MODELING NITROGEN TRANSPORT AND TRANSFORMATION IN A HETEROGENEOUS, THREE-DIMENSIONAL, TILE-DRAINED AQUIFER

BY

DAVID JAMES HILL

B.S., Cornell University, 1999

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Environmental Engineering in Civil Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2002

Urbana, Illinois

Abstract

Recently, there has been increased interest in nitrate contamination of groundwater in the Midwest because of its link to surface water eutrophication, especially in the Gulf of Mexico. The vast majority of this nitrate is the product of biologically mediated transformation of fertilizers containing ammonia, occurring in the vadose zone of agricultural fields. For this reason, it is imperative that mathematical models, which can serve as useful tools to evaluate both the impact of agricultural fertilizer applications and nutrient-reducing management practices, are able to specifically address transport in the vadose zone. This thesis presents the development of a three-dimensional, explicit numerical model to simulate the movement and transformation of nitrogen species through the subsurface on the scale of an individual farm plot. At this scale, nitrogen fate and transport is controlled by a complex coupling of hydrologic, agricultural and biogeochemical processes. The nitrogen model is a component of a larger modeling effort that focuses upon conditions found in agricultural fields in Illinois. These conditions include non-uniform, multi-dimensional, transient flow in both saturated and unsaturated zones, geometrically complex networks of tile drains, coupled surface-subsurface-tile flow, and dynamic levels of dissolved oxygen in the soil profile. The advection-dispersion-reaction equation is solved using an operator-splitting approach, which is a flexible and straightforward strategy. Advection is modeled using a total variation diminishing scheme, dispersion is modeled using an alternating direction explicit method, and reactions are modeled using rate law equations. The model's stability and accuracy are evaluated and discussed. The model is applied to two scenarios for a hypothetical two-dimensional section containing a single tile and a drainage channel. The first scenario uses steady-state flow conditions and demonstrates the ability of the model to predict the effect of variable dissolved oxygen concentrations on nitrate in the subsurface. The second scenario uses transient flow conditions, in response to actual rainfall, and demonstrates the ability of the model to predict solute breakthrough concentrations under these conditions.

Acknowledgments

I would like to express my gratitude to my advisor, Professor Albert J. Valocchi, for providing his insight and guidance during this research. As this research is merely a part of a greater endeavor, I am also indebted to all the other researchers working towards other goals of this project, whose hard work and dedication have provided the groundwork for this research. In particular, I would like to thank Professor Richard A.C. Cooke, Professor Timothy R. Ellsworth, Professor Ben Yen, Yanqing Lian, Jaswinder Singh, and Feng Yue. I would especially like to acknowledge Professor Robert J. Hudson, who provided the field expertise necessary to tie this research to real-world applications. Also, I would like to thank Kellee Caton, who edited many drafts of this thesis. Finally, I would like to thank the Illinois Council on Food and Agricultural Research Water Quality Strategic Research Initiative for their financial support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background Information

Nitrogen pollution affects many aspects of the environment. In drinking water supplies, nitrogen species have been linked to methemoglobinemia in babies, and to non-Hodgkin's lymphoma (*David and Gentry*, 2000; *Ward et al.*, 1996). Nitrous oxide (N_2O), a by-product of microbial degradation of nitrate, contributes to the greenhouse effect and global warming (*Korom*, 1992). However, it is the effect of nitrogen pollution on surface water ecosystems that is the most apparent. The link between nonpoint nitrogen sources and nitrogen contamination of surface water leading to eutrophication, toxic algal blooms, and hypoxia is well known (*Carpenter et al.*, 1998). One of the most highly publicized instances of these results is the summertime appearance of a hypoxic zone in the Gulf of Mexico. The occurrence of this zone is not merely the result of local practices, but rather the result of practices occurring throughout the entire Mississippi-Atchafalaya River Basin (MARB). In fact, Goolsby *et al.* (1999) estimate that 35% of the entire nitrate flux to the Gulf of Mexico is attributable to the states of Iowa and Illinois, states that account for only 9% of the land area of the entire MARB. This finding confirms the need for better nitrogen management strategies in these states. Based on their findings that 84% of the nitrate contributed to the Mississippi river by the state of Illinois comes from agricultural sources, David and Gentry (2000) suggest that no plan to reduce Illinois' nitrate contribution will be successful without addressing agricultural nitrogen inputs.

Agricultural fertilizers and nitrogen fixing crops, such as soybeans, are used to maximize crop yield. Therefore, uninformed regulations could at best be inefficient at reducing the nitrogen problems, and at worst be detrimental to the financial viability of the state's farmers. For this reason it is necessary for regulators to have a tool to help farm managers find methods

to reduce the amount of nitrogen lost from the fields. This would reduce both the farm managers' annual expenditures on fertilizer and the nitrogen input to surface water. Such a tool would have to address the specific conditions encountered on a typical Illinois field.

It has been long understood that ammonia-based fertilizers are degraded by soil microorganisms into nitrate (NO_3^-) under oxic conditions, and further degraded by different soil microorganisms into gaseous nitrogen (N_2) and nitrous oxide (N_2O) under anoxic conditions. These processes are termed nitrification and denitrification, respectively. In fact, given a large enough soil system and a long enough residence time, it is possible to see near complete conversion of ammonia-based fertilizers into gaseous nitrogen products due to the coupling of nitrification and denitrification (*Korom*, 1992). However, in the state of Illinois, the situation is different. Because of a high water table, farmers in Illinois rely on tile drains in the soil beneath their crops to lower the water table during rain events and keep the crops from being flooded. Unfortunately, these tile drains pose a unique problem when it comes to the subsurface transport of nitrogen species. Since the vadose zone is largely oxic, much nitrification takes place here, resulting in a high concentration of nitrate in the vadose zone. In a regular soil system, the nitrate would then be transported into the saturated zone, a region of mostly anoxic conditions, where denitrification could occur. However, because of the tile drains, the system is short-circuited. Some of the pore water flows directly from the vadose zone into the tiles, while the rest is routed along shallow flow paths in the saturated zone to the tile, resulting in a much shorter residence time than would have occurred if the drains were not there (*Goolsby et al.*, 1999). Because the tile drains remove water from the soil system before sufficient time has elapsed for denitrification to occur, all of the nitrate is not converted into gaseous nitrogen species, and so the drainage water is left with a high nitrate concentration (*Gentry et al.*, 1998). In fact, this concentration often exceeds the United States Environmental Protection Agency's minimum contaminant level (MCL) of 10 mg/L (*David et al.*, 1997). This drainage water, which has been removed from the soil system where further denitrification could have taken place, is then deposited in a surface water system, such as a river, where oxic conditions prevail and little denitrification can occur. The flux of nitrate from tile drains has been identified as a cause of increased nutrient loads in surface water in the region (*Logan et al.*, 1994; *Fausey et al.*, 1996), which has in turn been cited as a contributing factor to the hypoxic conditions recently experienced in the Gulf of Mexico (*Rabalais et al.*, 1991; *Turner et al.*, 1998; *David and Gentry*, 2000).

In order to evaluate the impacts of agricultural fertilizer application in Illinois on the Gulf of Mexico, it is important to consider the nitrogen balance of a typical Illinois farm. Agricultural soils are typically saturated with nitrogen either by application of an industrial fertilizer or by fixation of atmospheric nitrogen by leguminous plants (*David and Gentry*, 2000; *Vitousek et al.*, 1997). Once in the soil, some of the nitrogen is stored by plants and microorganisms, while the rest is lost, either through transport of soluble nitrogen species into drains or through degradation by soil bacteria (*Korom*, 1992). Studies have shown the correlation of these two types of losses to the following parameters: correspondence of a rain event with fertilizer application, location of drains, fertilizer application method, type of fertilizer used, type of crop, type of soil, and soil microbial population; however, there is no consensus for which of these factors control nitrogen loss (*David and Gentry*, 2000; *Gentry et al.*, 2000; *Vitousek et al.*, 1997). For this reason, numerical modeling is an important tool for keeping track of agricultural nitrogen, as well as other agro-chemicals.

1.2 Existing Models

Given the large number of processes which may occur in the subsurface, and the numerous methods of modeling these processes, it is not surprising that numerical models describing the subsurface fate and transport of nitrogen species in an agricultural setting vary from simplified to complex. This section offers a brief overview of existing models.

MacQuarrie and Sudicky (2001) developed a model to simulate the movement of nitrogen from septic waste in both the saturated and unsaturated zones. In this model, the spatial domain is represented in all three dimensions and spans both the saturated and unsaturated zones. Flow and solute transport are considered in both the porous medium and the septic drains, and the two systems are coupled using superposition. Flow in the porous medium is governed by a modified version of the Richards equation (*Cooley*, 1983; *Huyakorn et al.*, 1984), with either the Brooks-Corey (1964) or van Genuchten (1980) equations describing the relationship between the pressure head and moisture content. The coupled equation is solved using Galerkin's method for discretization and the GCSTAB (*VanderKwaak et al.*, 1995) method for solving sparse matrices. The transport of solute in soil is governed by the advection-dispersion-reaction (ADRE) equation for the gas, solid and liquid phases, which is determined from a mass balance on a unit representative volume of porous media, though the assumption is made

that the solid phase is immobile and that the transport of the gas phase is controlled by molecular diffusion. The reactions considered in this model include both biotic and abiotic reactions, modeled using a kinetic approach. Biologically mediated reactions, such as nitrification and denitrification, are expressed using multiple Monod expressions, while abiotic reactions, such as those that describe the carbonate system, are expressed using first- or second-order kinetic models. In order to simplify the problem, the local equilibrium assumption (LEA) is used to describe the partitioning between the gas and liquid phases and the solid and liquid phases. Henry's Law is used to describe the gas/liquid equilibrium, and a linear isotherm model is used to describe the solid/liquid equilibrium. To further simplify the problem, Strang-Splitting, a method of operator-splitting, is used to decouple the reactions from the transport terms. The transport equation is solved in the same way as the Richards equation described above, and the reaction system is solved using VODE, an implicit solver developed by Brown *et al.* (1989).

RISK-N, a model developed by Gusman and Mariño (1999), is another model that considers flow and transport in both the saturated and unsaturated zones. This model approximates the three-dimensional problem domain with a quasi-three-dimensional model domain, where the vertical dimension of the problem domain is divided into four regions: the upper root zone, the lower root zone, the intermediate vadose zone and the saturated zone. Flow in the model is governed by a one-dimensional flow assumption, in which the average infiltration into the upper root zone is specified on a seasonal basis. Percolation from the upper root zone into the lower root zone is calculated as the infiltration minus the fraction of evapotranspiration (ET) that occurs in the upper root zone. Percolation from the lower root zone into the intermediate vadose zone is calculated as the percolation into this zone minus the remaining portion of ET. Finally, the percolation from the intermediate vadose zone to the saturated zone is equal to the percolation into to the intermediate vadose zone. Transport in the three regions of the unsaturated zone is modeled by complete mixing (spatial averaging) in each zone. In the saturated zone, however, transport is governed by the two-dimensional, vertically-averaged form of the advection-dispersion equation, which is solved analytically. This model allows nitrogen to be added to the system through the application of fertilizer, through wet and dry deposition, and through the mineralization of crop residue. Dissolution of solid fertilizer is assumed to be instantaneous. The RISK-N model uses a constant fraction approach to ammonia volatilization. Depending on the

type of fertilizer applied, between 10% and 20% of the fertilizer volatilizes before entering the subsurface. Mineralization of nitrogen is modeled as a first-order process, with rates for both the labile and recalcitrant nitrogen being calculated as described by Kersebaum and Richter (1991). Both nitrification and denitrification are modeled as first-order processes. The rate for nitrification is a constant, while the rate for denitrification in the unsaturated zone is adjusted for the temperature and degree of soil saturation, following the convention of the CropSyst model described by Marchetti *et al.* (1997). Due to uncertainty about the processes that control denitrification in the unsaturated zone, the rate for this reaction must be specified by the user. The RISK-N model also has the ability to account for plant uptake, but the rate for this process must also be specified by the user.

Instead of attempting to explicitly model the spatial domain of a field, several existing models choose to define a representative two-dimensional, horizontally-averaged element, from which the three-dimensional problem domain can be constructed. For fields containing tile drains, the symmetry of an idealized field containing long, uniformly-spaced, parallel drains is used to define the representative element. In such a field, symmetry dictates that hydraulic boundaries parallel to the drains exist halfway between the drains. Thus, the sides of the element can be defined by hydraulic boundaries, and the top can be defined by the soil surface, leaving the model to define the bottom boundary. Such an element would contain a tile drain running perpendicular to the plane of the element and intersecting halfway between the side boundaries.

One of the models that uses this two-dimensional element approach is presented by Mohanty *et al.* (1998). The side boundaries of the two-dimensional element are no-flow boundaries, as described above, while the bottom boundary is chosen to be impermeable. Flow in the porous medium is governed by the two-dimensional form of the Richards equation, and the van Genuchten (1980) expression is used to define the relationship between pressure head and moisture content, while preferential flow through macropores is governed by piecewise-continuous hydraulic functions. Transport and reaction of solute are governed by the two-dimensional form of the ADRE, which considers advective transport of the liquid phase, dispersion of the liquid and gas phases, and a sink for the tile drain. All reactions considered by this model are described with an apparent first-order decay chain coupled with the transport equation. The reactions considered occur in all three phases and include the following: hydrolysis of fertilizer (urea or ammonium sulphate) in both the solid and aqueous phases, nitrification of both sorbed

and aqueous ammonium ions, volatilization of ammonia, and denitrification of aqueous nitrate. Plant uptake is also considered and modeled using the empirical equations presented by Nelson and MacDonald (1978). Immobilization, mineralization, and biological nitrogen fixation, however, are ignored. The equations were discretized using the Galerkin finite element method and solved with the "mass conservative" modified Picard iteration method (*Celia et al.*, 1990). During the discretization process, the mesh was adapted to flow conditions to adjust the Peclet and Courant numbers in each cell in order to minimize numerical oscillation and dispersion. This resulted in a mesh that is finer near the land surface and tile drain, and coarser elsewhere.

DRAINMOD-N, developed by Brevé *et al.* (1994), is another model that uses the two-dimensional element approach with an impermeable bottom boundary. This model is an extension of DRAINMOD (*Skaggs*, 1980), which adds nitrogen balance calculations to the simulation of hydrology of artificially drained soils in regions with shallow water tables. The consideration of flow in this model, however, is only quasi-two-dimensional, as the flow in the unsaturated zone is only vertical, while in the saturated zone, flow is both vertical and lateral. Infiltration, surface-storage, and runoff are calculated using the Green-Ampt equation. When rainfall exceeds infiltration, it is allowed to accumulate as surface storage, but when the surface storage reaches a specified depth, the additional rainfall is considered as runoff. The drainage rate of the tile drain is calculated using Hooghoudt's steady state equation as described in Bouwer and von Schilfgaarde (1963). This equation assumes that the rate of drainage is limited by the rate of groundwater flow, not the capacity of the drains. Vertical soil water flux in the saturated zone is defined to drop off linearly from the Hooghoudt's drainage flux at the water table to zero at the bottom boundary. Evapotranspiration is set equal to potential evapotranspiration (PET) as long as the soil moisture content is above a specified threshold value, usually a function of the wilting point moisture content. The moisture content profile is specified by breaking the soil profile into two sections, the wet zone and the root zone. In the wet zone, a hydrostatic profile is used. In the root zone, the moisture profile is uniform and determined by ET. If the upward soil moisture flux from the saturated zone cannot supply the ET demand, the moisture is taken from the root zone until the soil moisture content in the root zone reaches a specified threshold value at which point ET goes to zero. The movement of solute in the subsurface is modeled using an explicit finite difference approximation to the ADRE in one dimension in the unsaturated zone and

in two dimensions in the saturated zone. Only nitrate-nitrogen is considered, as ammonium-nitrogen nitrifies quickly or stays sorbed to the soil, and thus is not present in drainage water. Fertilizer dissolution is modeled as a zeroth-order process controlled by the moisture content. Only when the soil moisture content is above a specified threshold value will this process occur. The net effect of mineralization and nitrogen immobilization is modeled as a zeroth-order process, limited by soil moisture content and temperature. The moisture content limiting factor accounts for the effect of aerobic/anaerobic conditions on the processes, while the temperature limiting factor accounts for the effect of temperature on the rate of biologically mediated reactions. Denitrification is modeled as a first-order process that will only occur when the soil moisture content is greater than a specified threshold value. The formulation for the mineralization and denitrification processes were presented by Johnsson *et al.* (1987). Plant uptake is calculated based on crop data such as yield, percent nitrogen in crop, time in growing season, and root depth, as described by Shaffer *et al.* (1991). Runoff nitrogen concentration calcuation is based on CREAMS (*Knisel*, 1980) and takes into account the nitrogen concentration of the root zone, the nitrogen concentration of the rain, and the processes of infiltration and extraction. Finally, nitrogen input by legumes is considered to supply only the difference between the crop nitrogen demand and the available nitrate-nitrogen, an assumption validated by Knisel (1993). The resulting explicit finite difference equation is solved by an adaptive algorithm that adjusts the time step based on the Courant number to ensure that the method remains stable.

Bear, Wang, and Shaviv present another model based on a two-dimensional representative element (*Bear et al.*, 1998; *Wang et al.*, 1998). This model, however, defines the bottom boundary as a flux boundary located above or at the water table, and does not consider tile drains, though the side boundaries are still defined as no-flow boundaries. Liquid phase flow is determined based on a time dependent mass balance that includes a sink term for crop uptake, as described by Feng and Bar-Yosef (1995). Chemical transport and reaction of ammonium, organic carbon, oxygen, urea, and nitrate in the gaseous and aqeous phases are calculated using the ADRE. Henry's Law is used to describe partitioning between the gas and aqueous phases, while adsorption/desorption is described using a linear isotherm. Nitrogen can be introduced into the subsurface through the application of ammonia-based fertilizers, urea, or controlled-release nitrogen fertilizers, and can be removed through crop uptake or reaction. This model considers the following abiotic reactions: organic nitrogen mineralization, the carbonate

system, and the ammoniacal nitrogen system. Nitrogen mineralization is treated as a first-order kinetic process, while the other reactions are equilibrium reactions. The biotic reactions considered by the model are: nitrification, denitrification, urea hydrolysis, and oxygen consumption. Nitrification is modeled using a second-order kinetic model developed by Neden (1990), which depends on the ammonium, nitrate, and dissolved oxygen concentrations, the pH, and the population of nitrifying bacteria. Denitrification is treated as a first-order kinetic process that is dependent on the nitrate concentration, temperature, and dissolved oxygen concentration. The dissolved oxygen content scales the rate of the denitrification reaction by the factor $1 - C_{DO}$ when the dissolved oxygen concentration (C_{DO}) is between zero and one mg/L, a method suggested by Bachmat and Chetboun (1976). Urea hydrolysis is modeled by a sum of two Michaelis-Menten equations, as described by Cabrera and Kissel (1984). Aerobic decay of organic carbon is modeled as a zeroth-order kinetic process. Because some of these processes include acid/base reactions, an acid mass balance is computed to keep track of the pH within the soil. The equations from these processes, as well as those for gas phase flow, heat transport, root growth, evapotranspiration, and infiltration are solved simultaneously. The entire model consists of nine partial differential equations (PDEs) and twenty-two algebraic equations, and is solved by an implicit finite-difference method on a non-uniform grid. Picard's method is used to linearize the non-linear equations for gas and liquid flow. In order to lessen the effect of numerical dispersion on the solution, a correction factor to the true hydrodynamic dispersion coefficient is used when the grid Péclet number is larger than 2.0 or the Courant number is larger than 1.0. To solve the resulting system, either a Preconditioned Conjugate Gradient or a Preconditioned Orthomin method is used, depending on the characteristics of the matrix.

A further simplification of the three-dimensional problem domain is made by the one-dimensional, vertical-into-the-soil model domain used by the Root Zone Water Quality Model (RZWQM) (*Ahuja et al.*, 1999). This model reduces the spatial dimensions in order to incorporate very complex considerations of geochemical and agricultural processes. In this model, the porous medium is divided into micro-pore and meso-pore zones, in order to introduce the effect of preferential flow. The Green-Ampt equation is used to calculate infiltration, and the Richards equation, along with the Brooks-Corey relationship (1963), is used to calculate the soil moisture profile. When applied to a tile-drained field, the Hooghoudt steady-state equation (*Bouwer and van Schilfgaarde*, 1963) is used to determine the drainage

flux at the center point between two parallel tiles, which is then applied to the model as a point sink. Evapotranspiration is calculated using a Penman-Monteith model. The abiotic chemical reactions considered by the RZWQM are: precipitation and dissolution of lime, gypsum, and gibbsite; ion exchange between calcium and sodium, calcium and magnesium, sodium and ammonium, and calcium and aluminum; and complexing of sodium and aluminum with organic matter. The biotic chemical reactions considered by RZWQM are modeled using a kinetic approach, where the rate constant is a function of the involved microbial population. Rather than tracking dissolved oxygen concentration in the soil profile, the RZWQM scales the rate constants by a factor that is dependent on the degree of saturation of the soil to account for the effect of aerobic or anaerobic conditions on these reactions. The production of methane gas is modeled as a zeroth-order process. Denitrification, the aerobic decay of organic matter, and the hydrolysis of urea are modeled as first-order processes. Nitrification is modeled as a combination of both a zeroth-order process and a first-order process. In addition to simulating the movement of water and chemicals in the subsurface, the RZWQM also includes a crop yield model, which includes the processes of plant uptake of moisture and nitrogen, as well as either the removal of stored nitrogen from the system when the plants are harvested, or the replacement of stored nitrogen when the plants die.

1.3 Scope of Thesis

Despite the wide variety of available models, few attempts have been made to create a combined saturated-unsaturated zone transport model that simulates nitrogen fate and transport in all three spatial dimensions in an artificially drained aquifer. In this thesis, a numerical reactive solute transport model is developed to simulate the fate of nitrogen and related chemical species through the subsurface under conditions similar to those found on agricultural fields in Illinois. These conditions include, but are not limited to: non-uniform, multi-dimensional, transient flow in both the saturated and unsaturated zones, geometrically complex networks of tile drains, and dynamic levels of dissolved oxygen in the soil profile. This model uses chemical, soil, and groundwater flow properties to calculate the concentration of the various solutes, both at node points within the soil volume and at fixed locations within tile drains that pass through the soil. The flow properties will be produced by a conjunctive overland-groundwater flow model and will be transferred to the solute transport model through a dynamic interface

to allow for transient simulations. The conjunctive model is being developed by other investigators and is based upon the work of Morita and Yen (2000, 2002). In order to render this model accessible to both farm managers and regulators, special consideration has been made to minimize its computational demand.

Chapter 2

Model Development

2.1 Solute Transport Governing Equation

The partial differential equation that describes the fate and transport of a chemical through a soil system in three dimensions can be written as follows:

$$\frac{\partial \theta_w C^p_{(aq)}}{\partial t} + \frac{\partial \theta_g C^p_{(g)}}{\partial t} + \frac{\partial \rho_b C^p_{(s)}}{\partial t} = \frac{\partial}{\partial x_i}\Big(\theta_w D^w_{ij} \frac{\partial C^p_{(aq)}}{\partial x_j} + \theta_g D^g_{ij} \frac{\partial C^p_{(g)}}{\partial x_j}\Big) - \frac{\partial}{\partial x_i}\Big(\theta_w \nu_i C^p_{(aq)}\Big) - \Sigma q_d C^p_{(aq)} \pm \Sigma R_n \tag{2.1}$$

where:

$C^p_{(aq)}$	is the aqueous phase concentration of species p [M/L^3 water]
$C^p_{(g)}$	is the gas phase concentration of species p [M/L^3 air]
$C^p_{(s)}$	is the sorbed phase concentration of species p [M/L^3 soil]
θ_w	is the moisture content of the soil [L^3 water/L^3 aquifer]
θ_g	is the gas content of the soil [L^3 gas/L^3 aquifer]
ρ_b	is the bulk density of the soil [M soil/ L^3 aquifer]
t	is time [T]
x_i	is the location vector [L]
D^w_{ij}	is the aqueous phase dispersion tensor [L^2/T]
D^g_{ij}	is the gas phase diffusion tensor [L^2/T]
ν_i	is the pore water velocity [L/T]
q_d	is the sink term given by the volumetric flow rate into the tile drain (d) per volume of aquifer [L/T]
ΣR_n	is the chemical reaction source or sink term [M/T]

Note that there are two assumptions implicit in the above equation. The first assumption is that advection is not a significant transport mechanism

for the gaseous phase. The second assumption is that the sorbed phase is immobile. Both of these assumptions are made in order to simplify the right-hand side of Equation (2.1). The local equilibrium assumption is used to further simplify this equation. It asserts that if it can be assumed that the time it takes for the three phases to reach equilibrium is short in comparison with the transport time scale, then at any time, the concentration of the gaseous or sorbed phase can be expressed in terms of the aqueous phase concentration using an equilibrium relationship. In this research, Henry's Law, shown in Equation (2.2), and a linear isotherm model, shown in Equation (2.3), are used to describe the equilibrium between the gaseous and aqueous phases and the sorbed and aqueous phases, respectively.

$$C^p_{(g)} = \theta_g \frac{mw_w H_p}{RT} C^p_{(aq)} \tag{2.2}$$

where:

mw_w	is the molecular weight of water [M/mole]
H_p	is the Henry's Law coefficient of species p [atmosphere/mole fraction]
R	is the ideal gas constant [M*L^2/K*mole*T^2]
T	is the temperature [K]

$$C^p_{(s)} = K_d \rho_b C^p_{(aq)} \tag{2.3}$$

where:

K_d	is the soil-water partition coefficient [L^3/M]

Note that the species index p has been dropped for simplicity.

By using the relationships developed in Equations (2.2) and (2.3), the three-dimensional advection-dispersion-reaction equation (ADRE) shown in Equation (2.1) can now be written as follows:

$$\frac{\partial}{\partial t}\left(\left(\theta_w + K_d\rho_b + \theta_g \frac{mw_w H}{RT}\right) C^p_{(aq)}\right) = \frac{\partial}{\partial x_i}\left(\theta_w D^w_{ij} \frac{\partial C^p_{(aq)}}{\partial x_j} + \theta_g D^g_{ij} \frac{mw_w H_p}{RT} \frac{\partial C^p_{(aq)}}{\partial x_j}\right) - \frac{\partial}{\partial x_i}\left(\theta \nu_i C^p_{(aq)}\right) - \Sigma q_d C^p_{(aq)} + \Sigma R_n \tag{2.4}$$

2.2 Numerical Solution Strategy

There are many different methods that can be used to solve Equation (2.4), and each has its own particular strengths and weaknesses. Because of the

desire to make this model accessible to both farm managers and regulators, it is necessary to limit the number of potential methods to those that require minimal system resources.

Equation (2.4), shown above, is presented in its three-dimensional form. One method of reducing system resources is to solve this equation in only two dimensions and make an approximation for the third dimension. This approximation usually takes the form of an averaging scheme in the dimension of least importance. For example, if the flow is mostly horizontal, the vertical dimension would be averaged, while if the flow is mostly vertical, one of the horizontal dimensions would be averaged. However, the utility of these averaged models has been questioned, because dispersion is inherently a three-dimensional process, and because this sort of averaging can give a false picture of what is really happening (*Burnet and Frind*, 1987). Burnet and Frind (1987) point out that two-dimensional, vertically-averaged models give good average concentrations, but miss some peaks, while two-dimensional, vertical cross-section models predict inflated concentrations. Because of these limitations, it was decided that the model formulated for this research should be solved in all three dimensions.

