – LOGICAL

True: Output gradients to **prefix.hyd** file

False: Do not output gradients to **prefix.hyd** file.

16. **option(15)** – LOGICAL

True: Generate velocity field (if True, **option(12)** must be also be True)

False: Do not generate velocity field.

17. **option(16)** – LOGICAL

True: Print velocity field to **prefix.vel** file

False: Do not print velocity field to **prefix.vel** file.

18. **option(17)** – LOGICAL

True: Run solute transport using the deterministic-probabilistic routine

False: Do not run the deterministic-probabilistic routine.

19. **option(18)** – LOGICAL

True: Print concentrations/particle distribution

False: Do not print concentrations/particle distribution.

20. **option(19)** – LOGICAL

True: Output concentrations/particle distribution to **prefix.con** file

False: Do not output concentrations/particle distributions.

21. **option(20)** – LOGICAL

True: Output particle distribution (**prefix.prt** file)

False: Do not output particle distribution.

22. **option(21)** – LOGICAL

True: Output last particle distribution for later input (to file **prefix.hou**)

False: Do not output last particle distribution for later input.

23. **option(22)** – LOGICAL

True: input particles from an old run (**prefix.hin** file)

False: Do not input particles from an old run (see section A.5 for details).

24. **option(23)** – LOGICAL

**True:** Print mass and particle coordinates

**False:** Do not print mass and particle coordinates.

25. **option(24)** – LOGICAL

**True:** Run solute transport using the Galerkin finite element method

**False:** Do not run solute transport using the Galerkin method.

### A.2.3 Observation Points

26. **n\_obs** – INTEGER – Number of observation points at which water table elevations will be output at every time step.

**NOTE:** If **n\_obs** = 0, skip line 27, and continue on to section A.2.4

27. **nobs\_col(n\_obs), nobs\_row(n\_obs)** – 2 INTEGERS – Column and row corresponding to the desired location for output of water table elevations and nodal hydraulic heads. The hydraulic head will be written for this exact node, and the water table elevation will be written for the column **nobs\_col**. Both values will be written at each time step.

**NOTE:** Repeat this line **n\_obs** times. If **n\_obs** = 0, skip this line

### A.2.4 Grid/Boundary Condition/Time Stepping Input

28. **nrow, ncol** - 2 INTEGERS - Number of rows and columns of nodes.

**NOTE:** **ncol** must be exact and **nrow** must be large enough to accommodate the maximum elevation of the ground surface

29. **ngeol, nitmax** - 2 INTEGERS

**ngeol:** Number of geological units

**nitmax:** Maximum number of iterations per time step.

30. **var\_x** – LOGICAL

**True:** variable grid spacing in x-direction

**False:** constant grid spacing in x-direction.

31. **xe(ncol-1)**: REAL – Grid spacing in the x-direction (horizontal).

**NOTE:** If **var\_x** = **True**, enter (**ncol-1**) values of **xe** (for variable grid spacing in x); if **var\_x** = **False**, enter only one value (constant grid spacing in x)

32. **delz** - REAL - Grid spacing in the z-direction (vertical).

33. **ftime** - REAL - Final simulation time (the simulation will stop when it reaches this time regardless of whether or not steady state has been reached).

34. **n\_out\_time** - INTEGER - Number of desired times at which to create output.

35. **t\_out(n\_out\_time)** - REAL - Desired output times.

**NOTE:** The user must specify at least one output time.

36. **delt, dtmax**: 2 REALS

**delt** : Initial time step size

**dtmax** : Maximum time step size.

37. **headi**: REAL - Initial head value assigned to the entire computational domain.

38. **tolrnc, deltin** - 2 REALS

**tolrnc**: Convergence tolerance (for both steady-state and iteration convergence)

**deltin**: Factor by which to increase the size of each time step.

### A.2.5 Column Data

39. **elvgrd(ncol)** - REAL - Elevation of the ground surface. Enter one value for each nodal column.

40. **wtmove(ncol)** - LOGICAL

**True**: Water table is allowed to move;

**False**: Water table is not allowed to move. Enter one value for each nodal column.

41. **headwt(ncol)** - REAL - Initial head values along the water table. Enter one value for each nodal column. If there are any constant head nodes along the water table, the value of **headwt** at these nodes should be set to the initial constant-head value.