For this reason, it was determined that a resource-efficient solution to the three-dimensional ADRE must be used. The most efficient methods for solving the ADRE involve the use of analytical solutions, but unfortunately, because of the complexity of the partial differential equation, all analytical solutions require a host of simplifying assumptions which limit their usefulness (*Leij and Bradford*, 1994). For this reason, there is a field of research devoted to developing numerical approximations for this type of equation. Of these numerical approximations, one of the most straightforward, and hence, the most resource-efficient, is the finite difference method (FDM).

The first step in solving an FDM is to define the time and space discretization. The temporal domain of the model is divided into uniform sections, or time steps, of duration Δt. Next, the three-dimensional spatial domain is divided according to a block-centered approach, whereby the domain is broken into small rectangular volumes, or cells, which have a measurement location, or node, at the center. All the cells in the domain are the same size and have sides of lengths Δx, Δy, and Δz, which are specified by the user. The location of each node is represented by a tripartite index of the form (i,j,k). The indices i, j, and k refer to the Fortran array referencing paradigm, where i represents the rows, j represents the columns, and k represents the layers. Therefore, a unit increase in index i represents a Δy increase in distance along the y-axis, a unit increase in index j represents

a Δx increase in distance along the x-axis, and a unit increase in index k represents a Δz increase in distance along the z-axis. Figure (2.1) shows this cell referencing system in further detail.

Once the discretization is complete, the partial differential equations can be written as a truncated Taylor series. For example, the spatial derivatives in the ADRE can be approximated as the difference in property A, $(A_i - A_{i+1})$ divided by the spatial distance between location i and location $i+1$, which is equal to Δi, the discretization size in the i direction. Similarly, temporal derivatives can be approximated as the difference in property A, $(A^t - A^{t+1})$ divided by the temporal distance between those two measurements, which is equal to the temporal discretization size Δt. In this way, all of the derivative terms in Equation (2.4) can be written in terms of differences between concentration measurements and discretization sizes. There are two different ways to express the spatial derivatives in terms of time. One method is to solve all the spatial derivatives at the new time level $(t+1)$, while the other method is to solve the spatial derivatives at the old (t) time level. These two methods are termed implicit and explicit, respectively.

Implicit FDMs are usually more robust, but they require more computer memory than explicit FDMs, because they result in a system of linear equations that must be solved simultaneously. Explicit methods require less computing resources because the equations they create can be solved independently; however, they often have prohibitive stability restrictions on the time and space discretization. The restriction on the discretization can often be described with the following two dimensionless ratios:

$$\mathcal{D}_{ij} = \frac{D_{ij}\Delta t}{(\Delta x_i)(\Delta x_j)} \tag{2.5}$$

and:

$$\mathcal{U}_i = \frac{\nu_i \Delta t}{\Delta x_i} \tag{2.6}$$

Equation (2.5) relates the time and space discretization to dispersion, while Equation (2.6) relates the discretization to advection. For explicit methods, the size of these parameters is restricted (e.g. in one-dimension, $\mathcal{D}$ must be less than or equal to one-half and $\mathcal{U}$ must be less than or equal to one) (see *Zheng and Bennett*, 1995). However, for implicit methods, these parameters are unrestricted for stability, though accuracy considerations may still require some limits. For this reason, it can be seen that an implicit method would allow a larger time step than an explicit method. However,

this restriction may not be as prohibitive as it seems. If the time step of the model is short, it has been shown that the processes of advection, dispersion, and reaction can be separated and solved sequentially - a method known as operator splitting (OS) (*Valocchi and Malmstead*, 1992; *Chilakapati et al.*, 2000). Thus, each of the processes can be solved by a unique numerical method well suited to that type of problem. Therefore, in order to satisfy the requirement that this model use minimal computational resources, an operator-split, explicit finite difference method was chosen to solve Equation (2.4).

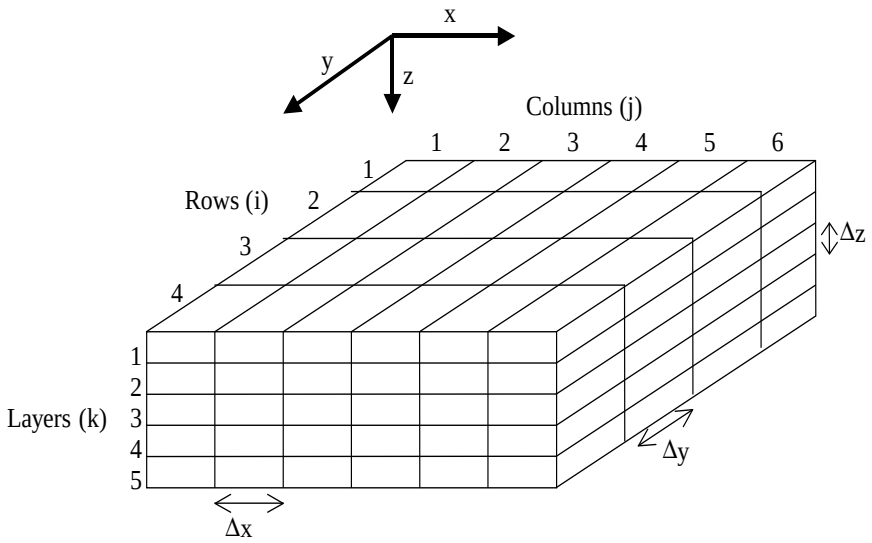

Figure 2.1: Diagram of relationships between the spatial discretization and the cell referencing and coordinate axis paradigms.

2.3 Operator Splitting

The ADRE has been described as an "embarrassingly" difficult problem to solve (*Mitchell*, 1984, p. 2), because the advection equation is a hyperbolic partial differential equation (PDE), while the dispersion equation is a parabolic PDE. Numerical methods that are good at solving parabolic PDEs may be unsuitable for solving hyperbolic PDEs, and vice-versa. For exam-

ple, Ding and Liu (1989) relate cases where parabolic solvers did not perform well under advection dominated conditions. Solving for chemical reactions further complicates the solution, because most reactions are described by highly non-linear systems of ordinary differential equations (ODEs). In order to incorporate the reaction equations directly into the transport equation, the ODEs must first be linearized, a very computationally intensive process (*MacQuarrie and Sudicky*, 2001). However, to overcome the difficulties of solving the ADRE, OS can be used to split the processes, and a suitable numerical method can be chosen for each of the subproblems. Therefore, instead of choosing a numerical method that is the best compromise between a hyperbolic and parabolic solver, the choice of method for each subproblem can be based on its ability to solve a particular type of differential equation, as well as on other criteria, such as stability or memory requirements.

Despite the many benefits of OS, it is important to consider the error associated with time splitting. Valocchi and Malmstead (1992) report an inherent mass balance error associated with using OS to time split reactions when using a continuous mass flux boundary condition; however, they show that the error decreases with decreasing time step size. Despite errors of this kind, Chilakapati *et al.* (2000) suggest that the benefits of having an accurate solution to each of the three subproblems outweighs the errors that arise from time splitting. Furthermore, because the time scale of a rain event is small, the model time step will also have to be small in order to capture local flow variability. In addition, because each of the transport subproblems is solved using an explicit FDM, numerical stability will also depend on the use of a small time step. Therefore, it is expected that the splitting errors will be small, and that overall, OS will be advantageous to this research.

The operator split explicit reactive transport model (OSERTM) splits Equation (2.4) into the following subproblems:

$$\frac{\partial}{\partial t}\left(\left(\theta_w + K_d\rho_b + \theta_g\frac{mw_wH_p}{RT}\right)C^p_{(aq)}\right) = - \frac{\partial}{\partial x_i}\left(\theta_w\nu_iC^p_{(aq)}\right) - \Sigma q_dC^p_{(aq)} \quad (2.7)$$

$$\frac{\partial}{\partial t}\left(\left(\theta_w + K_d\rho_b + \theta_g\frac{mw_wH_p}{RT}\right)C^p_{(aq)}\right) = \frac{\partial}{\partial x_i}\left(\left(\theta_w D^w_{ij} + \theta_g D^g_{ij}\frac{mw_wH_p}{RT}\right)\frac{\partial C^p_{(aq)}}{\partial x_j}\right) \quad (2.8)$$

$$\frac{\partial}{\partial t}\left(\left(\theta_w + K_d\rho_b + \theta_g\frac{mw_wH_p}{RT}\right)C^p_{(aq)}\right) = \Sigma R_n \quad (2.9)$$

where Equation (2.7) represents the advection subproblem, Equation (2.8) represents the dispersion subproblem, and Equation (2.9) represents the reaction subproblem. The three subproblems are solved sequentially in the following order: advection, dispersion, reaction.

The OSERTM uses a different subroutine for each of the three subproblems. This modularity allows for convenient swapping of different numerical solvers for each of the subproblems. The input to each of the subroutines is the output from the previous subroutine, so the hierarchy is circular, and the elapsed time of the simulation increments every cycle before the advection subproblem is solved. The overall model boundary conditions are implemented in the advection subroutine. The dispersion subroutine always has a zero flux boundary condition. Therefore, mass can only enter or leave the system through advection. This is a reasonable approximation, considering that transport of solute to a drain in an unconfined aquifer is most likely advection-dominated. Also, it is clear from Equation (2.7) that mass is removed by the drain in the advection subroutine. A more complete description of the model will be given in Chapter 3.

2.4 Advection

The advection equation describes the movement of solute due to the bulk groundwater flow. As noted in Section 2.2, a finite difference approximation to the true equation is used. Unfortunately, the behavior of this type of numerical approximation does not always match that of the true solution. Numerical dispersion and numerical oscillation are two common cases where an artifact of the FDM causes the numerical solution to deviate from the true solution. Numerical dispersion causes the numerical solution to appear more dispersed than the true solution, while numerical oscillation causes the numerical solution to oscillate above and below the true solution.

Numerical dispersion is often the result of applying a low-order approximation to an advection-dominated system, but Chilakapati and Yabusaki

(1999) show that non-uniform flow fields can also cause significant numerical dispersion. Because numerical dispersion causes errors, such as underestimation of the time it takes for a solute to reach a measurement point, overestimation of the time it takes for the maximum concentration of a continuous source to reach a measurement point, and underestimation of the maximum concentration to reach a measurement point from a pulse input, some higher-order FDMs have been proposed, such as the second-order upwind method. These methods do not show pronounced numerical dispersion, but they do have the tendency to exhibit numerical oscillation near the location of a sharp front, resulting in over- and under-shoots of the maximum concentration near the front. In order to manage the effects of these mathematical artifacts on the model solution, Harten (1983) suggests a numerical scheme, called the Total-Variation-Diminishing (TVD) method, which combines a high-order polynomial interpolation of the concentration at node points with a flux-limiting function to damp out oscillations.

In the TVD method, the first step to solve the advection equation is to discretize the PDE shown in Equation (2.7). For simplicity, only the one-dimensional form of this equation is shown; however, these methods can be extended to all three spatial dimensions. The discretized advection equation takes the following form:

$$\frac{1}{\Delta t}*\left(\left(\theta_w^{t+1}+K_d\rho_b+\theta_g^{t+1}\frac{mw_wH}{RT}\right)C_i^{t+1}-\left(\theta^t+K_d\rho_b+\theta_g^t\frac{mw_{H_2O}H}{RT}\right)C_i^t\right)= -\frac{\nu_{i+\frac{1}{2}}C^t_{i+\frac{1}{2}}-\nu_{i-\frac{1}{2}}C^t_{i-\frac{1}{2}}}{\Delta x}-\Sigma q_dC_i^{t+1} \quad (2.10)$$

Note that the subscript (aq) and superscript (p) have been dropped from the concentration term and replaced with a node reference index and a time reference index, respectively. The second step of the TVD method is approximating the concentration at the cell boundaries ($C_{i\pm\frac{1}{2}}$). The simplest TVD method is the first-order upstream method, where $C_{i+\frac{1}{2}} = C_i$ and $C_{i-\frac{1}{2}} = C_{i-1}$; however, this method exhibits numerical dispersion because of the low order approximation of the inter-nodal concentrations. There have been many suggestions for higher-order approximations (see *Van Leer*, 1974; *Roe*, 1984; *Leonard*, 1979; *Leonard*, 1991). These schemes suggest that as the order of the approximation increases, so does the accuracy of the solution. However, Leonard (1991) shows that for all orders, even-order approximations tend to be inferior to odd-order approximations of similar order. Conveniently, his third-order approach requires the same 5-point

stencil as a second-order method and can be easily solved explicitly. However, it should be noted that despite their ability to resolve sharp fronts, explicit TVD methods share the same stability limit as other explicit methods: $\mathcal{U} \leq 1$.

In this research, three different TVD methods are considered: a first-order upwinding scheme, Roe's Suberbee second-order upwinding scheme (*Roe*, 1984), and Leonard's universal limiter for transient interpolation modeling of the advective transport equations (ULTIMATE) (*Leonard*, 1991). Chilakapati *et al.* (2000) present a parsimonious formulation that allows these three methods to be specified by the following equations:

$$\begin{aligned} C^t_{i+\frac{1}{2}} &= C^t_i + \frac{1}{2}(1-\lambda_{i+\frac{1}{2}})\Psi_{i+\frac{1}{2}}(C_{i+1}-C_i) \qquad \text{if} \qquad \nu_{1+\frac{1}{2}} > 0 \\ C^t_{i+\frac{1}{2}} &= C^t_i + \frac{1}{2}(1-\lambda_{i+\frac{1}{2}})\Psi_{i+\frac{1}{2}}(C_i-C_{i+1}) \qquad \text{if} \qquad \nu_{1+\frac{1}{2}} < 0 \end{aligned} \tag{2.11}$$

where $\lambda = \frac{\theta_w \nu \Delta t}{\Delta x}$ is the Courant number, and Ψ is the flux limiting function, which can take one of the follwoing forms:

$$\boldsymbol{\Psi} = \begin{cases} 0 & \text{first-order upwind} \\ max\left(0, min(2r,1), min(r,2)\right) & \text{Roe's Superbee} \\ max\left(0, min\left(2, 2r, \frac{2-\lambda_{i+1/2}+r\left(1+\lambda_{i+1/2}\right)}{3}\right)\right) & \text{ULTIMATE} \end{cases} \tag{2.12}$$

where:

$$\begin{aligned} r &= \frac{C_i - C_{i-1}}{C_{i+1} - C_i} \qquad \text{if} \qquad \nu_{i+\frac{1}{2}} > 0 \\ r &= \frac{C_{i+2} - C_{i+1}}{C_{i+1} - C_i} \qquad \text{if} \qquad \nu_{i+\frac{1}{2}} < 0 \end{aligned}$$

Using this algorithm, the TVD method can be selected simply by indicating the form of the flux limiting function.

The advection subproblem uses the velocity at each of the cell faces, the moisture content in each of the cells, the location and flow rate of any drains in the model domain, and the porosity and bulk density in each of the cells to calculate the concentration at each node point in the model domain. As mentioned previously, the flux boundary condition of the overall model is implemented in the advection subroutine. This is done by specifying the chemical mass flux through the boundaries, which is used to calculate the concentration value just outside the model domain using the following

relationship:

$$C_g^t = \frac{F_b}{\theta_w^t \nu_b^t} \tag{2.13}$$

where:

C_g^t	is the concentration of the ghost node at time t [M/L^3]
F_b	is the chemical mass flux per unit area through the boundary [M/L^{2}T]
$\theta_w^t \nu_b^t$	is the specific discharge of water through the boundary [L/T]

This formulation of the boundary condition places one ghost-node adjacent to every boundary node in the model domain, as shown in Figure 2.2. It can be seen, however, that neither Roe's Superbee nor Leonard's ULTIMATE method will work near the domain boundaries, because they require two nodes upwind and one node downwind of the cell face $i \pm \frac{1}{2}$. When too few nodes are available, regardless of which TVD method is specified by Equation (2.12), the first-order upwind method, which needs only one node upwind and one node downwind of cell face $i \pm \frac{1}{2}$, should be used. This condition is sufficient to avoid any mathematical errors near the boundaries, because the existence of ghost nodes ensures that there will always be one node on either side of the cell face that defines the model's physical boundary.

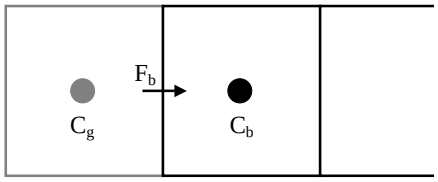

Figure 2.2: Diagram of boundary configuration for advection subproblem.

2.5 Dispersion

The dispersion equation describes the movement of solute associated with molecular diffusion and mechanical dispersion. Though this equation is merely an approximation of the microscale interactions that are occurring, it is commonly accepted as a valid description of the processes on a macroscale (*Zheng and Bennett*, 1995). To solve this equation with an FDM, it must first be discretized. The discretization of the temporal first derivative is straightforward, while that of the spatial second derivative is more difficult.

It can be seen that the spatial second derivative is composed of two major components: direct-terms (terms where both derivatives are taken with respect to the same direction) and cross-terms (terms where the two derivatives are taken with respect to different directions). The direct-term in any arbitrary direction x_i can be constructed using the differences between x_i and x_{i-1} and between xi and $xi+1$ as follows:

$$\frac{\partial}{\partial x_i}\hat{D}_{ii}\frac{\partial C}{\partial x_i} = \frac{\hat{D}_{ii(i+\frac{1}{2},j,k)}}{\Delta x_i^2}(C_{i+1,j,k} - C_{i,j,k}) - \frac{\hat{D}_{ii(i-\frac{1}{2},j,k)}}{\Delta x_i^2}(C_{i,j,k} - C_{i-1,j,k}) \tag{2.14}$$

where $\hat{D}_{ij} = \theta_w D_{ij}^w + \theta_g D_{ij}^g$ is the apparent dispersivity. The cross-term in any arbitrary directions x_i and x_j is expressed in terms of differences between concentrations between diagonally adjacent nodes in the x_i, x_j plane, as shown below:

$$\begin{aligned}\frac{\partial}{\partial x_i}\hat{D}_{ij}\frac{\partial C}{\partial x_j} \approx \\ \frac{1}{\Delta x_i \Delta x_j}\Bigg(\hat{D}_{ij(i+\frac{1}{2},j,k)}\big(C_{i+\frac{1}{2},j+\frac{1}{2},k} - C_{i+\frac{1}{2},j-\frac{1}{2},k}\big) - \\ - \hat{D}_{ij(i-\frac{1}{2},j,k)}\big(C_{i-\frac{1}{2},j+\frac{1}{2},k} - C_{i-\frac{1}{2},j-\frac{1}{2},k}\big)\Bigg)\end{aligned} \tag{2.15}$$

To determine the concentrations at the midway points between diagonal nodes, four nodes must be averaged. For example:

$$C_{i+\frac{1}{2},j+\frac{1}{2},k} = \frac{C_{i,j,k} + C_{i,j+1,k} + C_{i+1,j+1,k} + Ci+1,j,k}{4} \tag{2.16}$$

Using this averaging scheme, and collecting like terms, Equation (2.15) can be rewritten as follows:

$$\begin{aligned}\frac{\partial}{\partial x_i}\hat{D}_{ij}\frac{\partial C}{\partial y} \approx \\ \frac{\hat{D}_{ij(i+\frac{1}{2},j,k)}}{4\Delta x_i \Delta x_j}\Bigg(C_{i+1,j+1,k} + C_{i,j+1,k} - C_{i+1,j-1,k} - C_{i,j-1,k}\Bigg) - \\ - \frac{\hat{D}_{ij(i-\frac{1}{2},j,k)}}{4\Delta x_i \Delta x_j}\Bigg(C_{i,j+1,k)} + C_{i-1,j+1,k} - C_{i,j-1,k)} - C_{i-1,j-1,k}\Bigg)\end{aligned} \tag{2.17}$$

This formulation of the direct- and cross-terms was based on that of Zheng and Bennett (1995). Now, the entire three-dimensional discretized form of

the dispersion equation can be seen to be:

$$\frac{\left(\theta_w^{t+1} + K_d\rho_b + \theta_g^{t+1}\frac{mw_wH}{RT}\right) * C_{x,y,z}^{t+1} - \left(\theta_w^{t} + K_d\rho_b + \theta_g^{t}\frac{mw_wH}{RT}\right) * C_{x,y,z}^{t}}{\Delta t} =$$

$$\frac{\hat{D}_{xx(x+\frac{1}{2},y,z)}}{\Delta x^2}(C_{x+1,y,z} - C_{x,y,z}) - \frac{D_{xx}{}_{x-\frac{1}{2}}}{\Delta x^2}(C_{x,y,z} - C_{x-1,y,z}) +$$

$$+ \frac{\hat{D}_{xy(x+\frac{1}{2},y,z)}}{4\Delta x\Delta y}(C_{x+1,y+1,z} + C_{x,y+1,z} - C_{x+1,y-1,z} - C_{x,y-1,z}) -$$

$$- \frac{\hat{D}_{xy(x-\frac{1}{2},y,z)}}{4\Delta x\Delta y}(C_{x,y+1,z} + C_{x-1,y+1,z} - C_{x,y-1,z} - C_{x-1,y-1,z}) +$$

$$+ \frac{\hat{D}_{xy(x+\frac{1}{2},y,z)}}{4\Delta x\Delta y}(C_{x+1,y+1,z} + C_{x,y+1,z} - C_{x+1,y-1,z} - C_{x,y-1,z}) -$$

$$- \frac{\hat{D}_{xy(x-\frac{1}{2},y,z)}}{4\Delta x\Delta y}(C_{x,y+1,z} + C_{x-1,y+1,z} - C_{x,y-1,z} - C_{x-1,y-1,z}) +$$

$$+ \frac{\hat{D}_{yy(x,y+\frac{1}{2},z)}}{\Delta y^2}(C_{x,y+1,z} - C_{x,y,z}) - \frac{D_{yy}{}_{x,y-\frac{1}{2},z}}{\Delta y^2}(C_{x,y,z} - C_{x,y-1,z}) +$$

$$+ \frac{\hat{D}_{yx(x,y+\frac{1}{2},z)}}{4\Delta y\Delta x}(C_{x+1,y+1,z} + C_{x+1,y,z} - C_{x-1,y+1,z} - C_{x-1,y,z}) -$$

$$- \frac{\hat{D}_{yx(x,y-\frac{1}{2},z)}}{4\Delta y\Delta x}(C_{x+1,y,z} + C_{x+1,y-1,z} - C_{x-1,y,z} - C_{x-1,y-1,z}) +$$

$$+ \frac{\hat{D}_{yz(x,y+\frac{1}{2},z)}}{4\Delta y\Delta z}(C_{x,y+1,z+1} + C_{x,y,z+1} - C_{x,y+1,z-1} - C_{x,y,z-1}) -$$

$$- \frac{\hat{D}_{yz(x,y-\frac{1}{2},z)}}{4\Delta y\Delta z}(C_{x,y,z+1} + C_{x,y-1,z+1} - C_{x,y,z-1} - C_{x,y-1,z-1}) +$$

$$+ \frac{\hat{D}_{zz(x,y,z+\frac{1}{2})}}{\Delta z^2}(C_{x,y,z+1} - C_{x,y,z}) - \frac{D_{zz}{}_{x,y,z-\frac{1}{2}}}{\Delta z^2}(C_{x,y,z} - C_{x,y,z-1}) +$$

$$+ \frac{\hat{D}_{zx(x,y,z+\frac{1}{2})}}{4\Delta z\Delta x}(C_{x+1,y,z+1} + C_{x+1,y,z} - C_{x-1,y,z+1} - C_{x-1,y,z}) -$$

$$- \frac{\hat{D}_{zx(x,y,z-\frac{1}{2})}}{4\Delta z\Delta x}(C_{x+1,y,z} + C_{x+1,y,z-1} - C_{x-1,y,z} - C_{x-1,y,z-1}) +$$

$$+ \frac{\hat{D}_{zy(x,y,z+\frac{1}{2})}}{4\Delta z\Delta y}(C_{x,y+1,z+1} + C_{x,y+1,z} - C_{x,y-1,z+1} - C_{x,y-1,z}) -$$

$$- \frac{\hat{D}_{zy(x,y,z-\frac{1}{2})}}{4\Delta z\Delta y}(C_{x,y+1,z} + C_{x,y+1,z-1} - C_{x,y-1,z} - C_{x,y-1,z-1}) \quad (2.18)$$

The aqueous phase dispersion coefficient is expressed in the following

form:

$$D_{xx} = \alpha_L \frac{\nu_x^2}{\bar{\nu}} + \alpha_{TH} \frac{\nu_y^2}{\bar{\nu}} + \alpha_{TV} \frac{\nu_z^2}{\bar{\nu}} + D^\star \tag{2.19}$$

$$D_{yy} = \alpha_{TH} \frac{\nu_x^2}{\bar{\nu}} + \alpha_L \frac{\nu_y^2}{\bar{\nu}} + \alpha_{TV} \frac{\nu_z^2}{\bar{\nu}} + D^\star \tag{2.20}$$

$$D_{zz} = \alpha_{TV} \frac{\nu_x^2}{\bar{\nu}} + \alpha_{TH} \frac{\nu_y^2}{\bar{\nu}} + \alpha_L \frac{\nu_z^2}{\bar{\nu}} + D^\star \tag{2.21}$$

$$D_{xy} = D_{yx} = (\alpha_L - \alpha_{TH}) * \frac{\nu_x \nu_y}{\bar{\nu}} \tag{2.22}$$

$$D_{xz} = D_{zx} = (\alpha_L - \alpha_{TV}) * \frac{\nu_x \nu_z}{\bar{\nu}} \tag{2.23}$$

$$D_{yz} = D_{yz} = (\alpha_L - \alpha_{TH}) * \frac{\nu_y \nu_z}{\bar{\nu}} \tag{2.24}$$

where α_L, α_{TH}, and α_{TV} are the longitudinal, transverse horizontal, and transverse vertical dispersivities, respectively. This formula was presented by Burnet and Frind (1987) as an approximation of the anisotropic form of the dispersion tensor defined by Bear and Bachmat (1986). This approximation is used instead of the true anisotropic form of the equation, because the anisotropic formula requires five independent dispersivities, a level of information unlikely to be obtained in a field study.

The gas phase diffusion coefficient is expressed using the Millington and Quirk (1961) relationship:

$$\theta_g D_{ij}^g = \theta_g \frac{\theta_g^{\frac{7}{3}}}{\eta^2} * D_g^0 * \delta_{ij} \tag{2.25}$$

where D_g^0 is the free air diffusion coefficient [L^2/T], η is the porosity, and δ_{ij} is the Kronecker delta.

In order to solve the discretized equation shown in Equation (2.18), a numerical method must be chosen. As stated earlier, this research considered only explicit FDMs, thereby limiting the number of available methods that could be chosen. Most explicit FDMs are very restrictive in their stability requirements, so a survey of stable explicit methods for solving the dispersion equation was performed. During this survey, a report by Morita and Yen was discovered, which compared an alternating difference explicit (ADE) method presented by Larkin in 1964 to Brian's ADI method (1964), the purely ADI method, Barakat and Clark's ADE method (1966), and a purely explicit method for solving the three-dimensional Richards equation. Morita and Yen concluded that the ADE method worked as well as the ADI method, even though it required shorter time steps. Because of the favorable performance of the ADE method, it was chosen for this research.

Larkin's ADE method solves the average of an implicit formulation and an explicit formulation of the dispersion equation, giving it increased stability without the need to solve simultaneous equations. (This approach is similar to that of the Crank-Nicholson method.) To illustrate the method, a simplified one-dimensional version of the dispersion equation will be used.

$$\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\hat{D}_{xx}\frac{\partial C}{\partial x} \tag{2.26}$$

If this equation were to be solved with a fully implicit method, the concentrations in the spatial second derivative would be taken at the new time level, whereas if the equation were to be solved with a fully explicit method, these terms would be taken at the old time level. Taking the average of the two methods results in the following form:

$$\begin{aligned}\frac{1}{\Delta t}\left(\frac{C_x^{t+1}+C_x^{t+1}}{2}-\frac{C_x^t+C_x^t}{2}\right) = \\ \frac{\hat{D}_{xx(x+\frac{1}{2})}}{\Delta x^2}\left(\frac{C_{x+1}^{t+1}+C_{x+1}^t}{2}-\frac{C_x^{t+1}+C_x^t}{2}\right)- \\ -\frac{\hat{D}_{xx(x-\frac{1}{2})}}{\Delta x^2}\left(\frac{C_x^{t+1}+C_x^t}{2}-\frac{C_{x-1}^{t+1}+C_{x-1}^t}{2}\right)\end{aligned} \tag{2.27}$$

To solve this equation explicitly, however, further manipulation is required. First, two new variables, U and V, are defined as approximations of C. Then Equation (2.27) can be rewritten in terms of U and V as follows:

$$\begin{aligned}\frac{1}{\Delta t}\left(\frac{U_x^{t+1}+V_x^{t+1}}{2}-\frac{U_x^t+V_x^t}{2}\right) = \\ \frac{\hat{D}_{xx(x+\frac{1}{2})}}{\Delta x^2}\left(\frac{V_{x+1}^{t+1}+U_{x+1}^t}{2}-\frac{V_x^{t+1}+U_x^t}{2}\right)- \\ -\frac{\hat{D}_{xx(x-\frac{1}{2})}}{\Delta x^2}\left(\frac{U_x^{t+1}+V_x^t}{2}-\frac{U_{x-1}^{t+1}+V_{x-1}^t}{2}\right)\end{aligned} \tag{2.28}$$

Note that half of all the U and V values are taken at the new time level, and that there are an equal number of U and V values at both time levels. This creates a particular pattern vital to Larkin's ADE method, which will be described in more detail shortly. Now an approximation of C can be defined as the average of both U and V, and the above equation can be split into two equations, the first containing only U terms and the second containing only V terms.

$$C_x^{t+1} = \frac{U_x^{t+1}+V_x^{t+1}}{2} \tag{2.29}$$

$$\frac{U_x^{t+1} - U_x^t}{\Delta t} = \frac{\hat{D}^t_{xx(x+\frac{1}{2})}}{\Delta x^2}\left(U_{x+1}^t - U_x^t\right) - \frac{\hat{D}^{t+1}_{xx(x-\frac{1}{2})}}{\Delta x^2}\left(U_x^{t+1} - U_{x-1}^{t+1}\right) \quad (2.30)$$

$$\frac{V_x^{t+1} - V_x^t}{\Delta t} = \frac{\hat{D}^{t+1}_{xx(x+\frac{1}{2})}}{\Delta x^2}\left(V_{x+1}^{t+1} - V_x^{t+1}\right) - \frac{\hat{D}^t_{xx(x-\frac{1}{2})}}{\Delta x^2}\left(V_x^t - V_{x-1}^t\right) \quad (2.31)$$

Note that these equations are complementary in the form of the spatial second derivative, and that the apparent dispersivity can now be associated with a difference at a particular time level, and thus can be assigned a time reference. It can now be seen that both U and V can be solved explicitly, if the nodes are solved in a particular order. Equation (2.30) must be solved in increasing index x, while Equation (2.31) must be solved in decreasing index x. Though the above example is shown in only one dimension, this pattern can be extended to three dimensions. By observing the patterns in Equations (2.30) and (2.31), the following rules can be developed for expressing the complementary U and V discretizations of the dispersion equation. First, spatial terms taken at the new time level in U must be taken at the old time level in V, and vice-versa. Secondly, there must be the same number of terms taken at each time level in both the U and V formulations. Finally, the resulting form of U must be able to be solved explicitly in the opposite direction from which V can be solved explicitly. Using these rules, the following complementary forms of the dispersion equation in three

dimensions can be written as:

$$\frac{\left(\theta_w^{t+1}+K_d\rho_b+\theta_g^{t+1}\frac{mw_wH}{RT}\right)*U_{x,y,z}^{t+1}-\left(\theta_w^{t}+K_d\rho_b+\theta_g^{t}\frac{mw_wH}{RT}\right)*U_{x,y,z}^{t}}{\Delta t}=$$

$$\frac{\hat{D}^t_{xx(x+\frac{1}{2},y,z)}}{\Delta x^2}(U^t_{x+1,y,z}-U^t_{x,y,z})-\frac{D^{t+1}_{xx_{x-\frac{1}{2}}}}{\Delta x^2}(U^{t+1}_{x,y,z}-U^{t+1}_{x-1,y,z})+$$

$$+\frac{\hat{D}^t_{xy(x+\frac{1}{2},y,z)}}{4\Delta x\Delta y}(U^t_{x+1,y+1,z}+U^t_{x,y+1,z})-\frac{\hat{D}^{t+1}_{xy(x+\frac{1}{2},y,z)}}{4\Delta x\Delta y}(U^{t+1}_{x+1,y-1,z}-U^{t+1}_{x,y-1,z})-$$

$$-\frac{\hat{D}^t_{xy(x-\frac{1}{2},y,z)}}{4\Delta x\Delta y}(U^t_{x,y+1,z}+U^t_{x-1,y+1,z})-\frac{\hat{D}^{t+1}_{xy(x-\frac{1}{2},y,z)}}{4\Delta x\Delta y}(U^{t+1}_{x,y-1,z}-U^{t+1}_{x-1,y-1,z})+$$

$$+\frac{\hat{D}^t_{xy(x+\frac{1}{2},y,z)}}{4\Delta x\Delta y}(U^t_{x+1,y+1,z}+U^t_{x,y+1,z})-\frac{\hat{D}^{t+1}_{xy(x+\frac{1}{2},y,z)}}{4\Delta x\Delta y}(U^{t+1}_{x+1,y-1,z}-U^{t+1}_{x,y-1,z})-$$

$$-\frac{\hat{D}^t_{xy(x-\frac{1}{2},y,z)}}{4\Delta x\Delta y}(U^t_{x,y+1,z}+U^t_{x-1,y+1,z})-\frac{\hat{D}^{t+1}_{xy(x-\frac{1}{2},y,z)}}{4\Delta x\Delta y}(U^{t+1}_{x,y-1,z}-U^{t+1}_{x-1,y-1,z})+$$

$$+\frac{\hat{D}^t_{yy(x,y+\frac{1}{2},z)}}{\Delta y^2}(U^t_{x,y+1,z}-U^t_{x,y,z})-\frac{D^{t+1}_{yy_{x,y-\frac{1}{2},z}}}{\Delta y^2}(U^{t+1}_{x,y,z}-U^{t+1}_{x,y-1,z})+$$

$$+\frac{\hat{D}^t_{yx(x,y+\frac{1}{2},z)}}{4\Delta y\Delta x}(U^t_{x+1,y+1,z}+U^t_{x+1,y,z}-U^t_{x-1,y+1,z})-\frac{\hat{D}^{t+1}_{yx(x,y+\frac{1}{2},z)}}{4\Delta y\Delta x}(U^{t+1}_{x-1,y,z})-$$

$$-\frac{\hat{D}^t_{yx(x,y-\frac{1}{2},z)}}{4\Delta y\Delta x}(U^t_{x+1,y,z})+\frac{\hat{D}^{t+1}_{yx(x,y-\frac{1}{2},z)}}{4\Delta y\Delta x}(U^{t+1}_{x+1,y-1,z}-U^{t+1}_{x-1,y,z}-U^{t+1}_{x-1,y-1,z})+$$

$$+\frac{\hat{D}^t_{yz(x,y+\frac{1}{2},z)}}{4\Delta y\Delta z}(U^t_{x,y+1,z+1}+U^t_{x,y,z+1})-\frac{\hat{D}^{t+1}_{yz(x,y+\frac{1}{2},z)}}{4\Delta y\Delta z}(U^{t+1}_{x,y+1,z-1}-U^{t+1}_{x,y,z-1})-$$

$$-\frac{\hat{D}^t_{yz(x,y-\frac{1}{2},z)}}{4\Delta y\Delta z}(U^t_{x,y,z+1}+U^t_{x,y-1,z+1})-\frac{\hat{D}^{t+1}_{yz(x,y-\frac{1}{2},z)}}{4\Delta y\Delta z}(U^{t+1}_{x,y,z-1}-U^{t+1}_{x,y-1,z-1})+$$

$$+\frac{\hat{D}^t_{zz(x,y,z+\frac{1}{2})}}{\Delta z^2}(U^t_{x,y,z+1}-U^t_{x,y,z})-\frac{D^{t+1}_{zz_{x,y,z-\frac{1}{2}}}}{\Delta z^2}(U^{t+1}_{x,y,z}-U^{t+1}_{x,y,z-1})+$$

$$+\frac{\hat{D}^t_{zx(x,y,z+\frac{1}{2})}}{4\Delta z\Delta x}(U^t_{x+1,y,z+1}+U^t_{x+1,y,z}-U^t_{x-1,y,z+1})-\frac{\hat{D}^{t+1}_{zx(x,y,z+\frac{1}{2})}}{4\Delta z\Delta x}(U^{t+1}_{x-1,y,z})-$$

$$-\frac{\hat{D}^t_{zx(x,y,z-\frac{1}{2})}}{4\Delta z\Delta x}(U^t_{x+1,y,z})+\frac{\hat{D}^{t+1}_{zx(x,y,z-\frac{1}{2})}}{4\Delta z\Delta x}(U^{t+1}_{x+1,y,z-1}-U^{t+1}_{x-1,y,z}-U^{t+1}_{x-1,y,z-1})+$$

$$+\frac{\hat{D}^t_{zy(x,y,z+\frac{1}{2})}}{4\Delta z\Delta y}(U^t_{x,y+1,z+1}+U^t_{x,y+1,z}-U^t_{x,y-1,z+1})-\frac{\hat{D}^{t+1}_{zy(x,y,z+\frac{1}{2})}}{4\Delta z\Delta y}(U^{t+1}_{x,y-1,z})-$$

$$-\frac{\hat{D}^t_{zy(x,y,z-\frac{1}{2})}}{4\Delta z\Delta y}(U^t_{x,y+1,z})+\frac{\hat{D}^{t+1}_{zy(x,y,z-\frac{1}{2})}}{4\Delta z\Delta y}(U^{t+1}_{x,y+1,z-1}-U^{t+1}_{x,y-1,z}-U^{t+1}_{x,y-1,z-1}) \tag{2.32}$$

and,

$$\frac{\left(\theta_w^{t+1}+K_d\rho_b+\theta_g^{t+1}\frac{mw_wH}{RT}\right)*V_{x,y,z}^{t+1}-\left(\theta_w^{t}+K_d\rho_b+\theta_g^{t}\frac{mw_wH}{RT}\right)*V_{x,y,z}^{t}}{\Delta t}=$$

$$\frac{\hat{D}^{t+1}_{xx(x+\frac{1}{2},y,z)}}{\Delta x^2}(V^{t+1}_{x+1,y,z}-V^{t+1}_{x,y,z})-\frac{D^{t}_{xx_{x-\frac{1}{2}}}}{\Delta x^2}(V^{t}_{x,y,z}-V^{t}_{x-1,y,z})+$$

$$+\frac{\hat{D}^{t+1}_{xy(x+\frac{1}{2},y,z)}}{4\Delta x\Delta y}(V^{t+1}_{x+1,y+1,z}+V^{t+1}_{x,y+1,z})-\frac{\hat{D}^{t}_{xy(x+\frac{1}{2},y,z)}}{4\Delta x\Delta y}(V^{t}_{x+1,y-1,z}-V^{t}_{x,y-1,z})-$$

$$-\frac{\hat{D}^{t+1}_{xy(x-\frac{1}{2},y,z)}}{4\Delta x\Delta y}(V^{t+1}_{x,y+1,z}+V^{t+1}_{x-1,y+1,z})-\frac{\hat{D}^{t}_{xy(x-\frac{1}{2},y,z)}}{4\Delta x\Delta y}(V^{t}_{x,y-1,z}-V^{t}_{x-1,y-1,z})+$$

$$+\frac{\hat{D}^{t+1}_{xy(x+\frac{1}{2},y,z)}}{4\Delta x\Delta y}(V^{t+1}_{x+1,y+1,z}+V^{t+1}_{x,y+1,z})-\frac{\hat{D}^{t}_{xy(x+\frac{1}{2},y,z)}}{4\Delta x\Delta y}(V^{t}_{x+1,y-1,z}-V^{t}_{x,y-1,z})-$$

$$-\frac{\hat{D}^{t+1}_{xy(x-\frac{1}{2},y,z)}}{4\Delta x\Delta y}(V^{t+1}_{x,y+1,z}+V^{t+1}_{x-1,y+1,z})-\frac{\hat{D}^{t}_{xy(x-\frac{1}{2},y,z)}}{4\Delta x\Delta y}(V^{t}_{x,y-1,z}-V^{t}_{x-1,y-1,z})+$$

$$+\frac{\hat{D^{t+1}}_{yy(x,y+\frac{1}{2},z)}}{\Delta y^2}(V^{t+1}_{x,y+1,z}-V^{t+1}_{x,y,z})-\frac{D^{t}_{yy_{x,y-\frac{1}{2},z}}}{\Delta y^2}(V^{t}_{x,y,z}-V^{t}_{x,y-1,z})+$$

$$+\frac{\hat{D}^{t+1}_{yx(x,y+\frac{1}{2},z)}}{4\Delta y\Delta x}(V^{t+1}_{x+1,y+1,z}+V^{t+1}_{x+1,y,z}-V^{t+1}_{x-1,y+1,z})-\frac{\hat{D}^{t}_{yx(x,y+\frac{1}{2},z)}}{4\Delta y\Delta x}(V^{t}_{x-1,y,z})-$$

$$-\frac{\hat{D}^{t+1}_{yx(x,y-\frac{1}{2},z)}}{4\Delta y\Delta x}(V^{t+1}_{x+1,y,z})+\frac{\hat{D}^{t}_{yx(x,y-\frac{1}{2},z)}}{4\Delta y\Delta x}(V^{t}_{x+1,y-1,z}-V^{t}_{x-1,y,z}-V^{t}_{x-1,y-1,z})+$$

$$+\frac{\hat{D}^{t+1}_{yz(x,y+\frac{1}{2},z)}}{4\Delta y\Delta z}(V^{t+1}_{x,y+1,z+1}+V^{t+1}_{x,y,z+1})-\frac{\hat{D}^{t}_{yz(x,y+\frac{1}{2},z)}}{4\Delta y\Delta z}(V^{t}_{x,y+1,z-1}-V^{t}_{x,y,z-1})-$$

$$-\frac{\hat{D}^{t+1}_{yz(x,y-\frac{1}{2},z)}}{4\Delta y\Delta z}(V^{t+1}_{x,y,z+1}+V^{t+1}_{x,y-1,z+1})-\frac{\hat{D}^{t}_{yz(x,y-\frac{1}{2},z)}}{4\Delta y\Delta z}(V^{t}_{x,y,z-1}-V^{t}_{x,y-1,z-1})+$$

$$+\frac{\hat{D}^{t+1}_{zz(x,y,z+\frac{1}{2})}}{\Delta z^2}(V^{t+1}_{x,y,z+1}-V^{t+1}_{x,y,z})-\frac{D^{t}_{zz_{x,y,z-\frac{1}{2}}}}{\Delta z^2}(V^{t}_{x,y,z}-V^{t}_{x,y,z-1})+$$

$$+\frac{\hat{D}^{t+1}_{zx(x,y,z+\frac{1}{2})}}{4\Delta z\Delta x}(V^{t+1}_{x+1,y,z+1}+V^{t+1}_{x+1,y,z}-V^{t+1}_{x-1,y,z+1})-\frac{\hat{D}^{t}_{zx(x,y,z+\frac{1}{2})}}{4\Delta z\Delta x}(V^{t}_{x-1,y,z})-$$

$$-\frac{\hat{D}^{t+1}_{zx(x,y,z-\frac{1}{2})}}{4\Delta z\Delta x}(V^{t+1}_{x+1,y,z})+\frac{\hat{D}^{t}_{zx(x,y,z-\frac{1}{2})}}{4\Delta z\Delta x}(V^{t}_{x+1,y,z-1}-V^{t}_{x-1,y,z}-V^{t}_{x-1,y,z-1})+$$

$$+\frac{\hat{D}^{t+1}_{zy(x,y,z+\frac{1}{2})}}{4\Delta z\Delta y}(V^{t+1}_{x,y+1,z+1}+V^{t+1}_{x,y+1,z}-V^{t+1}_{x,y-1,z+1})-\frac{\hat{D}^{t}_{zy(x,y,z+\frac{1}{2})}}{4\Delta z\Delta y}(V^{t}_{x,y-1,z})-$$

$$-\frac{\hat{D}^{t+1}_{zy(x,y,z-\frac{1}{2})}}{4\Delta z\Delta y}(V^{t+1}_{x,y+1,z})+\frac{\hat{D}^{t}_{zy(x,y,z-\frac{1}{2})}}{4\Delta z\Delta y}(V^{t}_{x,y+1,z-1}-V^{t}_{x,y-1,z}-V^{t}_{x,y-1,z-1}) \tag{2.33}$$

By averaging these complementary discretizations, the resulting equation

for the concentration at the new time level includes information from all nodes at both the old and new time levels and can be solved explicitly.

In addition to including more temporal information at each time step than a fully explicit method, the mixture of implicit and explicit terms in the U and V discretizations results in increased stability of the overall method. The report by Morita and Yen, as well as the original publication of this form of the ADE method by Larkin in 1969, suggests that the stability of this method is unconditional. Larkin cites a Von Neuman analysis as the basis for this statement. However, Larkin's ADE method is based on an older method by Saul'ev, and Larkin notes that in his 1958 paper, Saul'ev writes that the ADE method is stable as long as a stability parameter of the form $\omega = (\Delta x)^2/\Delta t$ is greater or equal to one-half. Because of this discrepancy in the literature, a Von Neuman stability analysis was performed during the course of this research. The result indicated that the method is unconditionally stable. However, Von Neuman analyses are only valid away from the boundaries - an important distinction, because during the research, it was found that the method of implementing the boundary conditions greatly affected the stability of the model.

As mentioned previously, the dispersion solver enforces a no-flux boundary around the entire system. The most obvious way to implement this boundary condition is to define the flux in the direction x through the boundary face $x \pm \frac{1}{2}$ as follows:

$$F_x = \frac{\hat{D}}{\Delta x^2}(C_x - C_{x\pm1}) \tag{2.34}$$

Therefore, it can be seen that if there is no flux across the boundary between cells $x-1$ and x, then the difference $(C_x^{t+1} - C_{x-1}^{t+1})$ must be equal to zero. This information can be directly incorporated into the discretized form of the dispersion equation by setting this difference in the spatial second derivative to zero. This results in an abbreviated form of the discretization. For example, using Equations (2.29), (2.30), and(2.31) if the boundary node $x = 1$ is being evaluated, the abbreviated form of the discretizations would be as follows:

$$\frac{U_x^{t+1} - U_x^t}{\Delta t} = \frac{\hat{D}^t_{x+\frac{1}{2}}}{\Delta x^2}\left(U_{x+1}^t - U_x^t\right) - \frac{\hat{D}^{t+1}_{x-\frac{1}{2}}}{\Delta x^2}\left(0\right) \tag{2.35}$$

$$\frac{V_x^{t+1} - V_x^t}{\Delta t} = \frac{\hat{D}^{t+1}_{x+\frac{1}{2}}}{\Delta x^2}\left(V_{x+1}^{t+1} - V_x^{t+1}\right) - \frac{\hat{D}^{t}_{x-\frac{1}{2}}}{\Delta x^2}\left(0\right) \tag{2.36}$$

However, it can be seen that the U formulation is now fully explicit, while the V formulation is fully implicit. Therefore, the stability of the U formulation, and thus the resulting formula for C^{t+1}, is restricted to discretizations where $\mathcal{D}$ is less than one-half. Note that if the boundary considered had been between nodes x and $x+1$, then the V formulation would be fully explicit, and the U form would be fully implicit. Thus, the ADE method does not offer any advantages over other, less computationally intensive explicit methods. However, if the boundary conditions are implemented in such a way as not to remove the mixture of implicit and explicit terms, the method appears to be more stable. During the course of this research, two different methods were derived to meet this criterion.

The first method models the zero-flux boundary condition by assigning concentrations at ghost-nodes outside of the model domain, which are equal to the value of their neighbors inside the domain. Recall that the boundary condition requires $(C_x^{t+1} - C_{x-1}^{t+1})$ to equal zero. If node x is located at a boundary, then this equation can be rewritten as $(C_b^{t+1} - C_g^{t+1}) = 0$ and the concentration at the ghost-node can be described as $C_b^{t+1} = C_g^{t+1}$, where the subscripts b and g stand for boundary and ghost, respectively. However, the concentration at the boundary point at the new time level is not known, but rather is being determined during the current time step. If the assumption is made that the concentration does not change significantly from one time level to the next, then the ghost-node concentration can be approximated as C_x^t. Now the U discretization of the ADE method at the boundary can be expressed as a fixed concentration (first-type) boundary condition as follows:

$$\frac{U_b^{t+1} - U_b^t}{\Delta t} = \frac{\hat{D}^t_{xx(b+\frac{1}{2})}}{\Delta x^2}\left(U_{b+1}^t - U_b^t\right) - \frac{\hat{D}^{t+1}_{xx(b-\frac{1}{2})}}{\Delta x^2}\left(U_b^{t+1} - U_b^t\right) \tag{2.37}$$

This equation can be rewritten as:

$$U_b^{t+1}\left(\frac{1}{\Delta t} + \frac{\hat{D}^{t+1}_{xx(b-\frac{1}{2})}}{\Delta x^2}\right) = \frac{U_b^t}{\Delta t} + \frac{\hat{D}^t_{xx(b+\frac{1}{2})}}{\Delta x^2}U_{b+1}^t - \frac{\hat{D}^t_{xx(b+\frac{1}{2})}}{\Delta x^2}U_b^t + \frac{\hat{D}^{t+1}_{xx(b-\frac{1}{2})}}{\Delta x^2}U_b^t \tag{2.38}$$

If it can be assumed that $\hat{D}^t_{xx(b+\frac{1}{2})} \approx \hat{D}^{t+1}_{xx(b-\frac{1}{2})}$, then the last two terms on the right side of the equation cancel each other, leaving:

$$U_b^{t+1}\left(\frac{1}{\Delta t} + \frac{\hat{D}^{t+1}_{xx(b-\frac{1}{2})}}{\Delta x^2}\right) = \frac{U_b^t}{\Delta t} + \frac{\hat{D}^t_{xx(b+\frac{1}{2})}}{\Delta x^2}U_{b+1}^t \tag{2.39}$$

This equation has the same form as Equation (2.35), except that it solves for U_b^{t+1} implicitly. Therefore, by assigning ghost nodes around the model's physical boundaries which are equal to the old time-step concentration at the boundary node, the Equations (2.29), (2.32), and (2.33) can be used without modification to solve the dispersion equation. In order to solve the cross-terms, it will clearly be necessary to define some ghost-nodes that are not adjacent to boundary-nodes, but instead are adjacent to other ghost nodes. These nodes occur at the corners and edges of the model domain. To approximate the concentration at these points, a linear average of all the adjacent nodes is used.

The second method does not require the definition of ghost nodes, and thus requires less memory storage than the previously described method. In this method, whichever discretization (U or V) becomes fully explicit is changed to a fully implicit equation similar in form to Equation (2.39), using the assumption that the concentration at the boundary does not change much with time, and that $\hat{D}^{t+1}_{xx(b\pm\frac{1}{2})} \approx \hat{D}^{t}_{xx(b\pm\frac{1}{2})}$. For example, Equation (2.35) is rewritten in the following way:

$$\frac{U_b^{t+1} - U_b^t}{\Delta t} = \frac{\hat{D}^t_{b+\frac{1}{2}}}{\Delta x^2}\left(U^t_{b+1}\right) - \frac{\hat{D}^{t+1}_{b+\frac{1}{2}}}{\Delta x^2}\left(U_b^{t+1}\right) \tag{2.40}$$

This manipulation removes the ability to solve for U_b^{t+1} explicitly at the boundary, increasing the stability of the U formulation, and thus, the method as a whole. Unfortunately, this method of estimating the boundary conditions does not present a computationally efficient method of calculating the cross-terms. Therefore, in the interest of accelerating the computation, the cross-terms are neglected at the boundaries.

Both of the above methods for approximating the boundary conditions, which remove the ability to solve either the U or the V discretization explicitly, appear to be unconditionally stable, though the second method is preferable to the first because it requires neither the allocation of the ghost-node shell around the model's physical domain, nor the calculations necessary to initialize the ghost nodes. For these reasons, the second method for approximating the boundary conditions was implemented in the OSERTM, despite this method's omission of the cross-terms at the boundary nodes.

2.6 Reaction

The reaction equation describes the transformation of the chemical species being tracked as a function of time. In this research, the reactions of interest are: nitrification, the biological oxidation of the ammonium ion to nitrate; denitrification, the biological reduction of nitrate to gaseous nitrogen; and aerobic degradation of soil organic matter (SOM). The chemical equations that describe these three biologically mediated reactions are:

$$NH_4^+ + 2O_{2(aq)} \longrightarrow \quad NO_3^{-2} + 2H^+ + H_2O \tag{2.41}$$

$$CH_2O + \frac{4}{5}NO_3^{-2} \longrightarrow \quad \frac{2}{5}N_2 + \frac{4}{5}HCO_3^- + \frac{1}{5}CO_{2(aq)} + \frac{3}{5}H_2O \tag{2.42}$$

$$CH_2O + O_2 \longrightarrow \quad CO_2 + H_2O \tag{2.43}$$

where nitrification is represented by Equation (2.41), denitrification is represented by Equation (2.42), and the degredation of SOM is represented by Equation (2.43). Soil organic matter is assumed to be represented by CH_2O (*MacQuarrie and Sudicky*, 2001)

These reactions are the result of complex physical and biological processes, and can be modeled a number of different ways. As noted in Section 1.2, these processes are usually described with a kinetic model, where the rate constant is dependent on a combination of the following factors: the population of the responsible microorganisms, the concentration of the reacting species, and the concentration of any reaction inhibitors. While nitrification and SOM consumption are usually modeled as second- and first-order reactions, respectively, there is some variation in the representation of the denitrification reaction. The most likely reason for this variation is that little information is known about the processes involved in denitrification (*Kinzelbach et al.*, 1990). For example, many models assume that denitrification will never occur if the dissolved oxygen concentration is above 1 mg/L, yet in a review of the state of knowledge of denitrification, Korom (1992) states that this oxygen limitation is too strict, as there are bacteria known to denitrify at much higher dissolved oxygen levels.