42. **elvbas(ncol)** - REAL - Elevation of the base of the domain. Enter one value for each nodal column.

### A.2.6 Drainage Ditches

43. **ditch** - LOGICAL

**True**: Include a drainage ditch;

**False**: Do not include a drainage ditch.

**NOTE:** If **True**, include lines 44 to 46 below; if **False**, skip lines 45 to 46 and proceed directly to line 47; currently, only one single ditch can be added, and it must exist throughout the entire duration of the simulation

44. **dleft, dright, dbottom** - 3 REALS

**dleft**: Left boundary of the desired ditch (in grid coordinates, in metres).

**dright**: Right boundary of the desired ditch (in grid coordinates, in metres).

**dbottom**: Bottom boundary of the desired ditch (in grid coordinates, in metres).

**NOTE:** Data input from this section will override the information for the elevation of the ground surface that has been previously entered

45. **watlev\_ditch** - LOGICAL

**True**: Read ditch water levels from an external data file. Proceed to line 46, below.



**False:** Read maximum (**headmax\_ditch**) and initial (**headinit\_ditch**) ditch water levels. The program will calculate transient water levels in the ditch at each time step. If selected, skip line 46.

46. **ditchheadfile** – CHARACTER – File name for the data file that contains the water-level history for the ditch. The format of this data file is as follows: on the first line of the data file, the number of water-level changes within the ditch is required (**n\_ditchhead\_chg**). Subsequent lines in this other file should contain, first, the time at which the new water level is to occur (**t\_ditch\_head(n\_ditchhead\_chg)**), and second, the new water level for the current time (**ditch\_head(n\_ditchhead\_chg)**).

### A.2.7 Recharge/Evapotranspiration Data

47. **nrestp** – INTEGER - Number of different recharge periods. If **nrestp** = 0, skip to line 50. For each recharge period, the user can specify different zones, if the recharge varies spatially.
48. **t\_rech**, **nrzones** - REAL, INTEGER (Repeat this line for each value of **nrestp**).  
**t\_rech**: Time at which to apply the new recharge values  
**nrzones**: Number of spatial zones for current recharge period.
49. **nxfrom**, **nxto**, **rval** – 2 INTEGERS, REAL (Repeat this line for each value of **nrzones**).  
**nxfrom**: Starting nodal column for current recharge zone  
**nxto**: Ending nodal column for current recharge zone  
**rval**: Recharge value for current recharge zone.
50. **netper** – INTEGER - Number of different evapotranspiration periods. If **netper** = 0, skip to line 53. For each evapotranspiration period, the user can specify different zones, if the evapotranspiration varies spatially.
51. **t\_et**, **netzone** - REAL, INTEGER (Repeat this for each value of **netper**)  
**t\_et**: Time at which to apply the new evapotranspiration values  
**netzone**: Number of spatial zones for current evapotranspiration period.
52. **nxfrom\_et**, **nxto\_et**, **etval** – 2 INTEGERS, REAL (Repeat this for each value of **netper**)  
**nxfrom\_et**: Starting nodal column for current evapotranspiration zone  
**nxto\_et**: Ending nodal column for current evapotranspiration zone  
**etval**: Recharge value for current evapotranspiration zone.

### A.2.8 Constant Head Data

53. **nthead** – INTEGER - Number of constant head nodes. If **nthead** = 0, skip to line 58.

<b>NOTE:</b> For each constant head node, repeat lines 54 to 57
---

54. **ich(nthead)**, **jch(nthead)**, **chval**, **nchg\_head(nthead)** - 2 INTEGERS, REAL, INTEGER  
**ich**: Column number of current constant head node  
**jch**: Row number of current constant head node  
**chval**: Original value assigned to the current constant head node

**nchg\_head**: Number of times to change the head value at the current constant head node (if **nchg\_head** = 0, **chval** applies throughout the entire simulation. Skip lines 55-57 for current constant head node).