Multiple Monod-expressions were chosen to calculate the reaction rates of nitrification, denitrification, and consumption of SOM, because of their expressive abilities and successful application to the modeling of microbiologically mediated reactions in the sub-surface (*MacQuarrie et al.*, 1990; *Kinzelbach et al.*, 1991). The general form of a multiple-Monod expression for the reaction rate is shown in Equation (2.44) (*Chen et al.*, 1992; *Essaid*

et al., 1995; *MacQuarrie and Sudicky*, 2001):

$$R_a = k^a_{max} X_m F(X_m) \left[\frac{C_1}{K^a_1 + C_1}\right] \left[\frac{C_1}{K^a_1 + C_1}\right] \cdots \left[\frac{C_n}{K^a_n + C_n}\right] F(C_I) \quad (2.44)$$

where:

R_a	is the rate of reaction a
k^a_{max}	is the maximum primary substrate utilization rate for reaction a [M substrate/M biomass/T]
X_m	is the biomass of population m responsible for reaction a [M biomass/L^3]
$F(X_m)$	is a coefficient representing biomass inhibition
$C_1 \rightarrow C_n$	are the aqueous concentrations of the reacting species [mg/L^3 water]
$K^a_1 \rightarrow K^a_n$	are the half-saturation constants for the respective species [M species L^3 water]
$F(C_I)$	is a coefficient representing noncompetitive inhibition

The multiple-Monod expression shown above defines the rate of reaction a as the product of the maximum substrate utilization rate and several functions, called limiting terms, which describe the effects of microbial population dynamics, substrate availability, and non-competitive inhibition. In this research, however, the term representing the effect of the microbial population will not be considered for reasons discussed shortly. Each of the limiting terms vary continuously between zero and one, and describe the fraction of the maximum substrate utilization rate that can occur under the specified conditions for the control variable described. For example, if the ammonium ion concentration is close to zero, the term describing the availability of the ammonium ion in the multiple-Monod representation of the rate of nitrification will have a value of close to zero. On the other hand, if the ammonium ion concentration is large, then the limiting term will have a value close to one. Thus, it can be seen that the product of all the limiting terms effectively describes the fraction of the maximum substrate utilization rate that can occur, considering the state of the control variables. The form of the substrate availability limiting term used in this work is generally accepted, but the form for the non-competitive inhibition limiting term is subject to some interpretation, based on the nature of the inhibition. For this work, the only non-competitive inhibition considered was that of dissolved oxygen on denitrification. MacQuarrie and Sudicky (2001) suggest a hyperbolic function for describing the non-competitive inhibition of oxygen

on denitrification, which takes the following form:

$$F(C_{I(O_2)}) = \left[\frac{K_{I(O_2)}}{K_{I(O_2)} + C_{O_2}}\right] \tag{2.45}$$

This function, however, transitions from zero to one very slowly; thus, much denitrification can take place at high dissolved oxygen concentrations. Though it is not unusual for some denitrification to occur under oxic conditions, it seemed as if this form of the limiting function allowed too much to occur. For this reason, the following function for oxygen inhibition of denitrification, presented by (*Hudson*, 2001b), was used:

$$F(C_I) = 2 * \frac{e^{-k_I * C_{O_2}}}{1 + e^{-k_I * C_{O_2}}} \tag{2.46}$$

where k_I is a fitting parameter.

To apply this multiple-Monod model to the processes of nitrification and denitrification, however, the OSERTM would have to keep track of four different reacting species and two populations of microorganisms. For this reason, a couple of simplifying assumptions were made. The first assumption is that natural organic matter is present in excess in the soil, and thus does not inhibit denitrification. This effectively reduces the number of modeled species to three. The second assumption is that both nitrifiers and denitrifiers are present in such significant quantities that neither change with time, nor affect the reaction kinetics. This effectively removes the microbial population terms from Equation (2.44). These simplifications result in the following multiple-Monod expressions for the reaction rates of nitrification, denitrification, and organic matter oxidation (adapted from *MacQuarrie and Sudicky*, 2001):

$$R_{nit} = k_{nit}\left[\frac{C_{NH_4^+}}{K_{NH_4^+} + C_{NH_4^+}}\right]\left[\frac{C_{O_2}}{K_{O_2} + C_{O_2}}\right] \tag{2.47}$$

$$R_{denit} = k_{denit}\left[\frac{C_{NO_3^{-2}}}{K_{NO_3^{-2}} + C_{NO_3^{-2}}}\right]\left[2 * \frac{e^{-k_I * C_{O_2}}}{1 + e^{-k_I * C_{O_2}}}\right] \tag{2.48}$$

$$R_{ox} = k_{ox}\left[\frac{C_{O_2}}{K_{O_2} + C_{O_2}}\right] \tag{2.49}$$

These reactions can now be modeled by incorporating their equations into

Equation (2.9) and forming the following system of ODEs:

$$\begin{aligned}
\frac{\partial\left(\theta_w + K_d * \rho_b + \theta_g * \frac{mw_w H}{RT}\right) C_{(aq)}^{NH_4^+}}{\partial t} &= -mw_{NH_4^+} R_{nit} \\
\frac{\partial\left(\theta_w + K_d * \rho_b + \theta_g * \frac{mw_w H}{RT}\right) C_{(aq)}^{NO_3^{2-}}}{\partial t} &= mw_{NO_3^{-2}} \left(R_{nit} - \frac{4}{5} R_{denit}\right) \\
\frac{\partial\left(\theta_w + K_d * \rho_b + \theta_g * \frac{mw_w H}{RT}\right) C_{(aq)}^{O_2}}{\partial t} &= mw_{O_2} \left(-R_{ox} - 2R_{nit}\right) \quad (2.50)
\end{aligned}$$

Two different approaches were investigated to solve this system of equations. The first approach used a fourth-order Runge-Kutta method to solve Equation (2.50). However, due to the nonlinearity of these equations, it was found that this method may require shorter time steps than either the advection or dispersion equations, thus rendering it an unattractive option. The second approach approximated each reaction as a first-order kinetic process. Therefore, the system of ODEs shown in Equation (2.50) can be rewritten as the following independent equations:

$$\frac{\partial C_{(aq)}^{NH_4^+}}{\partial t} = \frac{\lambda_{nit} C_{NH_4^+}}{\left(\theta_w + K_d * \rho_b + \theta_g * \frac{mw_w H}{RT}\right)} \quad (2.51)$$

$$\frac{\partial C_{(aq)}^{NO_3^{2-}}}{\partial t} = \frac{\lambda_{denit} C_{NO_3^{-2}}}{\left(\theta_w + K_d * \rho_b + \theta_g * \frac{mw_w H}{RT}\right)} + \Upsilon_{nit} \quad (2.52)$$

$$\frac{\partial C_{(aq)}^{O_2}}{\partial t} = \frac{\lambda_{ox} C_{O_2}}{\left(\theta_w + K_d * \rho_b + \theta_g * \frac{mw_w H}{RT}\right)} \quad (2.53)$$

where λ and C are constants expressed as:

$$\begin{aligned}
\lambda_{nit} &= -k_{nit} \left[\frac{1}{K_{NH_4^+} + C_{NH_4^+}}\right] \left[\frac{C_{O_2}}{K_{O_2} + C_{O_2}}\right] \\
\lambda_{denit} &= -k_{denit} \frac{4}{5} \left[\frac{1}{K_{NO_3^{-2}} + C_{NO_3^{-2}}}\right] \left[2 * \frac{e^{-k_I * C_{O_2}}}{1 + e^{-k_I * C_{O_2}}}\right] \\
\lambda_{ox} &= -k_{ox} \left[\frac{1}{K_{O_2} + C_{O_2}}\right] - 2k_{nit} \left[\frac{C_{NH_4^+}}{K_{NH_4^+} + C_{NH_4^+}}\right] \left[\frac{1}{K_{O_2} + C_{O_2}}\right] \\
\Upsilon_{nit} &= \frac{R_{nit}}{\left(\theta_w + K_d * \rho_b + \theta_g * \frac{mw_w H}{RT}\right)}
\end{aligned}$$

where the concentrations are taken at the old time level. Equations (2.51), (2.52), and (2.53) can now be solved by integration, resulting in the following

equations:

$$C^{NH_4^+} = C_{t-1}^{NH_4^+} * e^{mw_{NH_4^+}\lambda_{nit}\Delta t} \quad (2.54)$$

$$C^{NO_3^{-2}} = mw_{NO_3^{-2}}\Upsilon_{nit}\Delta t + C_{t-1}^{NO_3^{-2}} * e^{mw_{NO_3^{-2}}\lambda_{denit}\Delta t} \quad (2.55)$$

$$C^{O_2} = C_{t-1}^{O_2} * e^{mw_{O_2}\lambda_{nit}\Delta t} \quad (2.56)$$

These three equations can be solved independently. Thus the reaction system is no longer being solved simultaneously, but as will be shown in Section 3.5, these equations still give a good approximation to the true system of ODEs shown in Equation (2.50). Because this method's accuracy is less dependent on time step size, it is preferred over the fourth-order Runge-Kutta method.

2.7 Summary

In this chapter, finite difference modeling of the ADRE was reviewed. The OS paradigm was presented as a method to simplify the numerical solution to the ADRE by separating the advection, dispersion, and reaction subproblems. Finally, unique numerical methods for solving these subproblems were presented, along with information about their stability. The ULTIMATE TVD method was chosen for the advection subproblem, because of its resistance to numerical dispersion and oscillation. The ADE method was chosen for the dispersion subproblem, because of its stable nature. A pseudo-analytical approximation to the reaction system was chosen, because of its independence of time step size. In the next chapter, the implementation of the OSERTM will be described, and the performance of the model will be compared to analytical solutions for some classic test problems.

Chapter 3

Program Description and Verification

3.1 Model Overview

This chapter will describe how the numerical methods described in Chapter 2 are joined together to create the OSERTM, as well as give a brief description of the required input data and the output data. Then the performance of the model on a number of test problems will be described. Before beginning, however, it is instructive to describe, in generic terms, the physical features of an agricultural field to which this model may be applied. This description will help the reader to understand naming conventions used in the input files described in Section 3.2, as well as the usefulness of certain output files described in Section 3.3.

A schematic of an idealized field is depicted in Figure 3.1. This field is adjacent to a drainage ditch and contains two tile lines. The drainage ditch defines the right side boundary of the field and runs the length of that side. In order to identify each tile line in the model domain, the tiles are each assigned a unique tile number. Tile 1 runs perpendicular to the ditch and intersects tile 2, which runs parallel to the ditch. In order to apply the OSERTM to this field, the physical domain would be discretized according to the naming and directional conventions shown in Figure 2.1. It is conceivable that a tile line may span multiple grid cells. In this case it is necessary to divide the tile line into segments, so that each segment exists in only one cell. Each tile segment is then given a segment number beginning with 1. For example, tile 1 in Figure 3.1, may be divided into segments, as shown in Figure 3.2. As a result of these naming conventions, each tile segment in a field can be uniquely identified by the combination of its tile number and segment number.

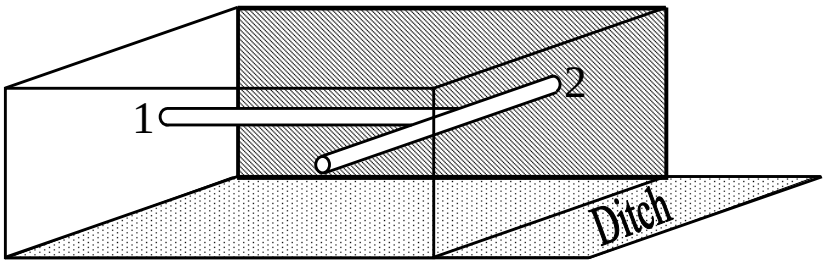

Figure 3.1: Schematic of a generic, idealized agricultural field indicating drain numbering conventions.

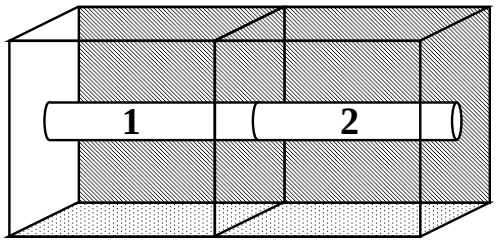

Figure 3.2: Schematic of the tile-segment numbering conventions used in the OSERTM.

The OSERTM code was written in Fortran 90 and was designed with modularity as a key feature, so that the model would be able to support many different numerical solvers for the advection, dispersion, and reaction subproblems described in Chapter 2. For this reason, all calculations are performed in subroutines called from the program file, drainmod3d.f90, which will be referred to as `Main`. The `Main` program file allocates and initializes all of the program parameters and global variables, invokes subroutines to read the temporal hydrologic data and solve the three subproblems, and writes the output files. A flow chart representation of the `Main` program can be seen in Figure 3.3, and will serve to clarify the program description that will be given in the following three sections.

3.2 Input Files

Before describing the steps taken by the OSERTM to solve a solute transport problem, it is instructive to first describe the input files and required data. Formats for all the files described here can be found in Appendix A. The first input file to be discussed is a namelist file, referred to as `PARAM`, which contains values for the parameters shown in Table 3.1. A namelist is a Fortran 90 construct that contains variable assignment statements, such as `alpha_L = 0.5`. The program reads these assignment statements and performs the variable assignments implied. Because this type of file is not read in the traditional way, the placement of the value within the row is unimportant, but it is important that the variable names are not changed, that at the end of each row there is a comma, and that the end of the list of assignments is signified by a backslash character. While this type of input is not traditional, it has the advantage that the user can explicitly see which variable is being assigned what value.

The name of the next input file to be accessed is specified in the `PARAM` namelist file as `FILESIN`. This input file contains the filename and path, relative to the home directory of the program, for all the input and output files. The I/O files can have any name, as long as the name and path of each file is less than or equal to 40 characters. This file contains names for what will be referred to as the `CHEM`, `ICOND`, `PATCH`, `SOIL`, `SPARAM`, `MOISTURE`, `SPDIS`, `DRAIN`, `VEC`, `CDRAIN`, `MDITCH`, and `PROFILE` files, in this particular order. The `CHEM` file contains an identifying code, the soil-water partition coefficient, and the coefficient of gas phase diffusion for each chemical, as well as the temperature of the subsurface environment. The first chemical in the list is assigned an identification code of 1, the second chemical in the list is

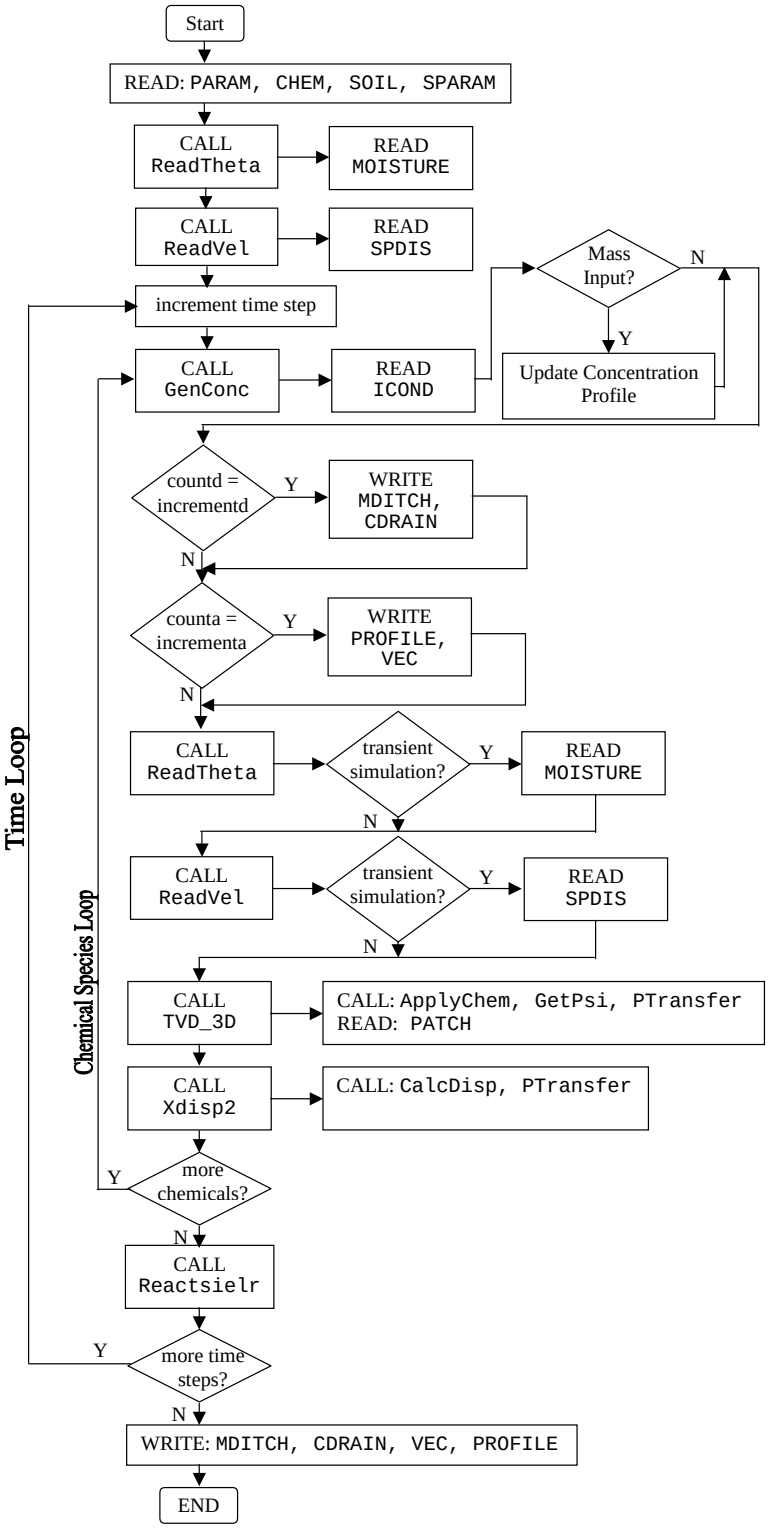

Figure 3.3: Flowchart for the `Main` program of the OSERTM code.

Table 3.1: Parameters specified in PARAM namelist.

`alpha_L`	longitudinal dispersivity
`alpha_TH`	transverse horizontal dispersivity
`alpha_TV`	transverse vertical dispersivity
`chem`	number of chemicals
`choice`	order of TVD method
`Dstar`	coefficient of aqueous molecular diffusion
`delta_i`	discretization length in y-direction
`delta_j`	discretization length in x-direction
`delta_k`	discretization length in z-direction
`delta_t`	temporal discretization size
`filesin`	file containing I/O file names
`incrementa`	output interval for concentration profile
`incrementd`	output interval for drain information
`m`	number of cells in y-direction
`n`	number of cells in x-direction
`o`	number of cells in z-direction
`stoptime`	length of simulation
`trans`	1 for transient simulations, otherwise 0

assigned a code of 2, and so on. This code is used to identify which chemical a specific set of input or output data are associated with. Any chemical except oxygen can be assigned any code; oxygen must be assigned the code 2. This file is read using a specified format for each line; therefore, it is important to follow this format. This is true for all of the following input files, as well. The format of each input file will be described in Appendix A. The `ICOND` input file contains the distribution of chemical within the model domain at the outset of the program. This is done by specifying the time step of the mass input, which, for the initial conditions, is equal to zero, and the x-, y-, and z-bounds of any contiguous sections that have a common concentration of the specified chemical. If a value greater than zero is input into the time step field, the model will continue to assign the specified concentration to those nodes until the model has completed the number of time steps indicated in the time step field. This feature is useful for approximating constant concentration sources within the model domain. A row of zeros marks the end of the file. If this file contains no information, the initial conditions will default to zero-concentration everywhere for each chemical except oxygen. The model assumes that the dissolved oxygen concentration in pore fluid in the unsaturated zone is initially at equilibrium with the atmosphere, while the pore fluid in the saturated zone has a dissolved oxygen concentration of

zero mg/L.

The `PATCH` file defines the mass flux into the model domain. As mentioned in Section 2.4, this mass flux is specified by assigning a ghost-node concentration according to Equation (2.13). The program assumes that mass enters only through the soil surface, so the user need only specify the x- and y-bounds of any contiguous regions that have a common ghost-node concentration of the specified chemical. Also, since the mass flux may begin or end at any time, or may change with time, the time step values when the flux begins and ends are specified in the second and third columns of the input file. Other than those differences, the format of this file is much the same as the `ICOND` file described above.

The files `SOIL` and `SPARAM` are used for both the hydrologic model, discussed in Section 4.1, and the OSERTM. The file `SOIL` uses the same region specifying technique as the `ICOND` file to define regions within the model domain that have a common soil type. This file specifies an integer code, corresponding to a soil type, for each node within the model domain. Once again, the end of the file is specified by a row of zeros. If no information is provided in the file, then each of the nodes is given a value of one. Therefore, if there is one soil type that is more pervasive than the others, it can be specified as type 1, and then this file will only have to specify where the other soil types are. This can save considerable time in writing the file. Also, if two soil types are assigned to a single node, the soil type that was specified last will have precedence. The soil parameters (porosity and bulk density) corresponding to the soil types represented by each integer code are specified in the `SPARAM` file.

The files `MOISTURE`, `SPDIS`, and `DRAIN` are output files of the hydrologic model. The `MOISTURE` file contains a single column containing the moisture content at each node written by increasing cell index j first, then i, and finally k for each time step (refer to Figure 2.1). The `SPDIS` file contains the directional components of the specific discharge for each of the cell faces for each time level. This information is written as a column with the data for the component of the specific discharge in the x-direction first, followed by that for the y-direction, and finally the z-direction. Each of the directional specific discharges are written out by increasing cell index j first, then i, and finally k. For each drain segment in the model domain at all time steps, the `DRAIN` file contains a code identifying the tile and segment numbers, the tripartite cell location index, and the volumetric flow rate of groundwater from the model domain into the drain segment (refer to Figures 3.1 and 3.2). If the simulation has transient flow conditions, which would be indicated by

the parameter assignment `trans = 1` in the `PARAM` file, the `MOISTURE`, `SPDIS`, and `DRAIN` files will contain many time steps of information ordered from shortest elapsed time to longest elapsed time; however, if the simulation does not have transient flow conditions, these files need only contain one temporal set of data.

3.3 Output Files

As mentioned in Section 3.2, the output files of the OSERTM are `VEC`, `CDRAIN`, `MDITCH`, and `PROFILE`. The formats for each of the files described here can be found in Appendix B. The `VEC` file writes out the directional components of the specific discharge located at each node in the following three columns: x-directional component, y-directional component, and z-directional component. This data is not created by the OSERTM, but because GMS (a product of Environmental Systems Incorporated) was used for visualization, and because GMS uses a different directional convention for the x and z directions, this output file is needed to convert the data in `SPDIS` into a GMS input file. In order to do this, the OSERTM reverses the sign on the x- and z-directional components of the specific discharge from the signs found in `SPDIS`. Also, it may be noted that the specific discharge is input at the cell faces and output at the node. This is because GMS requires the specific discharge to be specified at the node, not the cell faces. Linear interpolation is used to estimate the directional components using the data at either edge of the cell. The `CDRAIN` file outputs the mass flux into the specified drain segment at a specified time. The first column of this file contains the time the data was taken, and the remaining columns contain the value of the mass flux of each chemical species. The `MDITCH` file contains the mass flux of chemicals through a specified boundary of the model domain (for example into the ditch shown in Figure 3.1) using a similar format as the `CDRAIN` file. To determine this value each time the data is sampled, the mass flux for each chemical is summed over the entire ditch wall. Finally, the `PROFILE` file is actually a set of files, one for each chemical species. These files contain the concentration profiles of the associated chemicals at specified times. The concentration data is output as a column by first increasing cell index j, followed by i, and finally k. The frequency of data output to the `CDRAIN` and `MDITCH` files is controlled by the program parameter `incrementd`, which indicates the number of time steps between data points. For example, if `incrementd = 10` then data would be output every tenth time step. In a similar way, the parameter `incrementa` controls

the frequency of data output to the VEC and PROFILE files.

3.4 Program Function

Now that the form and content of the input and output files has been discussed, the method in which the Main file is used to coordinate the OSERTM can be described. Figure 3.3graphically depicts the following discussion. The first order of business of the Main program file is to declare all parameters and global variables and arrays. Then the user is prompted to input the file name of the PARAM namelist. This file name should include the path, relative to the main program directory, and should be no longer than 40 characters. The program then opens the namelist and performs the parameter value assignments specified therein. The time and space discretization information provided by the namelist is then used to determine the dimensions of the global arrays, such as the concentration field, specific discharge field, etc., and allocate dynamic memory to hold these arrays. Dynamic memory allocation allows the program to assign exactly the amount of memory needed for a particular problem, eliminating the need for the programmer to prespecify the size of the arrays, a process known as static memory allocation. It is advantageous to avoid static memory allocation because it often leads to inefficient memory use, or renders the program unable to solve very large problems. If insufficient memory is available for the allocation process, the computer will write an error of the form "<program filename>: WARNING - <variable name> not allocated" to the screen, and the program will crash. If the memory allocation is successful, however, the arrays will be initialized to zero, and the program will continue.

Next, the relative path and file names of the input and output files are read from FILES and stored in a character string array. Now the program knows where to find all of the necessary input information. If there is an error in one of the file names in FILES, the program will encounter a runtime error when it attempts to open the erroneous file, and this will cause the program to crash. The first of the files specified in FILES to be opened and read is CHEM. The temperature specified at the beginning of this file is used to calculate the density of water and the air-water partition coefficient for each chemical species using the relationship:

$$\rho_{H_2O} = 999.939900 + 4.216485 * 10^{-2} * (T) - 7.097451 * 10^{-3} * (T^2) + \\ + 3.509571 * 10^{-5} * (T^3) - 9.9037785 * 10^{-8} * (T^4) \quad (3.1)$$

and Equation (2.2). After the chemical properties have been input, the geological properties are read from the input files `SOIL` and `SPARAM`. The information in `SOIL` is used to fill an array the same size as the discretized spatial domain with integer values, which correspond to soil types defined by parameters specified in `SPARAM`.

Next, the `Main` program opens `MOISTURE`, `VEC`, and (if the problem contains tile drains) `DRAIN` and sets the reading frame to the first data value in each of the files. The subroutines `ReadTheta` and `ReadVel` are then called to input the moisture content and specific discharge fields, respectively. These functions are designed not only to input the initial conditions, but also to continue to update the hydrologic data for each time step during a transient simulation. Because the ADE method requires hydrologic data at both the old and new time levels, the OSERTM needs to store the information for both time levels. This is done using a four-dimensional array for each type of hydrologic data. The first three dimensions are spatial and correspond to the x-, y-, and z-dimensions of the problem domain, while the fourth dimension is temporal and can have either the value 1 (for old time level (t)) or 2 (for new time level (t+1)). Despite the necessity of two time levels of information, the functions `ReadTheta` and `ReadVel` only read in one time level of information each time they are called. For this reason, the first action these subroutines take is to overwrite the information at time level t with the information at time level t+1. Then the data at the new time level are read into the t+1 location of the array from the proper input file. This process is unimportant the first time these functions are called (when setting up the initial conditions), but it becomes important during the time step calculations, especially if the problem has transient hydraulic conditions, as will be seen momentarily.

Once the hydraulic information at the initial time step is read into memory, the output files specified in `FILES` are created. If the output file requires a header, it is written at this time. Finally, the last step in initializing the problem is to set the initial dissolved oxygen concentration to 8.52 mg/L in the unsaturated zone and 0.0 mg/L in the saturated zone.