**NOTE**: If the user wants to have a constant head node turned on at a later time, they must specify a value for **chval** here equal to  $-1.0 \times 10^{-35}$

**55. chead\_file** - LOGICAL

**True**: if changing constant head data is read from another file (go to line 56, then skip 57.

**NOTE**: if another file is used, data should be in the same format as line 57)

**False**: if changing constant head data is read from this data file (skip line 56).

**56. chead\_file\_name** - CHARACTER - filename which contains changing constant head data (if **thead\_file** = **True**).

**57: t\_new(nchg\_head), chg\_h\_val(nchg\_head)** - 2 REALS

**t\_end**: Time at which to change the constant head value assigned to the current node

**chg\_h\_val**: New constant head value assigned to the current node.

**NOTE**: If a node is initially not a constant head node, the user must still flag it as a constant head node and assign it a value equal to  $-1.0 \times 10^{-35}$ . The user can turn constant head nodes off at any time by assigning **chg\_h\_val** to be  $-1.0 \times 10^{-35}$

### A.2.9 Surface Water Body Data

**58. nwb** - INTEGER - Number of surface water bodies. **NOTE**: lines 59-67 must be repeated from 1 to **nwb** times. If **nwb** = 0, skip lines 69 to 70, and proceed directly to section A.2.10.1

**59. watlev\_swb(nwb)** - LOGICAL

**True**: user must specify the water-level history for current surface water body.

**False**: at each time step, a new water level in the surface water body is calculated

**60. nchnodes\_wb(nwb)** - INTEGER - Number of columns associated with current surface water body. These nodes should be chosen where the maximum head in the water-level history for the current surface water body exceeds the elevation of the ground surface for the column.

**NOTE**: If **watlev\_swb** = **True**, the user must include line 61, and skip lines 62 and 63. If **watlev\_swb** = **False**, line 61 is skipped.

**61. file\_wb(nwb)** - CHARACTER - If the user specifies that the program calculate water levels in the surface water body (**watlev\_swb** = **True**), this filename must be entered. It represents the filename that contains the water level history of the current surface water body.

**NOTE**: the filename must be in ASCII format. The first row in the file should contain the number of entries in the file. This row must be followed by a list of time and water level, with each time and a corresponding water level on a single line.

62. **max\_head\_swb(nwb)** – REAL – If the user specifies that the program calculate water levels in the surface water body (**watlev\_swb = False**), this value must be entered. It represents the maximum water level, above which, the level in the surface water body will not rise.

63. **init\_head\_swb(nwb)** – REAL – If the user specifies that the program calculate water levels in the surface water body (**watlev\_swb = False**), this value must be entered. It represents the initial water level in the surface water body.

64. **icolumns(nwb, nchnodes\_wb)** – INTEGERS - Columns (of nodes) at which to apply data from current surface water body.

65. **nwbpart** – LOGICAL

**True:** Include particle transport with current surface-water body as a source boundary.

**NOTE:** make sure **option(18)** is set to **True**

**False:** Do not include particle transport.

**NOTE:** If **nwbpart = True**, include steps 63 and 64 below. **nwbpart = False**, skip lines 66 and 67, and proceed to line 68

66. **npart\_wb(nwbpart)** – INTEGER - Number of particles added to each cell that is associated with a surface-water boundary.

67. **ton\_part\_wb(nwbpart), toff\_part\_wb(nwbpart)** – 2 REALS

**ton\_part\_wb:** Time at which to start adding particles to the surface-water boundary.

**toff\_part\_wb:** Time at which to stop adding particles to the surface water boundary.

68. **nwb\_ad** – LOGICAL

**True:** Include advection-dispersion transport with current surface-water body as a source boundary.

**NOTE:** make sure **Option(24)** is set to **True**

**False:** Do not include advection-dispersion.

**NOTE:** If **nwb\_ad = True**, include steps 69 and 70 below. **nwb\_ad = False**, skip lines 69 and 70, and proceed to line 71

69. **conc\_wb(nwb\_ad)** – INTEGER - source concentration that is associated with a surface-water boundary.

70. **ton\_ad\_wb(nwb\_ad), toff\_ad\_wb(nwb\_ad)** – 2 REALS

**ton\_ad\_wb:** Time at which to start advection-dispersion with the surface-water boundary as a source boundary

**toff\_ad\_wb:** Time at which to stop advection-dispersion with the surface water boundary as a source boundary.

#### A.2.10 Pumping Well Data

71. **nwells** - INTEGER - Number of pumping wells. If **nwells > 0**, loop over lines 72 to 76 for each 'nwell', otherwise if **nwells = 0**, skip to line 74.

72. **x1, x2** - 2 REALS - X-range within which all nodes will be flagged as well nodes with the following well parameters:

73. **z1, z2** - 2 REALS - Z-range as above. **NOTE:** **x1, x2, z1, z2** define a box inside which all nodes will be assigned the following well parameters. Each node within the box will be assigned the same pumping rate, and the flux is not 'distributed'.

74. **n\_pump\_per(nwells)** - INTEGER - Number of pumping periods for current well

**NOTE:** If a constant pumping rate is desired, **n\_pump\_per** must be set to 1. Thus, this variable should always be > 0 if wells are considered.

75. **ton\_pw(nwells, n\_change\_pw(nwells))** - REAL - Time at which to change to new pumping rate for current pumping period.

**NOTE:** Repeat Lines 75 and 76 up to the value in Line 74)

76. **prate(nwells, n\_change\_pw(nwells))** - REAL - New pumping rate for current pumping period.

#### A.2.11 Stratigraphy Data

77. **l(ngeol)** - INTEGER - Zone number.

78. **khorz, kvert** - 2 REALS

**khorz:** Horizontal hydraulic conductivities for the current zone l.

**kvert:** Vertical hydraulic conductivities for the current zone l.

79. **stor, por, specyd** - 3 REALS

**stor:** Storage coefficient for the current zone l.

**por:** Porosity for the current zone l.

**specyd:** Specific yield for the current zone l.

80. **disperl, dispert** - 2 REALS

**disperl:** Longitudinal dispersivity of the current zone l.

**dispert:** Transverse dispersivity for the current zone l.

**NOTE:** If transport is not done, these values are still read in, but are not used

81. **code** - LOGICAL

**True:** Particles are allowed to move within this zone

**False:** Particles are not allowed to move within this zone (via particle tracking).

**NOTE:** Repeat lines 77 to 81 from (1 to ngeol)

82. **mapgeo(ncol, nrow)** - INTEGER - Map of zone numbers. Repeat, beginning at the top of the grid, for each row (where each row contains the code for each column). This map should be entered as it appears - in other words, the lower-left corner of the cross section should be entered in the lower-left corner of this map.

### A.2.12 Particle Tracking Data

**NOTE:** This section is only used if **option(17)** is **True**

- 83. **xmassi** - REAL - Initial mass added per particle.
- 84. **half-life** - REAL - Half-life for particle decay.
- 85. **npart\_zones** - INTEGER - Number of particle source zones.

**NOTE:** Lines 85-88 are repeated from (1 to **npart\_zones**)

- 86. **npper** - INTEGER - Number of particles to add.
- 87. **t\_start\_part, t\_stop\_part** - 2 REALs - Time at which to add particle(s) for current particle source zone; time at which to stop adding particle(s) for current particle zone.
- 88. **plcol, prcol** - 2 INTEGERS - Left and Right columns of nodes defining the end columns for the current particle source zone.
- 89. **pbrow, ptrow** - 2 INTEGERS - Bottom and Top rows of nodes defining the end rows for the current particle source zone.

### A.2.13 Advection/Dispersion Data

**NOTE:** This section is only used if **option(24)** is **True**

- 90. **retfact, decay** - 2 REALS  
**retfact:** Retardation factor.  
**decay:** Decay coefficient.
- 91. **dstar, tort** - 2 REALS  
**dstar:** Diffusion coefficient.  
**tort:** Tortuosity.
- 92. **epsi** - REAL - Time-weighting factor (=0.5 for Crank-Nicolson time-weighting).
- 93. **consist** - LOGICAL - **True:** for consistent formulation; **False:** for lumped formulation.
- 94. **n\_conc\_zones** - INTEGER - Number of specified concentration zones.