Once the problem has been initialized, the OSERTM begins to step through time, solving for the concentration profile at each intermediate time level until the specified stop time has been reached. This time stepping is performed by a time loop. This loop begins with a call to the function `GenConc`, which uses the information from `ICOND` to set up the conditions at the old time level. Though the most common use of this function is to input the initial conditions, because it can also be used to introduce mass within

the model domain at any time during the simulation, it is located within the time loop. The most obvious advantage to this structure is that it enables the approximation of a constant concentration boundary condition somewhere within the model domain. This function call is followed by output statements that will output data to the output files described in Section 3.3. These output statements are controlled by two counters, which are initialized to zero at the start of the program. The counter `countd` controls the frequency of output to the `CDRAIN` and `MDITCH` files, while the other counter `counta` controls the frequency of output to the `VEC` and `PROFILE` files. When `countd` equals zero, data is written to the associated output files, and the counter is incremented by `incrementd`. Similarly, when `counta` equals zero, data is written to the associated files, and the counter is incremented by `incrementa`. Because the counters are initialized to zero, the first output of data will occur before any calculations on the concentration field have been performed; therefore, all the output files begin with time zero data.

Next, the subroutines `GetTheta` and `GetVel` are called, but depending on whether or not the problem has transient flow conditions, the input functions perform differently. If the problem has transient flow conditions, these functions behave as previously described, by overwriting the previous old time level information with the previous new time level and reading in the hydraulic data for the current new time level. Alternatively, if the OSERTM is solving a problem that does not have transient flow conditions, these functions merely copy the hydraulic conditions from the new time level to the old time level, but make no attempt to read new information from the `MOISTURE` or `SPDIS` files. When the necessary information for the new time step is held in memory, another loop, which contains calls to subroutines that solve the advection and dispersion subproblems described in Sections 2.4 and 2.5, begins and repeats for each chemical species.

The advection subroutine, `TVD_3d`, begins by creating a new three-dimensional array to hold the concentration field. This array is two cells larger in each dimension than the actual concentration field. The interior cells of this array are initialized with the concentration field that was passed into the `TVD_3d` function, leaving a shell of empty cells around the boundaries. The initial values for these ghost-nodes are determined by calling the function `ApplyChem`. `ApplyChem` reads in the information provided about the flux boundary conditions in the `PATCH` input file. Then the ghost-nodes at the soil surface are assigned values according to Equation (2.13). All the boundaries are treated as flux type, but as mentioned in Section 3.2, the assumption is made that mass will enter only through the soil surface. Once

the boundaries are set, Equation (2.11) is used to determine the internodal concentrations. The value of the limiter function necessary for this calculation is determined using the subroutine `GetPsi`. This function determines the value of the limiter function shown in Equation (2.12), depending on the input variable `choice`, which specifies the order of the TVD approximation. Once the internodal concentrations in each of the three directions have been determined, `TVD_3d` solves for the mass transfer between cells by solving Equation (2.10). This calculation is performed by a loop over all the cells in the model domain using a call to the function `PTransfer`. The `PTransfer` subroutine determines the contribution to the aqueous phase mass balance from both the gaseous and sorbed phases, based on the equilibrium assumptions shown in Equations (2.3) and (2.2). Also, during this loop, a check is made to see if one or more drains pass through the cell, and if so, the calculation accounts for the mass lost to those drains. When the new concentration at each of the cells in the model domain has been calculated, the new concentration field is copied to the global array that passes the result out of the `TVD_3d` subroutine. Finally, before exiting, the mass flux through the side boundaries is summed and stored for possible output by the `Main` program.

The dispersion subroutine `XDisp2` begins by allocating two new arrays that have the same dimensions as the actual concentration field and initializing these arrays with the concentration field at the old time level. These three-dimensional arrays will hold the U approximation and the V approximation to the concentration field at the new time level and will be called the U array and the V array, respectively. The function `CalcDisp` is then called. This subroutine calculates the apparent dispersivity tensor, defined in Equation (2.14), by first calculating the aqueous dispersion tensor using Equations (2.19), (2.20), (2.21), (2.22), (2.23), and (2.24). However, rather than using the pore water velocity, as specified in these equations, the specific discharge ($q = \theta_w \nu$) is used, resulting in the calculation of the effective aqueous dispersion tensor ($\hat{D_{ij}} = \theta D_{ij}$). Since at each cell face, only the directional component of the specific discharge vector in the direction perpendicular to the plane of the cell face is defined, linear averaging is used to determine the other components of the specific discharge vector needed to solve for the apparent dispersivity tensor at the cell faces. After the aqueous phase dispersion tensor has been defined, the effective gas phase diffusion terms shown in Equation (2.25) are added.

Once the apparent dispersion tensor is completely defined, Equations (2.32) and (2.33) are solved using zero-flux boundary conditions, as de-

scribed in Equation (2.40), and a call to the `PTransfer` subroutine, which is used to determine the effect of sorbed and gaseous phase concentrations on the aqueous phase mass balance, based on the equilibrium assumptions shown in Equations (2.3) and (2.2). After completing the calculations for the U and V approximations, the U and V arrays are averaged cell by cell, and the result is written in to the global concentration array, which is then passed out of the `XDisp2` subroutine.

After the dispersion subroutine has been called for each chemical species, the subroutine `Reactsielr` is called to solve for the nitrification, denitrification, and aerobic degradation of soil organic matter. This is done by solving Equations (2.54), (2.55), and (2.56) for each cell in the model domain, using a call to the `PTransfer` subroutine. The result is stored in the global concentration array, which is then passed out of the subroutine.

At this point, the time loop repeats itself for the next time step. When the stop time is reached, as specified in the `PARAM` namelist, the time loop exits, and data is output for the final conditions. Finally, the memory in which the global arrays are stored is deallocated and the program exits.

3.5 Verification of the Subroutines

After each of the subproblem solvers was developed, it was tested against accepted solutions to check for programming correctness and method accuracy. After the subroutines were tested individually, the OSERTM was validated using a three-dimensional analytical model.

To verify the advection subroutine, two tests were performed. The first test was a one-dimensional advancing front problem. In this problem, the model domain is a one-dimensional region of length 1 meter, with specific discharge ($\nu\theta$) equal to 0.005 m/day. The spatial domain was discretized into 20 cells, each of length (Δx) 0.05 meters, while the temporal domain was discretized into time steps (Δt) of 1 day. This time and space discretization resulted in a Courant number ($\mathcal{U}$) of 0.1. After 100 days, the concentration front should have been located at $x = 100\,(\nu\theta_w)$ m. The results of the true solution, the first-order upstream method, Roe's Superbee TVD method, and the ULTIMATE TVD method can be seen in Figure 3.4. As expected, the ULTIMATE TVD and Roe's Superbee methods exhibited significantly less numerical dispersion than the first-order upwind method, though surprisingly Roe's Superbee slightly outperformed the ULTIMATE TVD method in this test. However, despite varying amounts of non-physical dispersion, all three methods resulted in valid approximations for the true

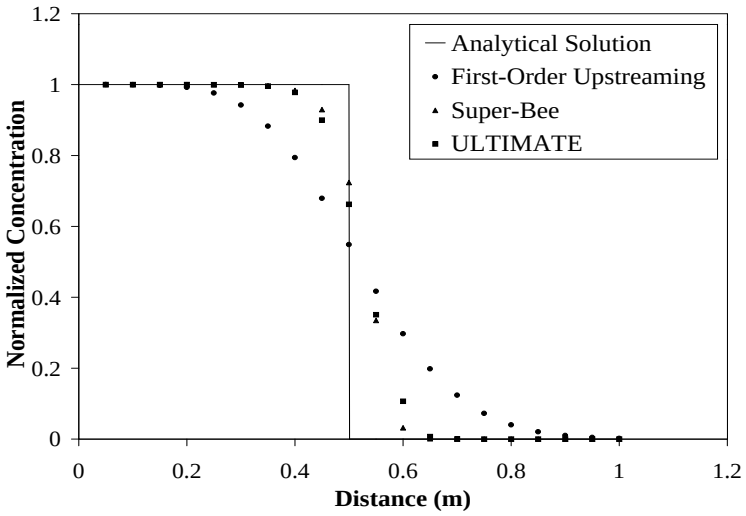

Figure 3.4: Concentration profiles of steady one-dimensional pure advection of a continuous source after 100 seconds, solved with a first-order upwind method, Roe's Superbee method, and the ULTIMATE TVD method. Note the relative accuracy of each method.

solution. To further support this claim, a mass balance was performed on each of the three solutions by numerically integrating the area under the curve. This calculation showed that mass was indeed conserved in each of the three cases.

The second test was a two-dimensional advancing pulse with uniform, multidirectional flow. In this case a two-dimensional grid was defined with $\Delta x = \Delta y = 0.5$ m. The specific discharge was specified to be 0.2 m/day at a 45° angle to the coordinate axis, so that the specific discharge in both the x-direction ($\theta\nu_x$) and the y-direction ($\theta\nu_y$) were equal to 0.1414 m/day. The time step was chosen to be 1 second, resulting in a Courant number of 0.14. After 40 days, the center of mass of the pulse should have moved to location X = 10.7 m, Y = 10.7 m without distortion, but instead, it can be seen in Figure 3.5 that the plume spread out along the axis perpendicular to the direction of flow; however, this result exhibits considerably less numerical dispersion than the first-order upwind method applied to the same problem, as shown in Figure 3.6.

These two comparisons of the advection subroutine to analytical solutions suggest that the TVD solver can accurately solve the advection subproblem, and that the higher-order TVD methods are subject to significantly less distortion than the first-order upwind method. This result is consistent with the literature (*Zheng and Wang*, 1999; *Chilakapati and Yabusaki*, 2000).

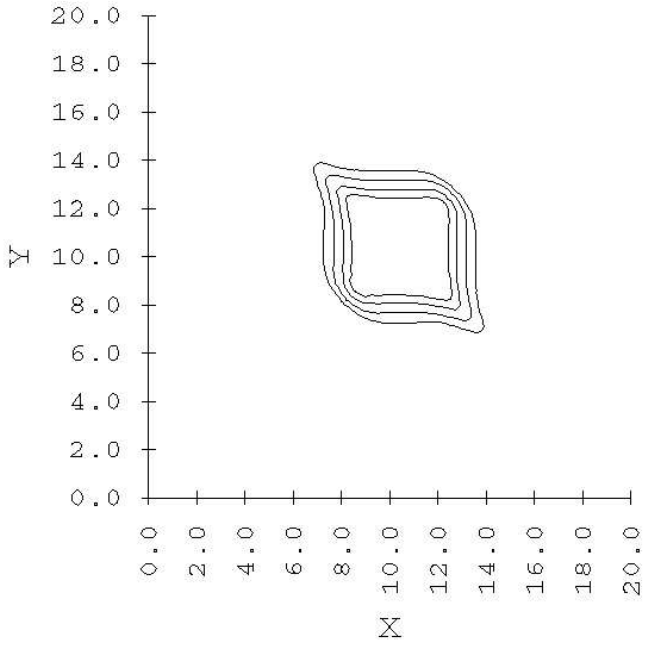

Figure 3.5: ULTIMATE TVD approximation of the concentration profile of a square plume with sides of length 10 meters after advecting for 40 days. Initially, the center of mass of the plume was located at X=Y=10 m. Flow is steady moving from the bottom left to the top right of the figure at a 45° angle to the axes. The contours are drawn at 90%, 75%, 50%, and 25% of the original concentration.

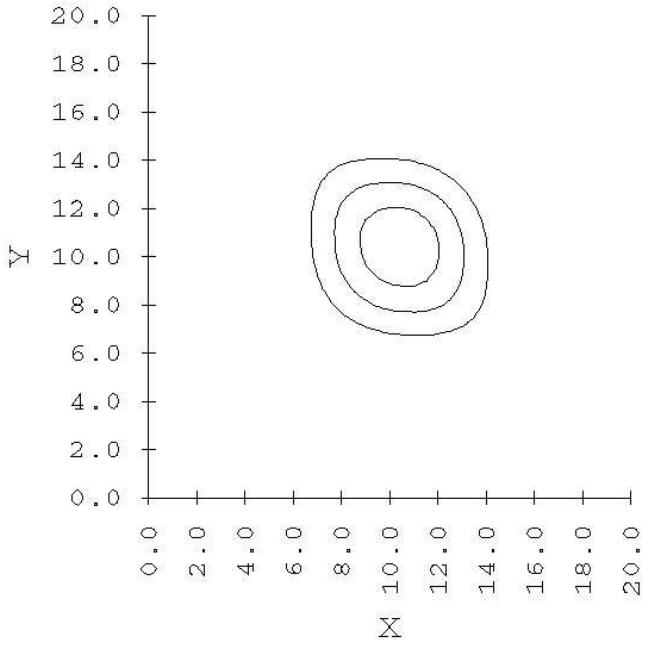

Figure 3.6: First-order upwind approximation of the concentration profile of a square plume with sides of length 10 meters after advecting for 40 days. Initially, the center of mass of the plume was located at X=Y=10 m. Flow is steady moving at 0.2 m/day from the bottom left to the top right of the figure at a 45° angle to the axes. The contours are drawn at 75%, 50%, and 25% of the original concentration.

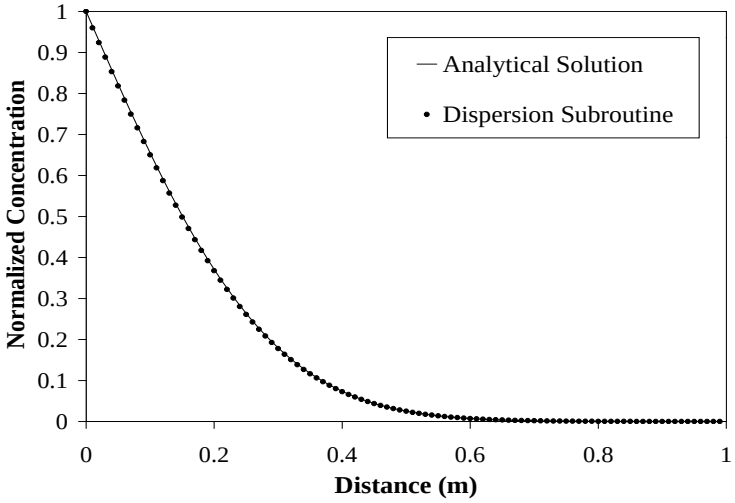

Figure 3.7: Concentration profile of a one-dimensional pure diffusion problem after 1000 seconds, solved by both an analytical model and the ADE method. Initially the concentration was zero everywhere except at X=0. The boundary X=0 is a constant concentration boundary with unit concentration. The diffusion coefficient is $2.5*10^{-5}$ m^2/s.

The dispersion subroutine was validated using two pure diffusion scenarios. The first scenario is a one-dimensional diffusion of a constant concentration boundary. The analytical solution for this scenario is:

$$\frac{C}{C_0} = erf\left(\frac{x}{2\sqrt{D^{\star}t}}\right) \tag{3.2}$$

where $D^{\star} = 2.5 * 10^{-5}$ m^2/s. To solve this problem with the dispersion subroutine, the spatial and temporal domains were discretized into 0.01 meters and 1.0 second lengths, respectively, resulting in a $\mathcal{D}$ value of 0.25. After 1000 seconds, a mass balance was performed by numerically integrating the area under the curve shown in Figure 3.7. This resulted in a quantification of only 0.73% error.

A similar scenario was also used to test the hypothesis that the ADE method is unconditionally stable. In this case, the same diffusion coefficient and space discretization were used. However, the discretization of the temporal domain was chosen to be either 1, 4, 10, or 100 seconds, resulting in stability parameters ($\mathcal{D}$) of 0.25, 1, 2.5, and 25, respectively. As can be seen in Figure 3.8, the ADE solutions after 1000 seconds do not oscillate, but seem to contain less mass as $\mathcal{D}$ increases. This latter observation was verified by numerical integration of the area under the curves, which revealed

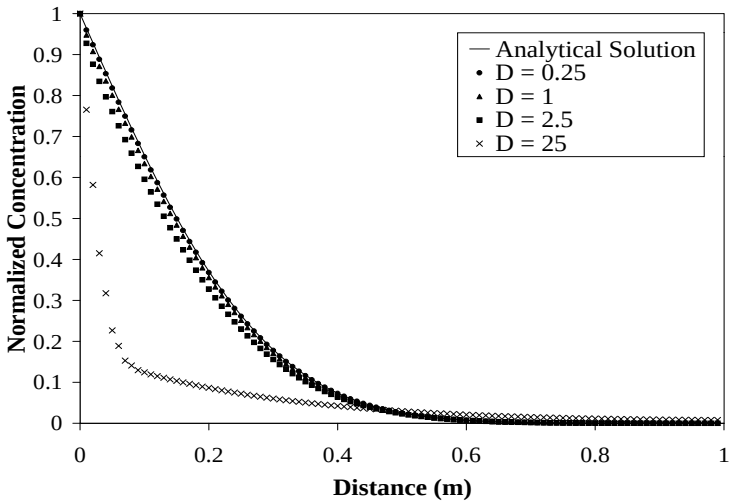

Figure 3.8: Concentration profile of the one-dimensional pure diffusion case presented in Figure 3.7, solved with the ADE method and increasing values of the stability parameter $\mathcal{D}$.

a mass error of 0.73%, 3.3%, 9.1%, and 62% for each of the cases, respectively. This test shows that the ADE method does not become unstable as $\mathcal{D}$ increases, but that the mass error of the method increases as $\mathcal{D}$ increases. Further tests suggest that the method has good accuracy when $\mathcal{D} \leq 1$. This result is similar to one presented by Morita and Yen (2000), who noted that the accuracy of the ADE method is better than that of Brian's ADI method when $\mathcal{D} \leq 1$.

The second scenario is a two-dimensional problem, which has the following boundary and initial conditions:

$$\begin{aligned} C(t = 0, x > 0, y > 0) &= C_0 \\ C(t = 0, x = 0, y = 0) &= 0 \end{aligned}$$

where: $D^\star = 1 * 10^{-5}$ m^2/s. Carslaw and Jaeger (1959, p. 171) have presented the following analytical solution for this scenario:

$$\frac{C}{C_0} = erf\frac{x}{2\sqrt{Dt}} erf\frac{y}{2\sqrt{Dt}} \tag{3.3}$$

This problem was solved with the dispersion subroutine by discretizing the spatial domain into 0.01 meter by 0.01 meter blocks, and the temporal domain into 10 second durations, resulting in a stability parameter of $\mathcal{D} = 1/2$. Since the dispersion subroutine implements zero flux boundary conditions,

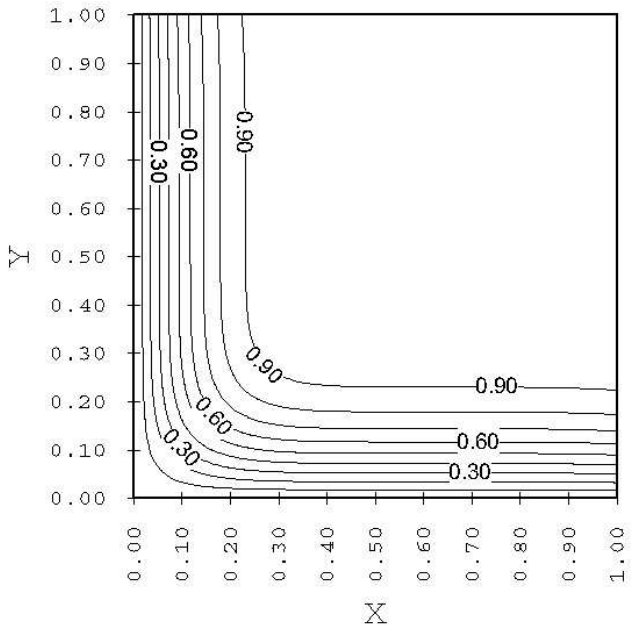

Figure 3.9: ADE method approximation to the concentration profile after 1000 seconds, resulting from a classic two-dimensional pure diffusion test problem. Initially the concentration was at unit concentration everywhere, except at the boundaries X=0 and Y=0. The boundaries X=0 and Y=0 are constant concentration boundaries, both with values of zero. The diffusion coefficient is $1*10^{-5}$ m^2/s.

the constant concentration boundary condition was approximated by setting the concentration at the boundary nodes to zero at the end of each time step. The resulting concentration profile after 1000 seconds (shown in Figure 3.9) had at most, 5% error relative to the initial concentration, when compared with the analytical solution. Figure 3.10 shows the difference between the dispersion subroutine solution and the analytical solution. It can be seen in this figure that the error is largest near the boundaries. This result was expected because of the method of implementing the boundary condition described above. Away from the boundaries, however, there is good agreement between the two solutions.

These comparisons of the dispersion subroutine to analytical solutions suggest that the ADE solver can effectively solve the dispersion subproblem, and that the ADE method will not oscillate with increasing stability parameter $\mathcal{D}$, though the best accuracy is achieved when this value is less than or equal to one.

In order to validate the use of the independent pseudo-analytical solu-

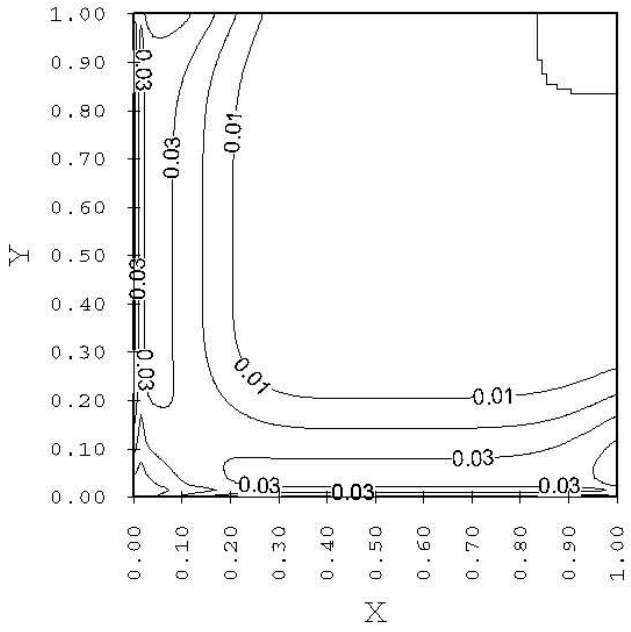

Figure 3.10: The difference between the ADE method approximation (shown in Figure 3.9 and the analytical solution to this classic two-dimensional pure diffusion test problem. The contours are drawn at 0%, 1%, 2%, 3%, 4%, and 5% of the initial concentration.

Table 3.2: Reaction parameters used in comparison of the pseudo-analytical reaction formulation to the Runge-Kutta method.

k_{nit}	3.47*10^{-8} mg/(cm^{-3}min)	Hudson (2001d)
k_{denit}	1.38*10^{-8} mg/(cm^{-3}min)	Hudson (2001a)
k_{ox}	1*10^{-8} mg/(cm^{-3}min)	assumed value
$K_{NH_4^+}$	1*10^{-4} mg/cm^{-3}	MacQuarrie and Sudicky (2001)
$K_{NO_3^{-2}}$	5*10^{-5} mg/cm^{-3}	MacQuarrie and Sudicky (2001)
K_{O_2}	1*10^{-4} mg/cm^{-3}	MacQuarrie and Sudicky (2001)
k_I	4.0	assumed value

tion to the reaction system described in Section 2.6, a reactive transport problem was solved with both the reaction subroutine, using the pseudo-analytical equations, and an alternate subroutine, which employs a fourth-order Runge-Kutta method. The problem domain was one-dimensional, 50 cm in length, and discretized into 0.05 cm blocks. Time was discretized into 1-minute durations. Flow was steady at 0.005 cm/min. Initially there was a dissolved oxygen concentration in the model domain of 8.52*10^-3 mg/cm^3. The initial concentrations of ammonium and nitrate were zero. Ammonium ion moved into the model domain through a constant flux boundary at a rate of 1.25*10^{-8} mg/min, which is equivalent to a ghost node concentration of 1*10^{-3} mg/cm^3. There is also a flux of dissolved oxygen into the model domain of 1.1*10^{-7} mg/min, which is equivalent to a ghost node concentration of 8.52*10^{-3} mg/cm^3. The reaction parameters can be seen in Table 3.2. After 100 time steps the results from both the pseudo-analytical and the Runge-Kutta solver for each of the chemical species can be seen in Figures 3.11, 3.12, and 3.13. The maximum deviation of the pseudo-analytical approximation from the fourth-order Runge-Kutta solution was 1.9%, 0.33%, and 0.28% for the ammonium, nitrate, and oxygen profiles, respectively. Therefore, it can be seen that the decoupled reaction system solved with the pseudo-analytical equations is a good approximation of the true system of ODEs that describe the reaction system.

In order to validate the combined transport model, 3DADE, an analytical model for the three-dimensional ADRE, developed by the U.S. Salinity Laboratory in Riverside, CA, was used (*Leij and Bradford*, 1994). In this model, solute is input as a rectangle in the x,y plane, centered around the z-axis. Flow is steady and one-dimensional in the z-direction. For this test, the source was defined to be continuous, with dimensions of 100 meters in the x-direction and 100 meters in the y-direction. The flow rate was constant

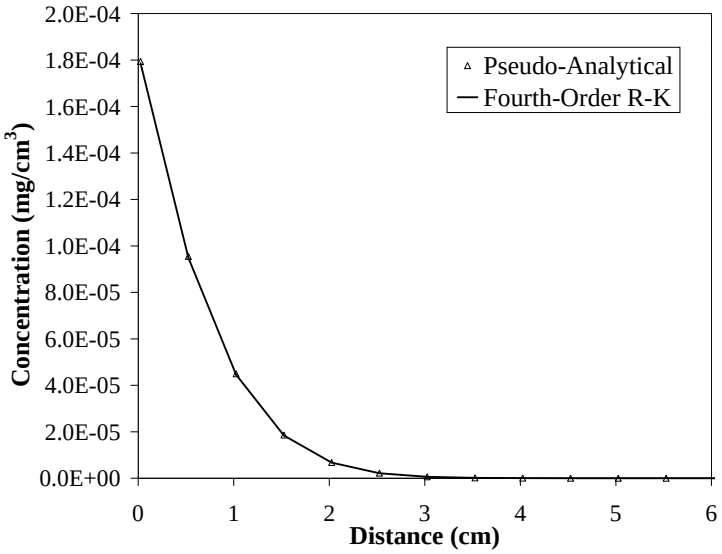

Figure 3.11: Concentration profile of the ammonium ion after 100 minutes, solved with the OSERTM using both the pseudo-analytical reaction subroutine and a fourth-order Runge-Kutta method. The maximum deviation of the pseudo-analytical method from the Runge-Kutta method is 1.9%.

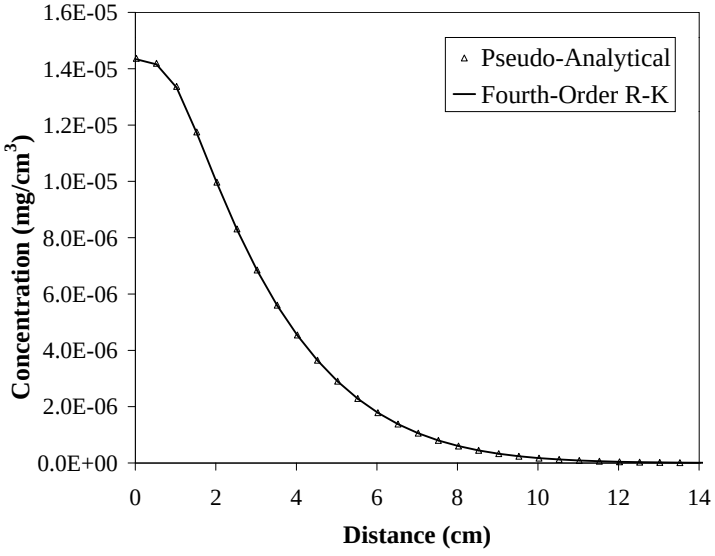

Figure 3.12: Concentration profile of nitrate after 100 minutes, solved with the OSERTM using both the pseudo-analytical reaction subroutine and a fourth-order Runge-Kutta method. The maximum deviation of the pseudo-analytical method from the Runge-Kutta method is 0.33%.

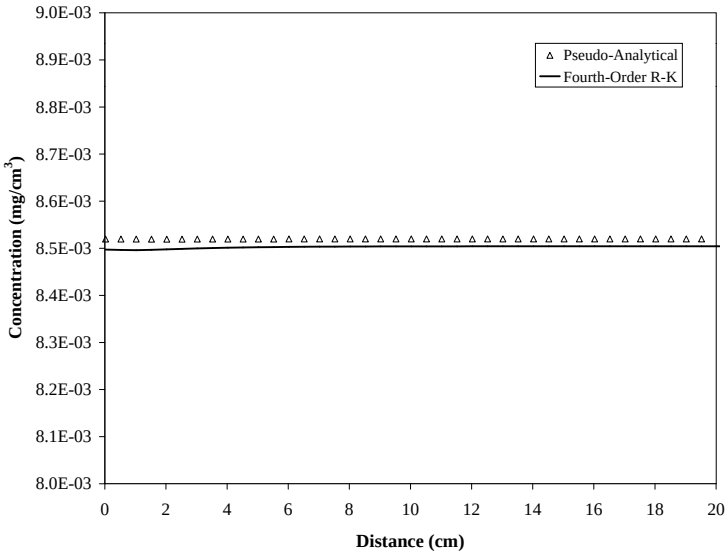

Figure 3.13: Concentration profile of dissolved oxygen after 100 minutes, solved with the OSERTM using both the pseudo-analytical reaction subroutine and a fourth-order Runge-Kutta method. The maximum deviation of the pseudo-analytical method from the Runge-Kutta method is 0.28%.

at 1.0 m/day, and the dispersion coefficients in the longitudinal, transverse horizontal, and transverse vertical directions were set to 3.0 m^2/day. This problem was solved with the OSERTM by discretizing the spatial domain into 64,000 uniform 10 meter by 10 meter by 10 meter cubes, and the temporal domain into 5-day intervals. This resulted in a Courant number of 0.125, and a $\mathcal{D}$ value of 0.01. The results along the transect X=0 of the OSERTM and the 3DADE model after a 125-day simulation can be seen in Figures 3.14 and 3.15, respectively. The error between these two models was, at most 3% of the initial concentration. Figure 3.16 shows the absolute difference between the two models along this transect. It can be seen in this figure that the error is largest near the source boundaries - most likely a result of the spatial discretization size.

As can be seen in Figure 3.16, the performance of the combined transport model for three-dimensional problems is comparable to that of an analytical model, though the space discretization may result in errors near the source boundaries.

Finally, to validate the combined reactive transport model, the solution of the OSERTM to a one dimensional problem including advection, dispersion, sorption, and first-order decay was compared to that of an analytical model developed by Valocchi *et al.* (2001). In this problem, the spatial

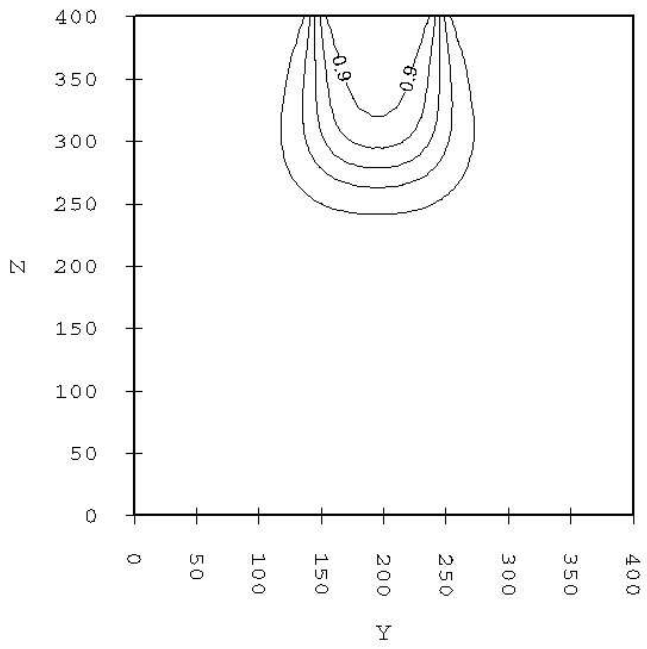

Figure 3.14: Solution to a three-dimensional problem with a steady one-dimensional flow field after 124 days using the OSERTM. Flow is downwards in the z-direction at m/s. Dispersion in both the lateral and transverse directions is m^2/s. The contours are drawn at 90%, 70%, 50%, 30%, and 10% of the original concentration.

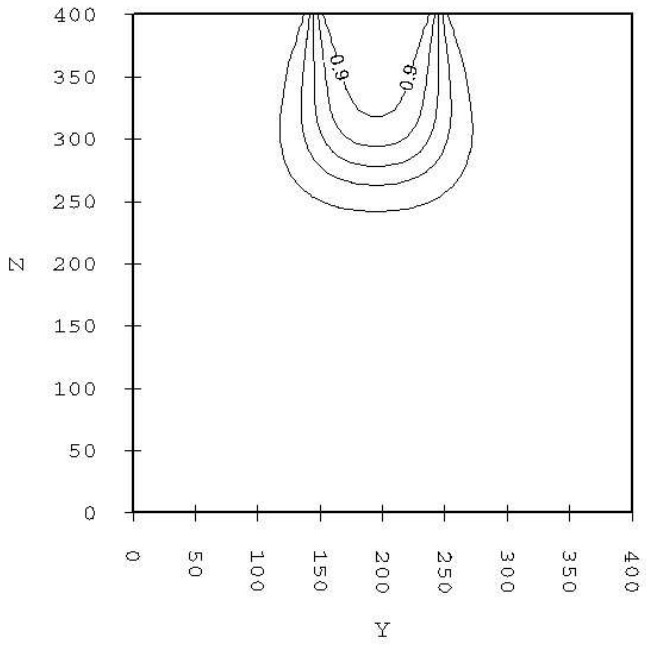

Figure 3.15: Analytical solution to the problem shown in Figure 3.14, solved with the 3DADE model (Leij and Bradford, 1994). The contours are drawn at 90%, 70%, 50%, 30%, and 10% of the original concentration.

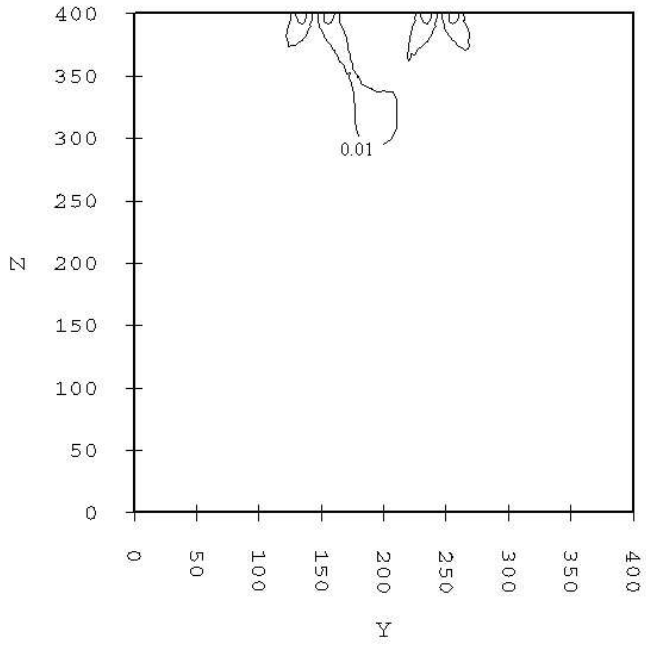

Figure 3.16: The difference between the two models shown in Figures 3.14 and 3.15. The contours are drawn at 3% and 1% of the original concentration.

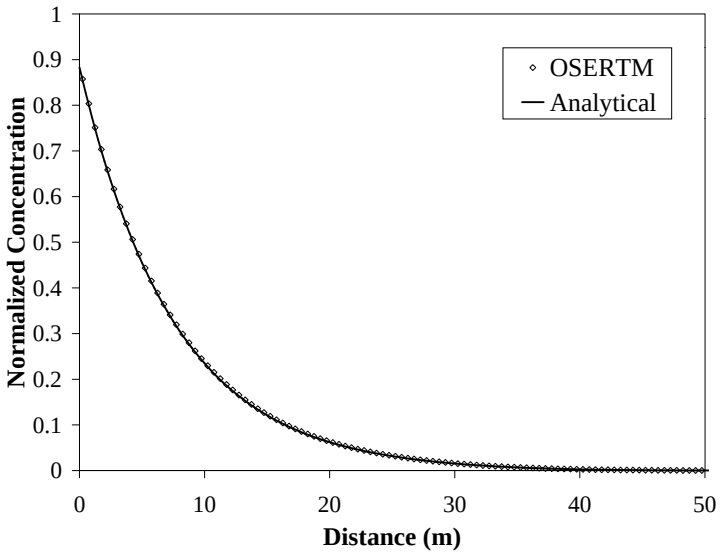

Figure 3.17: Results of the OSERTM and an analytical model for a 100 day simulation of a one-dimensional problem including advection, dispersion, sorbtion and first-order decay.

domain is semi-infinite, and the solute entered the model domain through a constant flux boundary. The retardation coefficient $(1 + K_d\rho_b)$ was 3.0, the specific discharge $(\nu\theta)$ was 1.0 m/day, the longitudinal dispersivity was 1.0 m (resulting in a coefficient of dispersion of 1.0 m^2/day), and the first-order rate constant was 0.05 day^{-1}. In order for the problem to be solved by the OSERTM, the spatial domain was discretized into 400 cells, each of length 0.5 meters, while the temporal domain was discretized into time steps of 0.2 days. This time and space discretization resulted in $\mathcal{U} = 0.133$ and $\mathcal{D} = 0.267$. The results from both models, after a simulation of 100 days, can be seen in Figure 3.17. In order to quantify the error of the OSERTM solution, a mass balance was performed by numerically integrating the area under the curves shown in this figure. This mass balance revealed an error of only 0.86%, confirming that the OSERTM solution was valid.

3.6 Summary

In this chapter, the format and content of the input and output files for the OSERTM were described. The structure of the program was described in detail, and the implementation of the mathematical equations developed in Chapter 2 was described. The advection, dispersion, and reaction models were tested separately by comparison with analytical solutions. Then the

combined model was tested by comparison with two analytical models. The first model was a one-dimensional reactive transport model, and the second was a three-dimensional transport model. In the next chapter, the OSERTM will be applied to two sample problems.

Chapter 4

Application of the OSERTM to Sample Problems

In this chapter, the OSERTM will be applied to a generic two-dimensional field scale system, which will provide a more realistic environment for the simulation of solute transport problems. Despite the ability of both the OSERTM and the conjunctive flow model to solve problems in three-dimensions, a two-dimensional system was chosen because it requires less computation than a similar three-dimensional problem. A two-dimensional cross-section was considered to be a good starting place to begin evaluating the abilities of the OSERTM. Before discussing these applications, however, it is necessary to describe the hydrologic model used in this research.

4.1 A Brief Introduction to the Conjunctive Hydrologic Model

As mentioned in Chapter 2, the OSERTM does not have the capability to solve for hydrologic conditions; instead this information must be input from a hydrologic model. For this research, a simplified two-dimensional version of the subsurface model described by Morita and Yen (2000, 2002) was used. In the future, however, the complete model, which combines a conjunctive overland-subsurface flow model with a pipe network model to simulate tile drains, will be used. In Morita and Yen's model, overland flow is solved in two dimensions, subsurface flow is solved in three dimensions, and tile flow is solved in one dimension. The simplified two-dimensional subsurface model uses Larkin's ADE method to solve the two-dimensional Richards equation shown below.

$$\frac{\partial \theta_w}{\partial t} = \frac{\partial}{\partial x} K(\theta_w) \frac{\partial \psi}{\partial x} + \frac{\partial}{\partial z} K(\theta_w) \left(\frac{\partial \psi}{\partial z} + 1 \right) \tag{4.1}$$

where:

θ_w	is the moisture content [L^3 water/ L^3 aquifer]
$K(\theta_w)$	is the effective hydraulic conductivity (assumed isotropic) [L/T]
ψ	is the pressure head [L]

The two parameter van Genuchten relationships shown in Equations (4.2) and (4.3) were used to relate pressure head to moisture content and moisture content to hydraulic conductivity, respectively (*Tindall and Kunkel*, 1999; *Charbeneau*, 2000).

$$\Theta = \left(\frac{1}{1+(\alpha\hat{\psi})^n}\right)^m \tag{4.2}$$

$$K(\theta_w) = K_{sat}\sqrt{\Theta} \tag{4.3}$$

$$\Theta = \frac{\theta_w - \theta_{wr}}{\eta - \theta_{wr}}$$

where:

α	is a fitted parameter [L^{-1}]
n	is a fitted parameter
$m = 1 - \frac{1}{n}$	is a fitted parameter
$\hat{\psi} = -\psi \geq 0$	is the suction head [L]
K_{sat}	is the saturated hydraulic conductivity [L/T]
θ_{wr}	is the irreducible moisture content [L^3 water/ L^3 aquifer]
η	is the porosity [L^3 voids/ L^3 aquifer]

Darcy's Law was used to calculate water flux into the drain. This equation was simplified by assuming that the pressure in the drain is equal to atmospheric pressure, and combining terms to define a drain conductance (C_d) as shown in Equation (4.4) below.

$$Q_d = C_d(H) \tag{4.4}$$

where:

Q_d	is the flow rate of water into the drain [L^3/T]
H	is the hydraulic head [L]

This approach is similar to the approach used in the MODFLOW drain package (*McDonald and Harbaugh*, 1988). For the following simulations, the drain conductance was set to a very large value so that flow into the drain would be limited only by the porous medium, not by the conductivity of the drain.

This model requires the user to input both the rainfall intensity over time and the spatial distribution of soil properties, using the van Genuchten parameters (α, n). Furthermore, the location of the tile drains and the type of boundary conditions for the sides of the model domain must be specified. The model requires the top of the model domain (soil surface) to have a specified flux boundary, and the bottom of the model domain to have a zero flux boundary. When this information has been properly specified, the hydrologic model will solve for the specific discharge at all cell boundaries, the moisture content at all nodes, the volumetric flow rate of water out of the model domain through the sides, and (if a tile drain passes through the cell) the volumetric flow rate of groundwater from that cell into the drain. All of this information is output for each of the time steps using the format shown in Appendix B. It is important to note that the hydrologic model must be constructed using the same spatial discretization as will be used in the OSERTM. If the solute transport simulation will have transient flow conditions, then the temporal discretization must be the same as well.

4.2 Contaminant Transport Scenario

The generic system was designed in collaboration with Dr. Robert Hudson of the Department of Natural Resources and Environmental Science, at the University of Illinois at Urbana-Champaign, and was based on conditions typically found on agricultural fields in Central Illinois. The hypothetical soil cross-section is drained by one tile line, as well as by a drainage ditch that borders the right side of the cross-section. Both the tile and the ditch run perpendicular to the plane of the cross-section. The tile line is located at a depth of 1 meter, and is oriented 15.5 meters to the right of the left boundary and 30.5 meters to the left of the ditch. This geometry reflects typical drain depths, half-spacing, and setbacks from drainage ditches in Central Illinois (*Gentry et al.*, 2000; *Hudson*, 2001d). The ditch is 3 meters deep, and it is assumed that water flows out of the ditch faster than it flows out of the soil column; therefore, the ditch is always empty. See Figure 4.1 for a schematic of the physical domain of the problem.

In addition to the cross-section geometry and soil properties, it is also necessary to define the boundary conditions for the hydrologic model. The wall of the ditch presents a logical location for the right boundary of the model domain; however, because this boundary must allow for a transient, non-uniform flux of water, the boundary condition was simplified by making the assumption that the water level in the ditch was always at the level of

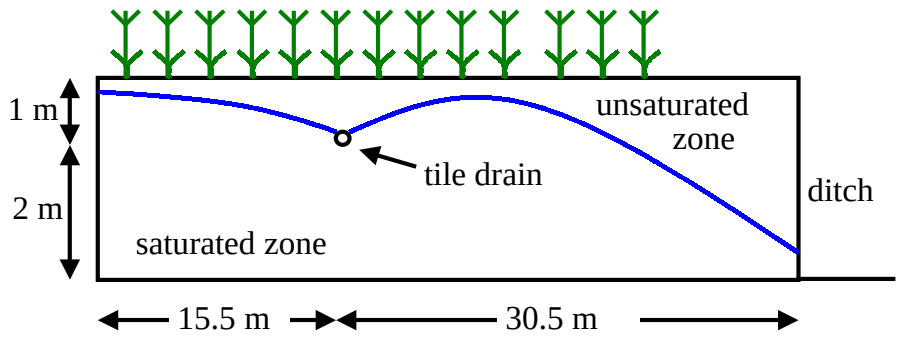

Figure 4.1: Schematic of a hypothetical field cross-section.

the bottom of the ditch. This allowed a series of ghost-nodes, adjacent to the boundary nodes located at the ditch wall, to be defined with a constant pressure equal to atmospheric pressure (0 atm). The left boundary was defined using the further simplification that there is a symmetrical drain and ditch configuration on the opposite side of this boundary. This symmetry results in a hydraulic divide located at the left boundary; thus, this boundary can be defined as a no-flow boundary. For simplicity, the bottom boundary was also defined as a no-flow boundary. The top boundary was modeled as a flux-type boundary. Additionally, it was assumed that the recharge water is at equilibrium with atmospheric oxygen, and thus has a dissolved oxygen concentration of 8.52 mg/L.

The chemicals of interest were ammonium, nitrate, oxygen, and a non-reactive tracer. It is assumed that only ammonium will sorb to the soil, and that the only chemical that can be in the gas phase in appreciable quantities is oxygen. The soil-water partition coefficient for ammonium in Central Illinois soil is taken to be $1.0*10^{-2}$ cm^3/mg (*Hudson*, 2001d), the Henry's Law coefficient of oxygen is taken to be $4.38*10^4$ atm/mole fraction (*Thibodeaux*, 1979), and the gas phase diffusion coefficient is taken to be 34.7 cm^2/min (*MacQuarrie and Sudicky*, 2001). All of these chemicals were assumed to have a coefficient of molecular diffusion of 6.94*10-4 cm^2/min, a value within the range of common ions (*Burnett and Frind*, 1987).

An attempt was made to determine reaction parameters for Central Illinois; where data were lacking, values were either taken from the literature or assumed. The maximum rates of nitrification and denitrification in Central Illinois were taken to be $3.47*10^{-8}$ $mg/(cm^3\ min)$ (*Hudson*, 2001d) and $1.38*10^{-8}$ $mg/(cm^3\ min)$ (*Hudson*, 2001a), respectively. No data were avail-

Table 4.1: Solute transport parameters used in sample applications.

$K_{d_{NH_4}}$	1.00*10^{-2} cm^3/mg	Hudson (2001d)
H_{O_2}	4.38*10^4 atm /mole fraction	Thibodeaux (1979)
D_g^0	34.7cm^2/min	MacQuarrie and Sudicky (2001)
$D^\star$	6.94*10-4 cm^2/min	Burnett and Frind (1987)
k_{nit}	3.47*10^{-8} mg/(cm^{-3}min)	Hudson (2001d)
k_{denit}	1.38*10^{-8} mg/(cm^{-3}min)	Hudson (2001a)
k_{ox}	1.38*10^{-8} mg/(cm^{-3}min)	assumed value
$K_{NH_4^+}$	1*10^{-4} mg/cm^{-3}	MacQuarrie and Sudicky (2001)
$K_{NO_3^{-2}}$	5*10^{-5} mg/cm^{-3}	MacQuarrie and Sudicky (2001)
K_{O_2}	1*10^{-4} mg/cm^{-3}	MacQuarrie and Sudicky (2001)
k_I	4.0	assumed value

able on the rate of degradation of soil organic matter (SOM). Because rates of biological reactions in the subsurface are highly dependent upon location, it was preferable to assume a value for this reaction, rather than to use a value from the literature. Because MacQuarrie and Sudicky (2001) use equivalent values for the rate of the degradation of SOM and the rate of denitrification, this rate was assumed to be 1.38*10^{-8} mg/(cm^3min), the rate of denitrification used in this research. The half-rate constants for nitrification, denitrification, and degradation of SOM were taken to be 1*10^{-4} mg/cm^3, 5*10^{-5} mg/cm^3, and 1*10^{-4} mg/cm^3, respectively (*MacQuarrie and Sudicky*, 2001). The inhibition factor k_I was fitted to the expected result when applied to several preliminary model runs. Table 4.1 summarizes the solute transport parameters.

Initially, there was no ammonium, nitrate, or tracer in the model domain. Dissolved oxygen was present in the unsaturated zone at a concentration of 8.52*10^{-3} mg/cm^3. Ammonium, dissolved oxygen, and the tracer were allowed to enter the system through an advective flux boundary at the land surface. Oxygen entered the model domain via the recharge water at a concentration of 8.52*10^{-3} mg/cm^3. This flux of oxygen was applied over the entire land surface for the duration of the simulation. Ammonium, however, was only allowed to enter the model domain over a portion of the land surface. Farm managers often leave a buffer zone between a drainage ditch and the crop. Because of this practice, it was assumed that fertilizer would not be applied within 5 meters of the ditch. Furthermore, because fertilization is not a constant process, it was assumed that the fertilizer (represented by ammonium) would be applied only during a specified interval, and in such a

quantity as to enter the model domain in the recharge water at a concentration of $1.0*10^{-3}$ mg/cm^3. The tracer was allowed to enter the model domain in a similar manner as ammonium, and was used to check for continuity of the flow regime.

4.3 Steady State Flow Scenario

The first problem to which the OSERTM will be applied was designed not only to demonstrate the model's capabilities, but also to reveal defects in the implementation of the model. To meet these goals, a problem scenario was designed to comprise aspects of a real world application, while, at the same time, maintaining the degree of simplicity necessary to make any errors in the model results apparent. This scenario uses steady, two-dimensional flow conditions in a heterogeneous aquifer in order to evaluate the effect of dissolved oxygen concentration on the amount of nitrate that is lost from the soil system to surface water.

The soil in the hypothetical cross-section was modeled after Drummer soil, typical of Central Illinois, and consists of seven distinct layers. The four-parameter van Genuchten model (*Vogel and Cislerova*, 1988) parameters for these layers were provided by Dr. Richard Cooke of the Department of Natural Resources and Environmental Science at the University of Illinois at Urbana-Champaign, and can be seen in Table 4.2. Drain tiles are commonly installed by trenching, a method that involves digging a trench, placing a tile line and some filter material in its bottom, and backfilling it with soil material (*Broughton and Fouss*, 1999). This process may disturb the soil layers directly above the tile line (*Cooke*, 2001b). For this reason, it was assumed that this soil would have different hydraulic properties than the surrounding soil. For this reason, a homogeneous soil column with higher permeability was placed directly above the drain. This column had a width of 1 meter, a size that falls within the range of typical widths of trenches dug by machines (*Broughton and Fouss*, 1999). Since there is no information from which to determine van Genuchten parameters for this soil, it was assumed to behave similarly to the AP layer, defined in Table 4.2, because this layer has the highest permeability of any of the Drummer soil layers. Furthermore, because data for the soil properties below 117 centimeters were unavailable, it was assumed that the soil below this depth is the same as the soil comprising the bottom-most stratum. Figure 4.2 shows the location of each of the seven soil types.

The steady water flux through the top boundary was defined to be 10^{-4}

Table 4.2: The four-parameter van Genuchten model parameters for the seven soil types used in the steady state flow scenario (*Cooke*, 2001a).

soil layer	index	depth	θ_s	α cm^{-1}	n	K_{sat} cm/min
AP	1	0-18	0.5045	0.0359	1.1651	0.022987
A	2	18-36	0.4862	0.2712	1.1155	0.022840
BA	3	36-48	0.4760	0.5399	1.1121	0.013215
Bgl	4	48-64	0.4143	0.1374	1.0971	0.012058
Bg2	5	64-81	0.3921	0.0649	1.0997	0.0070216
Bg3	6	81-99	0.4049	0.0299	1.1200	0.0079233
Cg	7	99-117	0.4460	0.0060	1.1474	0.0064650

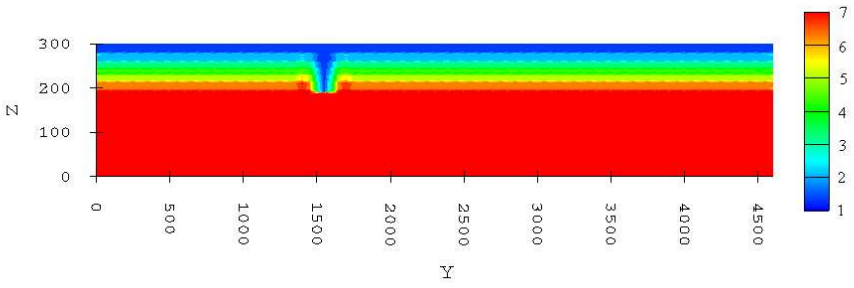

Figure 4.2: Location of each of the seven soil types used in the steady state flow scenario. Note that the vertical axis has been exaggerated by a factor of 3.

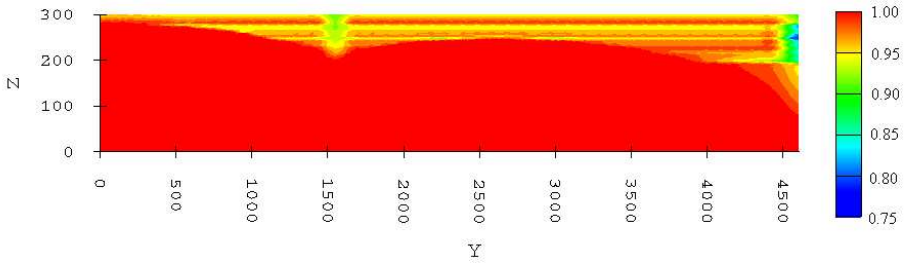

Figure 4.3: Saturation profile of the steady state flow scenario. Note that the vertical axis has been exaggerated by a factor of 3.

cm/min. This value reflects the average annual precipitation minus evapotranspiration in Central Illinois (*Hudson*, 2001c).

The initial condition of the model was chosen to be a flat water table 1 meter below the land surface, with hydrostatic conditions above and below. This initial condition would represent an equilibrium saturation profile, if the drain and ditch were impervious to flow. However, because both the drain and ditch walls are permeable, this saturation profile will immediately cause flow out of the model domain when the simulation begins. Because this model will be run with a constant recharge until steady state is reached, the flow out of the model domain resulting from the arbitrary specification of the initial conditions will not affect the model results.