**NOTE:** Lines 94-96 are repeated for (1 to **n\_conc\_zones**)

- 95. **x1conc, x2conc** - 2 REALS - X-coordinates of the start and end values that define a box inside which all nodes are constant concentration.
- 96. **z1conc, z2conc** - 2 REALS - Z-coordinates of the start and end values that define a box inside which all nodes are constant concentration.
- 97. **cinit** - REAL - Concentration value assigned to current zone.

### A.3 Output Files

The current version of the code produces the following output files:

- prefix.exe** - This file contains ASCII output of the elemental incidences. These are written to this file if OPTION(4) is set to TRUE.
- prefix.hyd** - This file contains ASCII output of the hydraulic gradients for each element. These are written to this file if OPTION(14) is set to TRUE.
- prefixn.imp** - Files that can be 'imported' into *GridBuilder* (NOTE: the '**n**' that is appended on the prefix represents a counter for each output time. For example, if three output times are specified, three files will be created: **prefix1.imp**, **prefix2.imp**, **prefix3.imp**)
- prefixn.n01** - Files containing hydraulic head values at each output time (**n**). As before, one file is created for each specified output time. These files are in ASCII format and can be read in as nodal files in *GridBuilder*.
- prefixn.e01** - Files containing elemental hydraulic conductivity values at each output time, (**n**). As before, one file is created for each specified output time. The files are in ASCII format and can be read in as element files in *GridBuilder*. (NOTE: One file is created at each specified output time because the finite element grid may be changing in time.) This output file is created at each output time and is not controlled by any of the OPTIONS.
- prefixn.e02** - Files containing particle concentrations at each output time, (**n**). Because particle concentrations are formulated on a cell basis, cell concentrations are converted to an elemental basis (where each cell is comprised of two triangular finite elements). As before, one file is created for each specified output time. The files are in ASCII format and can be read in as element files in *GridBuilder*. (NOTE: One file is created at each specified output time because the finite element grid may be changing in time.)
- prefixn.v01** - Files containing elemental velocity values ( $V_x$ ,  $V_z$ ) at each output time. As before, one file is created for each specified output time. The files are in ASCII format and can be read in as velocity files (in 'Edit Elements') in *GridBuilder*.
- prefixn.f01** - Files containing cell concentrations from the particle tracking routine. This file can be read by *GridBuilder* using the 'Field Data' option.
- prefixn.f02** - Files containing particle locations (coordinates) from the particle tracking routine, and associated particle mass. This file can be read by *GridBuilder* using the 'Field Data' option.
- prefixn.c01** - Files containing nodal concentrations from the advection/dispersion equation. These are ASCII files and can be read by *GridBuilder* as nodal files.
- prefix.ob1** - **prefix.ob9** - Files containing water-table heads versus time for specified columns. Up to 9 columns can be specified.
- prefix.mbl** - Fluid mass balance information.
- prefix.swb** - Fluxes at surface water bodies. For example, if two surface water bodies exist, this file will contain a list of columns containing (1) time, (2) fluid flux at the first surface water body, (3) fluid flux at the second water body.

- prefix.tim** - Computation time. This file contains 2 columns of data: (1) cumulative CPU time at each time step, and (2) total elapsed time at each time step. **NOTE:** CPU time calculations will only work for the current version of the program, which was compiled using Microsoft PowerStation FORTRAN, version 4.0.
- prefix.out** - This file is useful when checking simulations that are not working properly, because it outputs an all of the information from the data file. Also, this file is used to output the progress of the simulation as well as head output at specified output times.
- prefix.hds** - This file contains the heads along the water table at the specified output times.
- prefix.swn (n=1,2...)** - For surface water boundaries, if the user specifies that the program calculate the transient water level, this file will contain the volume-to-water-level ratio for the surface water body. The integer 'n' represents a counter if more than one surface water body is present.
- prefix.wbn (n=1,2...)** - For surface water boundaries, if the user specifies that the program calculate the transient water level, this file will contain the water level and volume of the surface water body n for each time step.

The procedure for using *GridBuilder* to view output is as follows:

In *GridBuilder*, click on 'File/Import'. Go to the directory that contains the **prefixn.imp** and click on the file which you want to import. After a series of prompts, the grid should appear in the *GridBuilder* window. On the main *GridBuilder* menu, choose 'File/Save' and create a new file name. Then convert the file to ASCII format, and save all subsequent files that are imported in ASCII format. In a DOS window, copy the prefixn.n01 (and/or .v01, .e01, .f01, .f02, .c01) file to a new file with the new prefix which you specified in *GridBuilder*. The extension must remain the same. (**NOTE:** the \*.c01 files must be renamed with .n01 extensions to be read as nodal files in *GridBuilder*).

## A.4 Dimensioning of Arrays

The number of arrays which may require re-dimensioning has significantly decreased because of the implementation of dynamic memory allocation with Version 1.2. As such, the list below is smaller than in previous versions of GW-WETLAND, and most simulations will not require re-dimensioning of arrays and subsequent re-compilation of the program. In some instances, however, the user may need to recompile. The list below contains the parameters and the maximum values as specified in the current version. If the user requires larger arrays for these parameters, re-dimensioning and re-compilation will be required.

parameter(mxchd=235)

parameter(maxbnd=251)

parameter(maxtime\_chd=4300)

parameter(maxwellperiod=1)

parameter(maxpt=250000)

parameter(maxts=4000)

parameter(mx\_conc\_zone=1)

parameter(max\_part\_zone=10)

parameter(mx\_conc\_node=50)

where:	mxchd	= maximum number of constant head nodes
	maxbnd	= maximum band width
	maxtime_chd	= maximum number of times to change the value of hydraulic head at a constant head node
	maxwellperiod	= maximum number of times to specify different pumping rates
	maxpt	= maximum total number of particles
	maxts	= maximum number of time steps
	max_part_zone	= maximum number of source zones for particles
	mx_conc_zone	= maximum number of fixed (Dirichlet) concentration zones
	mx_conc_node	= maximum number of fixed (Dirichlet) nodes



## A.5 Units

grid spacing	m
hydraulic conductivity	m/d
elevations (w.t., ground surface, base of domain)	m
initial value of head	m
initial value heads along w.t.	m
initial values of constant heads	m
value of recharge	m/d
value of evapotranspiration	m/d
pumping rates	m/d
time step	d
output times	d
times when recharge or evapotranspiration applied	d
dispersivity	m
solute concentration	mg/L
initial mass of each particle	mg
half-life	1/d

```
! flood simulation
! March 22, 2000
Flood.dat
f ;(1)Print nodal coordinates
f ;(2)Print elemental incidences
f ;(3)Print elemental K and S
f ;(4)Print nodes/elements to file
t ;(5)Calculate Heads
f ;(6)Print Heads
t ;(7)Print Heads to a file (*.hds)
f ;(8)Print only last heads to a file (*.hds)
f ;(9)Print water table convergence info
f ;(10)Input old heads to continue a run
f ;(11)print last heads for input for a later run
t ;(12)generate gradient/direction field
f ;(13)print gradients/directions
f ;(14)output gradients/directions
t ;(15)generate velocity field
f ;(16)output velocity field
t ;(17)run mass transport
f ;(18)print concentrations/particle distributions
f ;(19)output concentrations/particle distributions
f ;(20)output particle distribution
f ;(21)output last particles for input to continue a run
f ;(22)input last particles to continue a run
f ;(23)print mass, X,Z coordinates of particles
f ;(24) Run advection/dispersion
0 ;nobspts
181 201 ;nrow, ncol
1 100 ;ngeol,nitmax
f ;var_x (T for var. delx, F for const. delx - if F, enter only 1 value)
1.0
.2d0 ;delz
730.0 ;ftime
2 ;n_out_time
365.0
730.0
0.5 2.5 ;delt,dttmax
21.0 ;headi
0.200 .005 1.05 ;specydc,tolrnc,deltin
! elvgrd
34.5 34.5 34.5 34.5 34.5 34.5 34.5 34.5 34.5 34.5
34.5 34.5 34.5 34.5 34.5 34.5 34.5 34.5 34.5 34.5
34.5 34.5 34.5 34.5 34.5 34.5 34.5 34.5 34.5 34.5
34.5 34.5 34.5 34.5 34.5 34.4 34.35 34.3 34.25 34.2
34.15 34.1 34.05 34 33.95 33.9 33.85 33.8 33.775 33.75
33.725 33.7 33.675 33.65 33.625 33.6 33.575 33.55 33.525 33.5
33.4 33.35 33.3 33.25 33.2 33.1 33 32.9 32.8 32.7
32.6 32.5 32.4 32.3 32.2 32.1 32 31.8 31.65 31.5
31.4 31.3 31.1 30.9 30.7 30.5 30.3 30.2 30.1 30
29.9 29.8 29.75 29.7 29.65 29.6 29.5 29.415 29.4 29.385
!
29.37 29.355 29.34 29.325 29.31 29.295 29.28 29.265 29.25 29.235
29.22 29.205 29.19 29.175 29.16 29.145 29.13 29.115 29.1 29.085
29.07 29.055 29.04 29.025 29.01 28.995 28.98 28.965 28.95 28.935
28.92 28.905 28.89 28.875 28.86 28.845 28.83 28.815 28.8 28.785
28.77 28.755 28.74 28.725 28.71 28.695 28.68 28.665 28.65 28.635
28.62 28.61 28.6 28.59 28.58 28.57 28.56 28.55 28.54 28.53
28.52 28.51 28.5 28.49 28.48 28.47 28.46 28.45 28.44 28.42
28.4 28.3 28.2 28 27.7 27.4 27 26.5 25.7 24.9
24.2 23.5 22.9 22.3 21.7 21.2 20.7 20.1 19.6 19.2
18.8 18.5 18.3 18.2 18.1 18 17.9 17.8 17.7 17.6
17.5
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! ;wtmove
!INITIAL WATER TABLE
21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0
21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0
21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0
21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0 21.0
```