The flow model was applied to the hypothetical cross-section using a spatial discretization of 5 centimeters in the vertical direction and 100 centimeters in the horizontal direction. A temporal discretization of 1 minute was used. Based on previous studies employing a similar model cross-section, it was estimated that a simulation of 10,000,000 minutes would be sufficient for the model to reach steady state. The resulting flow and saturation profiles, which can be seen in Figures 4.3 and 4.4, reflect only a 5% difference between the flow into the profile through the soil surface and the flow out of the profile through the drain and ditch walls.

After completing the flow simulation, it was observed that the simulated soil was much less permeable than expected. The explanation for this observation was that the van Genuchten parameters used corresponded to the four-parameter van Genuchten model, whereas the conjunctive flow model employs the two-parameter van Genuchten model (shown in Equations (4.2) and (4.3)). This error caused the maximum steady state infiltration rate for a particular saturation profile of the soil to be approximately 2 orders of magnitude lower than if the correct parameters had been used. Despite this

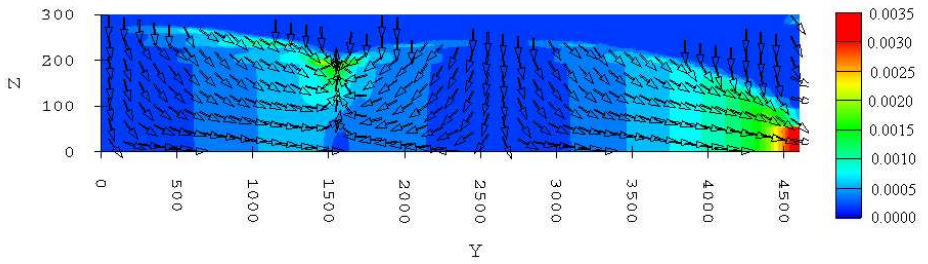

Figure 4.4: Magnitude and direction of specific discharge vectors in units of cm/min for the steady state flow scenario. Note that the vertical axis has been exaggerated by a factor of 3.

error, this simulation is still valuable to demonstrate the ability of the OSERTM to solve solute transport problems in a heterogeneous aquifer with non-uniform, multi-dimensional flow conditions in both the saturated and unsaturated zones.

To validate that the OSERTM would produce valuable results when using this flow regime, as well as to get an estimate of the travel time distribution, the transport of a non-reactive tracer was simulated using these hydrologic data, the same spatial discretization as the flow model, and a temporal discretization of 10 minutes. Furthermore, the longitudinal, transverse horizontal, and transverse vertical dispersivities for the steady state profile were chosen to be 0.2 cm, 0.02 cm, and 0.002 cm, respectively. The simulation was run for 525,600 time steps, a total of 10 years. The tracer was applied at a concentration of 1 mg/cm^3 across the entire land surface during the first minute of the simulation. A mass balance was performed by numerically integrating the area under the breakthrough curves of the drain and ditch shown in Figure 4.5. The resulting mass error of 1.7% was considered to be acceptable.

The results of this simulation indicate the existence of short flow paths, through which the solute can reach the ditch wall. The tracer that takes these paths begins to exit the soil profile almost immediately. The simulation reveals that after approximately 0.5 years, all the solute that was within the capture zone of these flow paths has exited the model domain. Around this time, the tracer has also begun to exit the model domain through the drain, indicating that the shortest flow paths to the drain take approximately 0.5 years. After approximately 1 year, the mass flux through the drain peaks and then begins to decline. The mass flux through the ditch resumes after approximately 1.25 years, and reaches a peak after approximately 1.75 years.

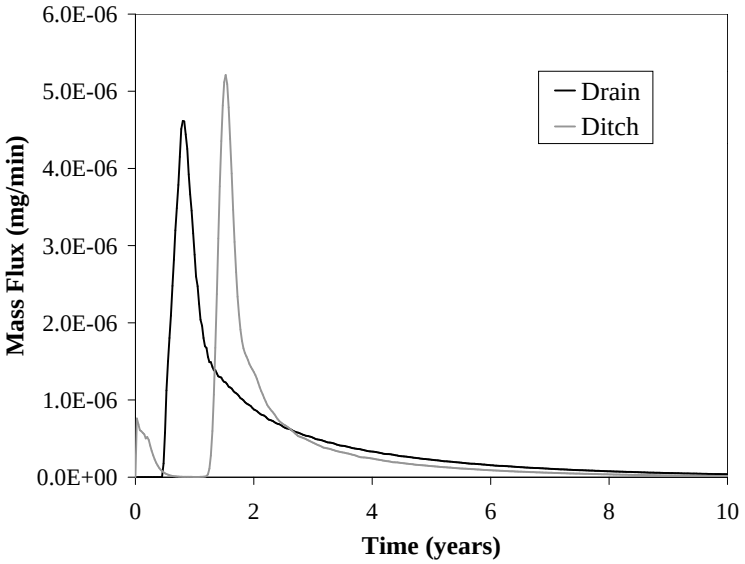

Figure 4.5: Mass breakthrough of a non-reactive tracer from the drain and ditch resulting from the steady state flow scenario.

Finally, after approximately 10 years, most of the tracer has left the model domain. These results indicate that the majority of the tracer exits the model domain via flow paths of moderate length, while the remainder exits through the short flow paths previously described, or through long flow paths (4-10 years).

The OSERTM was then applied to a series of three simulations that focused on the effect of the dissolved oxygen concentration on the amount of nitrate that exited the model domain through the drain and ditch. Each of these simulations used a different value for the rate of the consumption of SOM. In the first simulation, this rate was set to zero; thus, the effects of this reaction on the amount of nitrate that left the model domain were ignored. In the second simulation a rate of $1.38*10^{-8}$ mg/(cm^3min), the assumed value shown in Table 4.1, was used, and in the third simulation a rate of $5.00*10^{-8}$ mg/(cm^3min) was used. These three simulations will be referred to as scenarios A, B, and C, respectively. In all three of these simulations, ammonium was applied to the land surface during the first day, after which, no more ammonium entered the soil profile. Table 4.3 lists the different reaction rates described here.

All three scenarios were run for 210,240 time steps, a total of 4 years. The breakthrough curves of oxygen and nitrate for each of the three scenarios are shown in Figures 4.6 and 4.7. The breakthrough curve of ammonium

Table 4.3: Rate of SOM degradation for the three scenarios of the steady state flow problem.

Scenario	Max rate of consumption of SOM
A	0.00 mg/(cm^{-3}min)
B	1.38*10^{-8} mg/(cm^{-3}min)
C	5.00*10^{-8} mg/(cm^{-3}min)

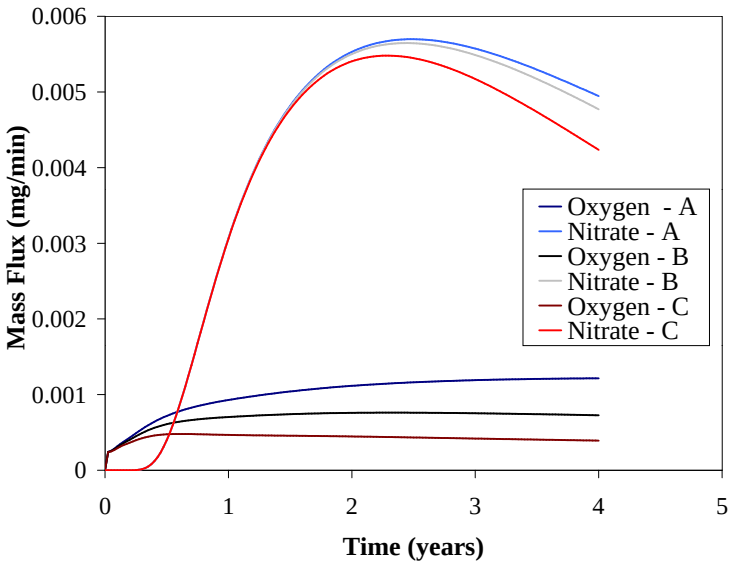

Figure 4.6: Mass breakthrough of nitrate and oxygen from the drain resulting from reaction scenarios A, B, and C.

is not shown, because no ammonium exited the model domain. The concentration profiles at 3, 6, 9, and 12 months that resulted from the simulation of scenario A are shown in Figures 4.8, 4.9, and 4.10 for ammonium, nitrate, and oxygen, respectively.

These figures show that ammonium sorbs very strongly to the soil in the top layers, and is converted to nitrate before it can reach the tile line or ditch. The large nitrate plume created by this conversion of ammonium moves through the soil profile to the drain or ditch, where it is removed from the profile. If the soil profile is dominated by regions with high concentrations of dissolved oxygen, as is the case in scenario A, then more nitrate will exit the model domain than if the profile contains more area of lower dissolved oxygen concentration, as can be seen in Figures 4.6 and 4.7. Therefore, it is very important to accurately model the dissolved oxygen concentration in

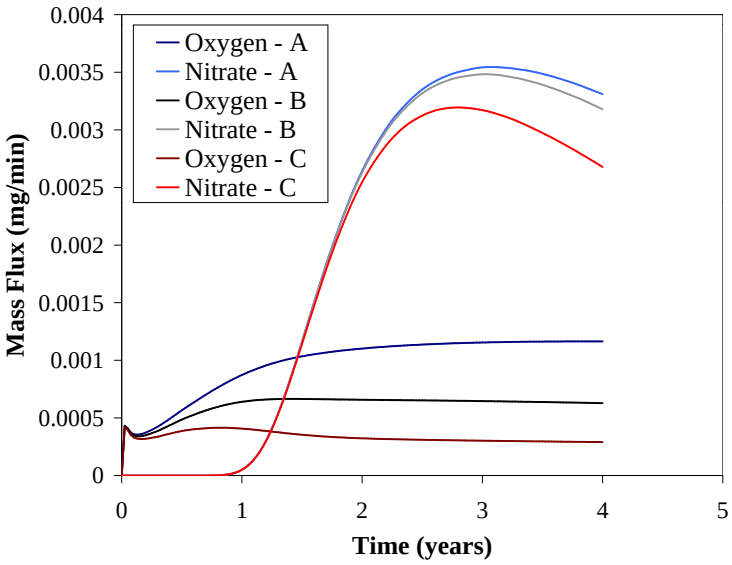

Figure 4.7: Mass breakthrough of nitrate and oxygen from the ditch resulting from reaction scenarios A, B, and C.

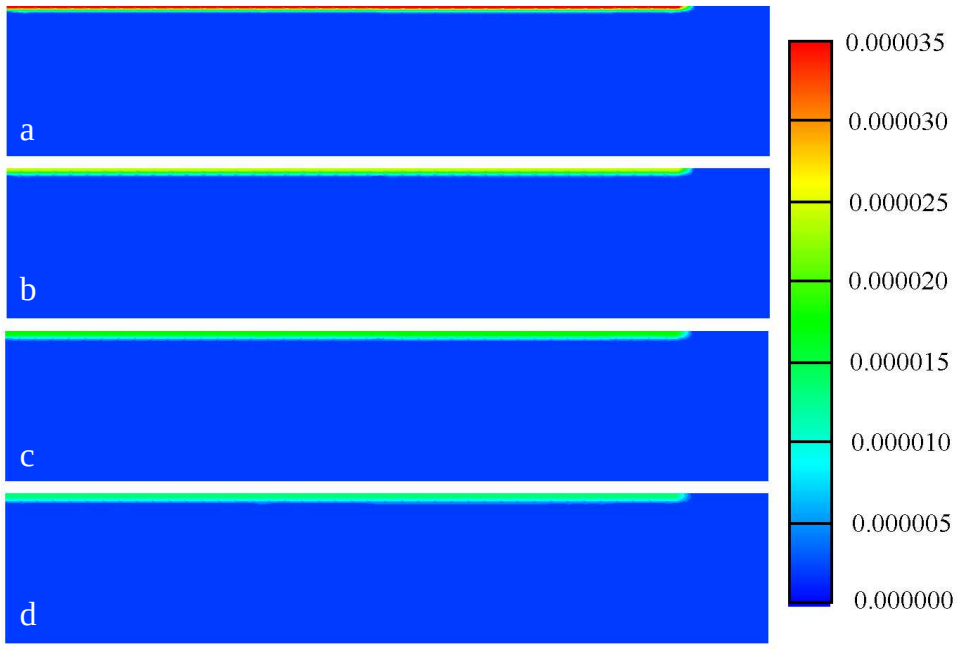

Figure 4.8: Concentration profiles of ammonium (mg/cm^3) for scenario A at (a) 3 months, (b) 6 months, (c) 9 months, and (d) 12 months. Note that the vertical axis has been exaggerated by a factor of 3.

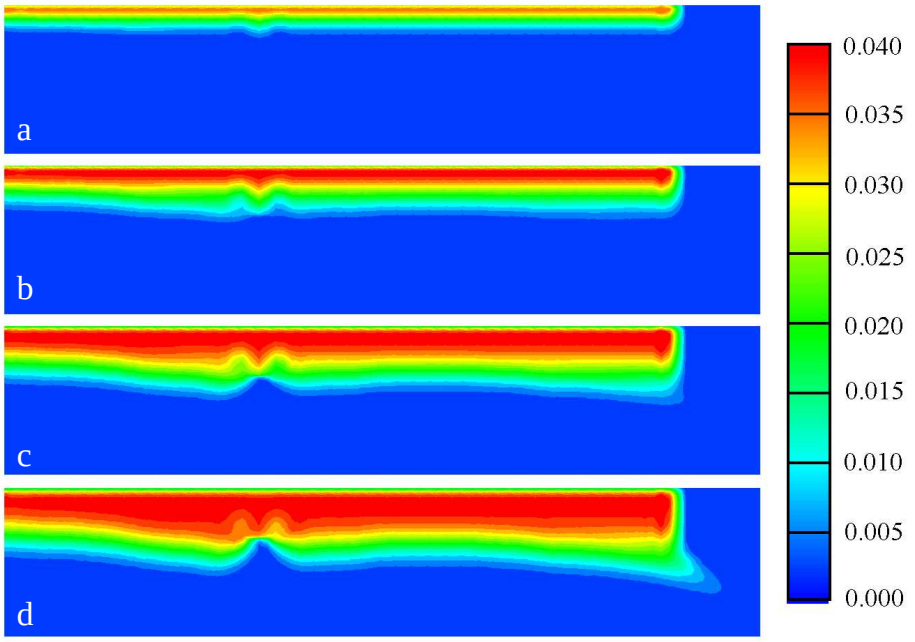

Figure 4.9: Concentration profiles of nitrate (mg/cm^3) for scenario A at (a) 3 months, (b) 6 months, (c) 9 months, and (d) 12 months. Note that the vertical axis has been exaggerated by a factor of 3.

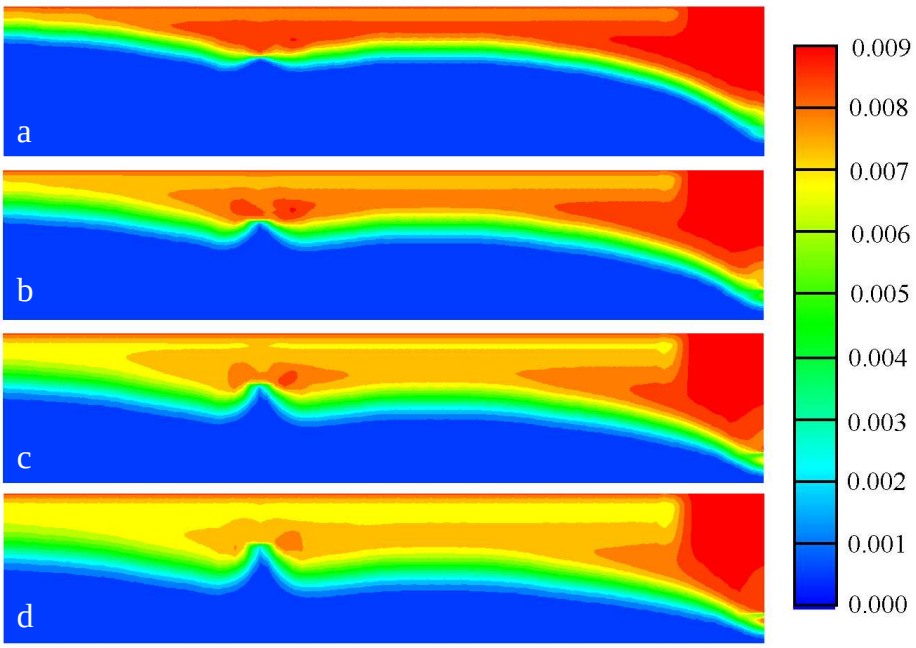

Figure 4.10: Concentration profiles of oxygen (mg/cm^3) for scenario A at (a) 3 months, (b) 6 months, (c) 9 months, and (d) 12 months. Note that the vertical axis has been exaggerated by a factor of 3.

Table 4.4: The van Genuchten parameters for the seven soil types used in transient flow scenario (*Ellsworth*, 2002).

soil layer	index	depth	θ_s	α cm^{-1}	n	K_{sat} cm/min
AP	1	0-18	0.5060	0.0070	1.5450	0.022917
A	2	18-36	0.5060	0.0080	1.5450	0.022917
BA	3	36-48	0.4810	0.0080	1.5380	0.013194
Bgl	4	48-64	0.4900	0.0080	1.5150	0.011806
Bg2	5	64-81	0.3840	0.0090	1.4660	0.006944
Bg3	6	81-99	0.4750	0.0080	1.5210	0.007639
Cg	7	99-117	0.4040	0.0060	1.6020	0.006465

the soil profile. Furthermore, it can be seen that the nitrate plume reaches the tile drain after approximately 6 months, while it takes approximately 1 year for it to reach the ditch wall. This lag is seen in the maximum nitrate breakthrough concentration as well.

Had the correct van Genuchten parameters been used, it is expected that nitrate breakthrough would have occurred much sooner, and the amount of nitrate that exited the model domain would have been larger for all scenarios, because there would have been less total saturated area in the model domain.

4.4 Transient Flow Problem

The second problem to which the OSERTM was applied was designed to demonstrate the model's ability to solve transient flow problems. This problem uses the same soil stratigraphy as shown in Figure 4.2, which is described by the parameters for the two-parameter van Genuchten model shown in Table 4.4.

Because this model has a transient saturation and flow profile, a more realistic initial condition than that used for the steady state profile was needed in order to reduce the impact of the initial condition on the flow result. To determine such an initial condition, a flow simulation was run using a flat water table 1 meter below the land surface, with hydrostatic conditions above and below, as the initial condition. This simulation had a spatial discretization of 5 centimeters in the vertical direction and 100 centimeters in the horizontal direction, and a temporal discretization of 1 minute. The simulation was run for 2999 time steps, a total of approximately 2.1 days, with no recharge. The resulting moisture profile (shown in Figure 4.11) and flow regime were used as initial conditions to the flow

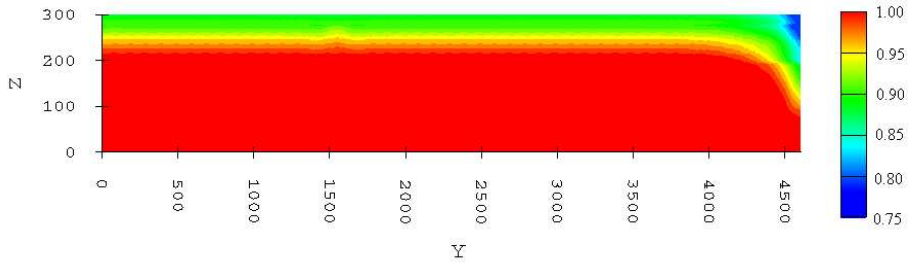

Figure 4.11: Initial saturation profile of the transient flow scenario. Note that the vertical axis has been exaggerated by a factor of 3.

model for the transient flow simulation. The initial specific discharge field was unremarkable and is not shown.

The rainfall intensity at each time step was based on rainfall records for Urbana, Illinois, which were provided by Jaswinder Singh, a graduate student in the Department of Natural Resources and Environmental Science at the University of Illinois at Urbana-Champaign. From these records, a period of approximately 40 days during the spring of 1993 was chosen. This period was chosen because it contained several rain events of varying intensity. Because the rainfall data were recorded every hour, and the time scale of the model was 1 minute, it was assumed that the rainfall intensity distribution was uniform during each hour. Using this assumption, the average rainfall rate for each time step of the simulation could be calculated by dividing the appropriate hourly rainfall figure by 60. The rainfall hydrograph for the simulation is shown in Figure 4.12.

The flow model was applied to the hypothetical cross-section using a spatial discretization of 5 centimeters in the vertical direction and 100 centimeters in the horizontal direction, and a temporal discretization of 1 minute. The simulation was run for 57,000 time steps, a total of approximately 40 days. Figure 4.13 shows the temporal distribution of water flow out of the soil profile through the drain and ditch wall. This figure shows that a rain event produces a fast response of flow into the ditch (approximately 5 minutes) and drain (approximately 15 minutes). This lag can be attributed to the length of the shortest paths that water can take to these two destinations. Additionally, the ditch has high background seepage, due to the high water table caused by the initial conditions. This seepage tends to dampen the flow response.

The hydrologic results from this flow simulation were then used in two

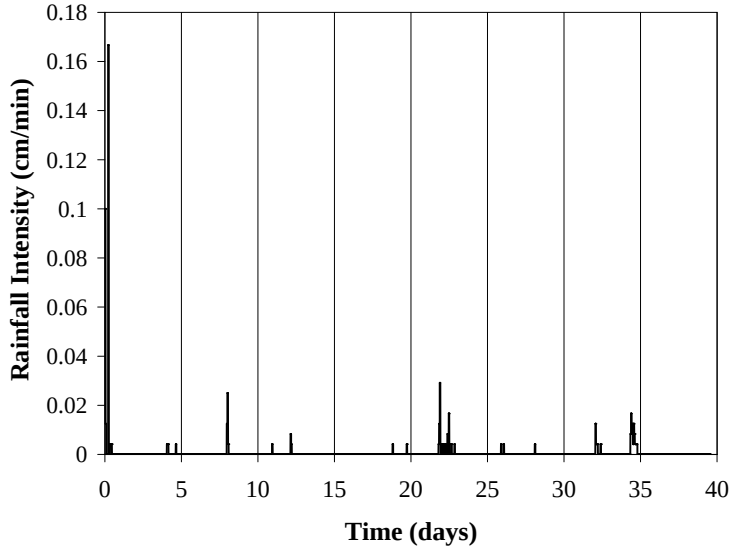

Figure 4.12: Rainfall intensity distribution used for the transient flow simulation

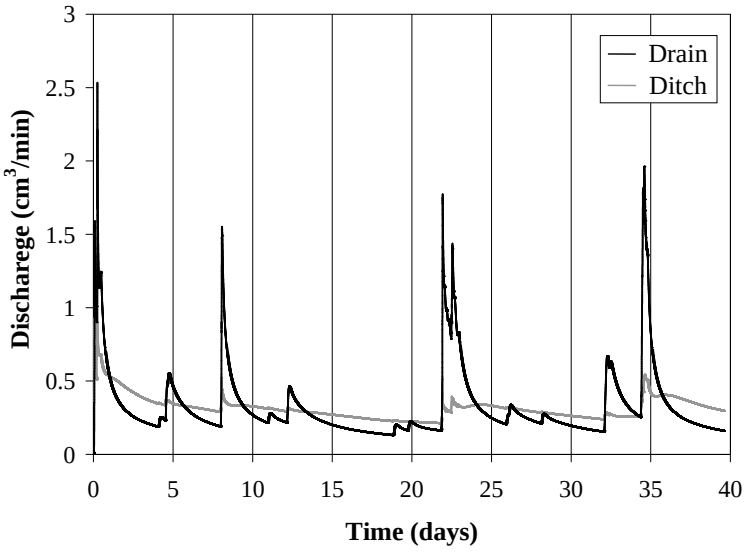

Figure 4.13: Volumetric flow rate of water out of the soil profile through the drain and ditch wall. The rainfall intensity is plotted on the upper horizontal axis in black.

separate solute transport simulations. Both of these simulations used the same spatial and temporal discretization as the flow model described above, and used longitudinal, transverse horizontal, and transverse vertical dispersivities of 10.0 cm, 1.0 cm, and 1.0 cm, respectively.

In the first simulation, a non-reactive tracer was applied at a concentration of 1 mg/cm^3 in the infiltration water across the entire land surface during the first 30 minutes of the simulation. The mass breakthrough of this tracer to the drain and ditch is shown in Figure 4.14. Though not all of the tracer left the soil profile during the 40 day period, a few generalizations can still be made about the solute travel time distribution to both the ditch and drain. First of all, it took approximately 4 days for the tracer to reach the drain, whereas the flux of tracer through the ditch wall began almost immediately. The response time of the drain to the tracer input was much longer than that of the drain to a rain event. This is because the tracer had to travel down from the soil surface before it could enter the drain (a distance of approximately 1 meter). The water, on the other hand, was able to begin to enter the drain as soon as the water table rose above the elevation of the drain. Additionally, the peaks of the mass flux from both the drain and the ditch are highly correlated with rain events. In the case of the drain, these peaks seem to be increasing, indicating that the majority of the tracer that would exit the soil profile through the drain had not yet done so. On the other hand, the mass flux peaks for the ditch wall appear to be decreasing, indicating that the majority of tracer that would exit through the ditch had already done so. Finally, the majority of the solute that left the soil profile during the first 10 days of the simulation left via the ditch, whereas for the remainder of the simulation, the drain carried the majority of the solute lost from the soil profile. It is suspected that the reason for this occurrence is that the flow paths from the capture zone of the ditch to the ditch wall are short, compared to those from the capture zone of the drain to the drain itself.

In the second of these simulations, the OSERTM was applied to a reactive transport scenario, as described in Section 4.2. In this simulation, ammonium was applied during the first 30 minutes, and oxygen consumption scenario B was used. The simulation was run for 57,000 time steps, approximately 40 days.

The breakthrough curve of nitrate to the tile drain is shown in Figure 4.15. The breakthrough curve for oxygen was typical, given the initial conditions of 8.52 mg/L in the unsaturated zone and 0 mg/L in the saturated zone. The curve showed peaks corresponding with each rain event, with no

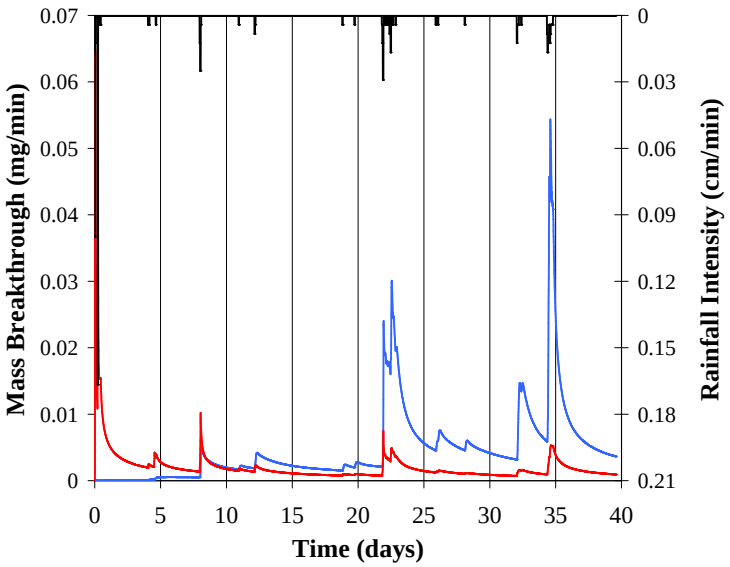

Figure 4.14: Mass breakthrough of a non-reactive tracer from the drain and ditch resulting from the transient flow scenario. The mass flux of tracer through the drain is shown in blue, while the mass flux of tracer through the ditch wall is shown in red. The rainfall intensity is plotted on the upper horizontal axis in black.

initial lag. The curve for ammonium is not plotted, because no ammonium reached the drain during the simulation. The breakthrough curve for the ditch is not shown, because no nitrogen species exited the soil profile through the ditch wall during the simulation. The concentration profiles at 10, 20, 30, and 40 days that resulted from this simulation are shown in Figures 4.16, 4.17, and 4.18 for ammonium, nitrate, and oxygen, respectively.