[illegible][illegible]



16

17

!8

19

110

- 62 -

[illegible]

- 63 -



[illegible][illegible][illegible]

16

!17

- 66 -

```

2-D example with 2 wetland boundaries
water levels in wetlands are calculated by code
wnf-i.dat

```

- 67 -

7.5  
 7.5  
 7.5  
 7.5  
 7.5  
 7.5  
 7.5

! ELEVATION OF BASE  
 0  
 0  
 0  
 0  
 0  
 0  
 0  
 0  
 0  
 0

;elvbas  
 F ;nditch  
 13 ;NRESTP  
 0 1  
 1 201 0  
 60 1  
 1 201 0.00354  
 121 1  
 1 201 0.0025725  
 152 1  
 1 201 0.00177  
 182 1  
 1 201 0.000683  
 244 1  
 1 201 0.00177  
 365 1  
 1 201 0  
 425 1  
 1 201 0.00354  
 486 1  
 1 201 0.0025725  
 517 1  
 1 201 0.00177  
 547 1  
 1 201 0.000683  
 609 1  
 1 201 0.00177  
 730 1  
 1 201 0

15 ;NETPER  
 0 1  
 1 201 0.00  
 91 1  
 1 201 0.0025  
 121 1  
 1 201 0.003  
 152 1  
 1 201 0.004  
 182 1  
 1 201 0.005  
 213 1  
 1 201 0.003  
 244 1  
 1 201 0.0025  
 274 1  
 1 201 0.00  
 456 1  
 1 201 0.0025  
 486 1  
 1 201 0.003  
 517 1  
 1 201 0.004  
 547 1  
 1 201 0.005  
 578 1  
 1 201 0.003  
 609 1  
 1 201 0.0025  
 639 1





Environment Canada Library, Burlington



3 9055 1017 5671 5



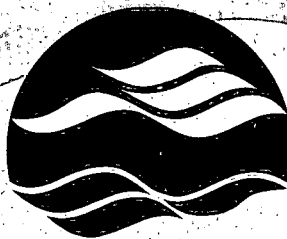
PRINTED IN CANADA  
IMPRIMÉ AU CANADA



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**