Similar to the steady state simulation, this simulation indicates that ammonium sorbs strongly to the soil and is converted to nitrate before it can exit the soil profile, and that there are regions with lower dissolved oxygen concentrations, in which denitrification may occur. Thus, it can be assumed that a higher rate of aerobic degradation of SOM would result in less nitrate reaching the drain. This simulation also shows that the shortest flow path nitrate can take to exit the soil profile through the ditch wall is much longer than the shortest flow path to exit through the tile drain. It takes approximately 7 days for nitrate to begin appearing in the drainage water, and approximately 20 days before large quantities of nitrate exit the drain during rain events. On the other hand it will take much longer than 40 days for nitrate to reach the ditch wall if it does in fact, reach the ditch wall at all. This result indicates that the decision not to apply fertilizer

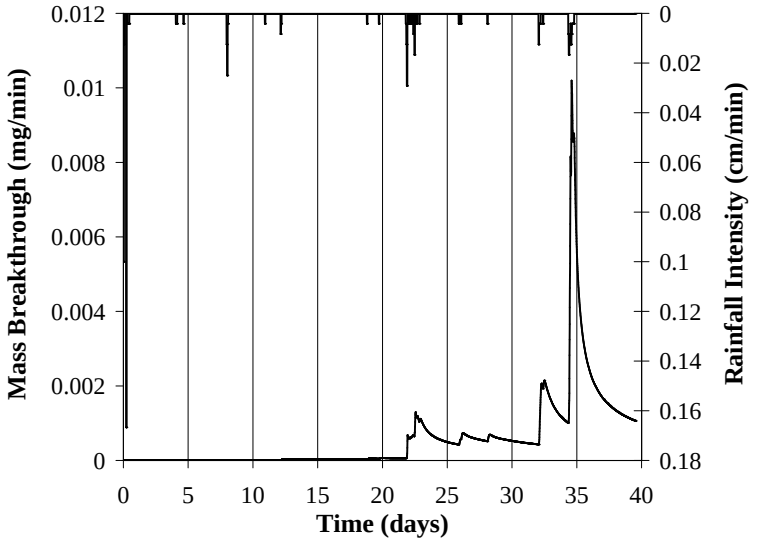

Figure 4.15: Mass breakthrough of nitrate from the drain resulting from the transient flow scenario. The mass flux of nitrate is plotted on the lower horizontal axis, while the rainfall intensity is plotted on the upper horizontal axis.

Figure 4.16: Concentration profiles of ammonium (mg/cm^3) for the transient flow scenario at (a) 10 days, (b) 20 days, (c) 30 days, and (d) 40 days. Note that the vertical axis has been exaggerated by a factor of 3.

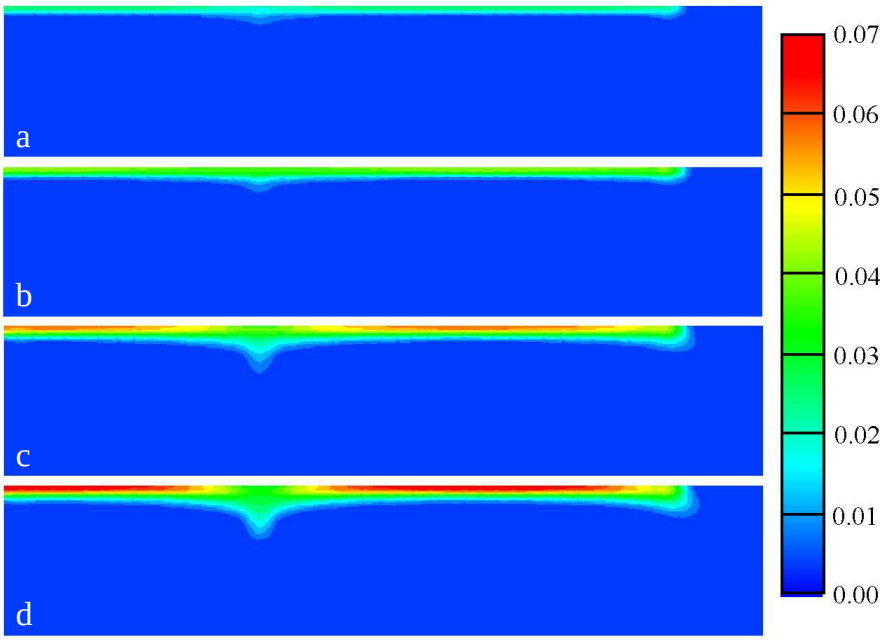

Figure 4.17: Concentration profiles of nitrate (mg/cm^3) for the transient flow scenario at (a) 10 days, (b) 20 days, (c) 30 days, and (d) 40 days. Note that the vertical axis has been exaggerated by a factor of 3.

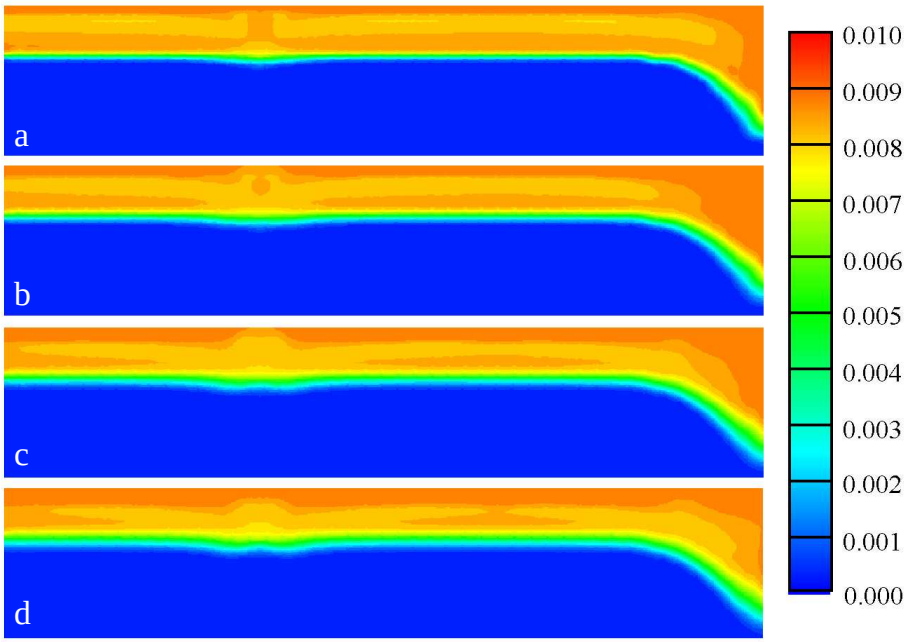

Figure 4.18: Concentration profiles of oxygen (mg/cm^3) for the transient flow scenario at (a) 10 days, (b) 20 days, (c) 30 days, and (d) 40 days. Note that the vertical axis has been exaggerated by a factor of 3.

within 5 meters of the ditch was a good management strategy to reduce the amount of nitrogen exiting the soil profile. If the buffer zone had not existed, the results of the tracer simulation suggest that nitrate would have exited through the ditch wall in large quantities early on in the simulation, while the amount that exited the drain would have remained unchanged. By increasing the residence time of the nitrate in the soil, there is a greater chance for denitrification to occur, thus reducing the total amount of nitrate that will exit the soil profile.

One further observation from this simulation is that nitrogen export through the tile line is highly correlated with rain events. This pattern has also been observed in the field (*Gentry et al.*, 1998), indicating that the OSERTM can reproduce conditions consistent with field observations.

4.5 Summary

In this chapter, the OSERTM is applied to several solute transport problems, which include both non-reactive and reactive solutes, and both steady and transient flow conditions. Though the results presented here are limited to a two-dimensional soil profile with idealized geometry, analysis of results of these applications indicate that the OSERTM provides valid solutions to solute transport problems under conditions consistent with those found on agricultural fields in Central Illinois. Further, these results begin to illustrate the usefulness of the OSERTM for making nitrogen management decisions and quantifying nitrogen losses from agricultural fields. In the next chapter, some recommendations will be given to indicate the clearest path toward preparing the OSERTM for field-scale applications.

Chapter 5

Conclusions and Recommendations

This research has shown that the OSERTM provides a viable solution to reactive transport in all three spatial dimensions, with transient, non-uniform, multi-dimensional flow in both the saturated and unsaturated zones. Admittedly, there are some drawbacks, in the form of memory and execution time, to using the OSERTM rather than one of the other models described in Chapter 1. With some refinement, however, the OSERTM may provide a more powerful alternative to these models.

The following sections will outline some suggestions for future research that may increase the accuracy, efficiency, and applicability of the OSERTM.

5.1 Spatial Discretization

As mentioned in Section 2.2, the OSERTM discretizes the physical domain using a grid that has a uniform cell length and number of nodes in each of the three spatial dimensions. A uniform grid was used in order to simplify the calculations. This allowed the spatial discretization and the number of nodes in each of the cardinal directions to be chosen based on the uniform temporal discretization and the flow field at the beginning of the simulation. This discretization remained constant in space and time. The most notable deficiency of this approach is that it renders the model unable to replicate the true geometry of the model domain. The most obvious example of this problem can be seen in the ditch wall, in the simulations described in Chapter 4. The wall was modeled as being vertical, and as intersecting the land surface at a right angle. Obviously, this is not the true physical geometry of the land surface. While there is not a great deal of relief in Central Illinois, it is still important to be able to model variations in the

topography of the physical domain as accurately as possible.

Another, less obvious, weakness of this approach is that it may require a larger number of nodes than is necessary for a particular level of accuracy. In non-uniform, multi-dimensional flow fields, such as those considered in this research, the magnitude and direction of the specific discharge vectors may vary greatly both within and between different regions of the model domain. For example, Figure 4.4 shows that the direction of the specific discharge vectors in the region around the tile drain varies from almost horizontal, to nearly vertical, depending on the location of the vectors relative to the drain, and that the magnitude of these vectors is much higher than it is in the rest of the model domain, except near the bottom of the ditch wall. On the other hand, the specific discharge in the far left region of the model domain is moving nearly uniformly downward, with an almost uniform magnitude. Because numerical errors occur more readily in regions where the flow is changing magnitude and direction than in regions where it is nearly uniform, it is important to use a fine grid in the more variable regions (*Chilakapati and Yabusaki*, 1999). In regions of nearly uniform flow, however, it is unnecessary, and even wasteful to use such a fine grid, because little accuracy is gained from the higher resolution. For this reason, it has been suggested that a non-uniform grid that is finer in regions where the flow is highly variable, and coarser elsewhere, should be used (*Bear et al.*, 1998; *Wang et al.*, 1998). By employing a non-uniform grid for use on a specific problem, it is possible to reduce the number of nodes, while maintaining or increasing the model's accuracy, thus resulting in a lower computational demand for a fixed level of accuracy.

It is also important to consider that during transient flow conditions, these "variable" regions actually experience highly non-uniform flow conditions only temporarily. For this reason, in addition to the spatial discretization of the OSERTM being non-uniform, it may also be useful for it to be able to change with time. This idea will be revisited in the next section, but before moving on, it is important to mention that the conjunctive flow model must also be able to support any modification of the spatial discretization.

5.2 Temporal Discretization

The temporal domain is also currently discretized uniformly. However, as with the spatial discretization, it is conceivable that this uniform discretization may be excessively fine in certain regions of the temporal domain, and too coarse in others. For example, during a rainstorm, it is important to

capture the temporal rainfall distribution and subsurface flow variability. For this purpose, a fine temporal discretization is necessary. However, during dry spells, the subsurface flow changes very slowly with time, so a fine temporal discretization is unnecessary. For this reason, an adaptive temporal discretization would be ideal. Such a method would refine the temporal discretization just before a rainstorm and coarsen it after the subsurface flow ceased to change rapidly with time, thus allowing the model to move quickly from one rain event to the next. This would allow the majority of the computational effort to be spent solving solute transport during high flow events.

The temporal discretization, however, is dependent on the spatial discretization and the flow characteristics at specific points within the model domain, as shown in Equations (2.5) and (2.6). It may be noted that Equation (2.5) contains the coefficient of dispersivity as a parameter, but it should be recalled that this parameter is a function of the flow field according to Equations (2.19), (2.20), (2.21), (2.23), and (2.24). Therefore, any refinement of the temporal discretization may affect the stability of the model for a particular spatial discretization. For this reason, a fully adaptive time and space discretization may be the best option. Such a method would calculate the best tradeoff between time and space discretization based on a particular flow instance, in order to minimize the computational demand for a particular level of accuracy. In such a method, both the spatial and temporal discretizations would change with time during a simulation.

Once again, however, it is necessary to note that the conjunctive flow model must also be able to support any modification of the temporal discretization.

5.3 Numerical Methods

Because, as mentioned above, it is important to evaluate the benefits of a non-uniform spatial discretization, it should be noted that the third-order accuracy of Leonard's ULTIMATE TVD method has been questioned when applied to a non-uniform grid (*Liu et al.*, 1995). However, the authors provide a modified form of the TVD method, which should be considered if a non-uniform grid is implemented in the OSERTM (*Liu et al.*, 1995).

Furthermore, because several researchers have noted the inaccuracy of operator splitting (OS), and because the OSERTM currently uses a very simple splitting algorithm with a low order of accuracy, the benefit of higher-order splitting algorithms should be evaluated. The Strang splitting algo-

rithm (*Strang*, 1963; *Valocchi and Malmstead*, 1992) is one of the most commonly used splitting algorithms, and it has been proven to have a higher level of accuracy than normal time splitting. More recently, however, several researchers have presented other splitting algorithms of higher order. The most notable of these are described below.

Ding and Liu (1989) describe an OS algorithm that performs well for two-dimensional advection-dispersion-reaction problems, where all three operations were solved independently. The authors note that this algorithm is similar to that of Holly and Preissmann (1977). In another paper, the authors describe a modified version of this algorithm and apply it to three-dimensional advection-dispersion problems (*Ding and Liu*, 1993).

Khan and Liu (1995) developed a second-order accurate OS algorithm adapted from that of Holly and Preissmann. This algorithm's performance is described when applied to two- and three-dimensional advection-dispersion problems (see Khan and Liu 1998a and Kahn and Liu 1998b, respectively). Kahn and Liu also developed an algorithm based on the Strang method (*Kahn and Liu*, 1995). This algorithm proved to be second order accurate when applied to one-dimensional advection-dispersion-reaction problems, where all three operations were solved independently.

5.4 Physical Processes

The soil in the unsaturated zone often contains fractures, fissures, worm holes, large-scale textural variations, and inter-aggregate pore spaces, generally referred to as "macropores" (*Ray*, 1994). These macropores have been cited as the cause of sharp increases in tile drain flow and its solute concentration during heavy irrigation or rainfall events (*Mohanty et al.*, 1998). In fact, in a recent study of nitrogen loss from agricultural fields in Illinois, Gentry *et al.* (2000) noted that flow in Drummer soil is controlled by preferential flow through macropores. For this reason, in the future, it is important for the OSERTM to address the role of macropores in the transport of nitrogen. For an overview of the many methods available for modeling flow and transport in macroporous media, see Mills *et al.* (1991), Gerke and van Genuchten (1993), and Ray (1994). Once again it is important to note that any consideration of macropores by the OSERTM would also have to be supported by the conjunctive flow model.

5.5 Biochemical Processes

The reaction system shown in Equations (2.41), (2.42), and (2.43), treat the fate of ammonia fertilizer somewhat like that of a reactive tracer. In reality, however, the biogeochemical cycling of nitrogen in soil is considerably more complex. A variety of more complex biogeochemical models could be implemented in the OSERTM, depending on whether management or research questions were being addressed.

For management purposes, a nitrogen cycle comparable to the newest version of DRAINMOD-N (*Brevé et al.*, 1994) will be sufficient. This can be implemented by adding the release of ammonia from the mineralization of soil organic matter and the uptake of nitrate by crops to the OSERTM. Such a model may prove very useful in further investigations of field studies.

For more in-depth research, immobilization of ammonium in soil organic matter, transport of dissolved organic nitrogen, and the formation of the nitrite intermediate in denitrification are the next processes to add to the OSERTM. It may also be worth examining whether complex models of soil biogeochemistry (such as CENTURY (*Parton et al.*, 1992)), which divide soil organic matter into multiple fractions, can be implemented in this framework as well. While such models seem complex, their implementation in the OSERTM may be able to be simplified because they emphasize transformations among immobile components. This would allow interesting studies to be conducted in order to assess the degree to which the more complex model accurately describes field data.

5.6 Interaction of the OSERTM with the Conjunctive Flow Model

Currently, the OSERTM reads in hydrologic data on an "as needed basis" from output files created by the conjunctive flow model. For example, at each time step, the OSERTM reads the moisture content data for that time step into memory and deletes from memory any unneeded moisture content data from the previous time step. This method of providing hydrologic data to the OSERTM requires the use of enormous data files, as well as a lot of hard disk access, a time-consuming process. For example the specific discharge data file for the transient simulation described in Section 4.4 was over ten gigabytes large. Certainly some benefit could be gained by using binary, rather than ASCII formatted data, but even so, the hydrologic data for a single year would still be much too large to store on the hard disk of

a personal computer. For this reason, the two models should be adapted to work either in a "macroscale" batch environment or in a fully dynamic environment. In a macroscale batch environment, the problem would be divided into macroscale time intervals each consisting of a finite number of time steps. The flow model would produce the hydrologic data for a time interval and write it to the hard disk. Then, the OSERTM would solve the solute transport just as before, using disk access to read in the data on an as needed basis. At the end of the solute transport simulation, the flow model would read in the hydrologic data for the last time step, and then begin a new simulation for the next macroscale time interval. When this flow simulation was complete, the OSERTM would read in the final solute concentration distributions as initial conditions and begin the solute transport simulation for the macroscale time interval. This process would continue until all the macroscale time intervals, and thus all the time steps, had been simulated. The advantage of this method is that the data files could be kept to a manageable size.

The fully dynamic environment is really just a special case of the macroscale batch environment, where each macroscale time interval consists of just one time step. This method has the added benefit of the data files possibly being small enough to be kept in memory, thus eliminating the need for slow hard disk access. This would dramatically increase the speed of OSERTM/flow model interaction. For this reason, this method is the most efficient, and thus is the best option for implementation of the OSERTM.

References

[1] Ahuja, L.R., K.W. Rojas, J.D. Hanson, M.J. Shaffer, and L. Ma (eds.). 1999. *Root Zone Water Quality Model: Modeling Management Effects on Water Quality and Crop Production.* Colorado, Water Resources Publications, LLC.

[2] Bachmat, Y., and G. Chetboun. 1976. A Mathematical Model for Predicting the Concentration of Nitrogen Compounds in Surface and Groundwater Streams, Hydro. Report No. 1976/4. Israel: Hydrological Service, Ministry of Agriculture Water Commission.

[3] Barakat, H.Z., and J.A. Clark. 1966. On the Solution of the Diffusion Equation by Numerical Methods. *Journal of Heat Transfer*, *88*: 421-427.

[4] Bear, J., F. Wang, and A. Shaviv. 1998. An N-Dynamics Model for Predicting N-Behavior Subject to Environmentally Friendly Fertilization Practices: I-Mathematical Model. *Transport in Porous Media*, *31*: 249-274.

[5] Bouwer, H., and J. van Schilfgaarde. 1963. Simplified Method of Predicting Fall of Water Table in Drained Land. *Transactions of the ASAE*, *6*(4): 288-291.

[6] Brevé, M.A., R.W. Skaggs, J.E. Parsons, and J.W. Gilliam. 1997. DRAINMOD-N, A Nitrogen Model for Artificially Drained Soils. *Transactions of the ASAE*, *40*(4): 1067-1075.

[7] Brian, P.L.T. 1961. A Finite-Difference Method of High-Order Accuracy for the Solution of Three-Dimensional Transient Heat Conduction Problems. *American Institute of Chemical Engineering Journal*, *7*(3): 367-370.

[8] Brooks, R.H., and A.T. Corey. 1964. Hydraulic Properties of Porous Media, Hydrology Paper No. 3. Fort Collins, CO: Colorado State University.

[9] Brown, P.N., G.D. Byrne, and A.C. Hindmarsh. 1989. VODE, A Variable-Coefficient ODE Solver. *SIAM Journal on Scientific and Statistical Computing*, *10*: 1038-1051.

[10] Broughton, R.S., and J.L. Fouss. 1999. Subsurface Drainage Installation Machinery and Methods. In *Agronomy A Series of Monographs: No. 38 Agricultural Drainage*, ed. R.W. Skaggs and J. van Schilfgaarde. Madison, WI: American Society of Agronomy, Inc.; Crop Science Society of America, Inc.; Soil Science Society of America, Inc.: 963-1003.

[11] Burnet, R.D., and E.O. Frind. 1987. Simulation of Contaminant Transport in Three Dimensions 2. Dimensionality Effects. *Water Resources Research*, *23*(4): 695-705.

[12] Cabrera, M.L., and D.E. Kissel. 1984. Rate of Urea Hydrolysis in Soil as Affected by Urea Concentration. *Agronomy Abstracts*, American Society of Agronomy, Madison, WI: 268.

[13] Carpenter, S.R., N.E. Caraco, D.L. Corell, R.W. Howarth, A.N. Sharpley, and V.H. Smith. 1998. Nonpoint pollution of surface waters with phosphorous and nitrogen. *Ecological Applications*, *8*: 559-568.

[14] Celia, M.A., E.T. Bouloutas, and R.L. Zabra. 1990. A General Mass Conservative Numerical Solution for the Unsaturated Flow Equation. *Water Resources Research*, *26*: 1483-1496.

[15] Charbeneau, R.J. 2000. *Groundwater Hydraulics and Pollutant Transport*. New Jersey: Prentice Hall.

[16] Chen, Y.M., L.M. Abriola, P.J.J. Alvarez, P.J. Anid, and T.M. Vogel. 1992. Modeling Transport and Biodegradation of Benzene and Toluene in Sandy Aquifer Material: Comparisons With Experimental Measurements. *Water Resources Research*, *28*(7): 1833-1847.

[17] Chilakapati, A., and S. Yabusaki. 1999. Nonlinear Reactions and Nonuniform Flows. *Water Resources Research, 35*(8): 2427-2438.

[18] Chilakapati, A., S. Yabusaki, J. Szecsody, and W. MacEvoy. 2000. Groundwater Flow, Multicomponent Transport and Biogeochemistry: Development and Application of a Coupled Process Model. *Journal of Contaminant Hydrogeology*, *43*:303-325.

[19] Cooke, R.A., Ph.D. 2001a. Personal communication. Urbana, IL, 25 July 2001.

[20] Cooke, R.A., Ph.D. 2001b. Personal communication. Urbana, IL, 07 Aug. 2001.

[21] Cooley, R.L. 1983. Some New Procedures for Numerical Modeling of Variably Saturated Flow Problems. *Water Resources Research*, *19*(5): 1271-1285.

[22] David, M.B., and L.E. Gentry. 2000. Anthropogenic Inputs of Nitrogen and Phosphorous and Riverine Export for Illinois, USA. *Journal of Environmental Quality*, *29*: 494-508.

[23] David, M.B., L.E. Gentry, D.A. Kovacic, and K.M Smith. 1997. Nitrogen Balance in and Export from an Agricultural Watershed. *Journal of Environmental Quality*, *26*: 1038-1048.

[24] Ding, D., and P.L.-F. Liu. 1989. An Operator Splitting Algorithm for Two-Dimensional Convection-Dispersion-Reaction Problems. *International Journal for Numerical Methods in Engineering, 28*: 1023-1040.

[25] Ding, D., and P.L.-F. Liu. 1993. An Operator Splitting Algorithm for Three-Dimensional Convection-Diffusion Problems. *Journal of Hydrodynamics, B*: 22-34.

[26] Ellsworth, T.R., Ph.D. 2002. Personal communication. Urbana, IL, 08 Feb. 2002.

[27] Essaid, H.I., B.A. Bekins, E.M. Godsy, E. Warren, M.J. Baedecker, and M.J. Cozzarelli. 1995. Simulation of Aerobic and Anaerobic Biodegradation Processes at a Crude Oil Spill Site. *Water Resources Research*, *31*(12): 3309-3327.

[28] Faussey, N.R., L.C. Brown, H.W. Belcher, and R.S. Kanwar. 1996. Drainage and Water Quality in Great Lakes and Cornbelt States. *Journal of Irrigation and Drainage Engineering*, *121*: 283-288.

[29] Feng, S., and B. Bar-Yosef. 1995. Simulation of Long Term Nitrogen Behavior in Agricultural Soils, Report on Cooperative Project. Bet-Degan, Israel: Agricultural Research Organization.

[30] Gentry, L.E., M.B. David, K.M. Smith, and D.A. Kovacic. 1998. Nitrogen Cycling and Tile Drainage Nitrate Loss in a Corn/Soybean Watershed. *Agriculture, Ecosystems and Environment*, *68*: 85-97.

[31] Gentry, L.E., M.B. David, K.M. Smith-Starks, and D.A. Kovacic. 2000. Nitrogen Fertilizer and Herbicide Transport from Tile Drained Fields. *Journal of Enviromental Quality*, *29*(1): 232-240.

[32] Gerke, H.H., and M.T. van Genuchten. 1993. A Dual-Porosity Model for Simulating Preferential Movement of Water and Solutes in Structured Porous Meida. *Water Resources Research*, *29*: 305-319.

[33] Goolsby, D.A, W.A. Battaglin, G.B. Lawrence, R.S. Artz, B.T. Aulenbach, R.P. Hooper, D.R, Keeney, and G.J. Stensland. 1999. Flux and Sources of Nutrients in the Mississippi-Atchafalaya River Basin: Topic 3 Report for the Integrated Assessment on Hypoxia in the Gulf of Mexico, NOAA Costal Ocean Program Decision Analysis Series No. 17. Silver Spring, MD: NOAA Coastal Ocean Program.

[34] Gusman, A.J., and M.A. Mariño. 1999. Analytical Modeling of Nitrogen Dynamics in Soils and Ground Water. *Journal of Irrigation and Drainage Engineering*, *125*(6): 330-337.

[35] Harten, A. 1983. High Resolution Schemes for Hyperbolic Conservation Laws. *Journal of Computational Physics*, *49*: 357-393.

[36] Holly, F.M., Jr., and A. Preissmann. 1977. Accurate Calculation of Transport in Two-Dimensions. *Journal of Hydraulic Engineering, 103*: 1259-1277.

[37] Hudson, R.J., Ph.D. 2001a. Personal communication. Urbana, IL, 20 Feb. 2001.

[38] Hudson, R.J., Ph.D. 2001b. Personal communication. Urbana, IL, 19 Sept. 2001.

[39] Hudson, R.J., Ph.D. 2001c. Personal communication. Urbana, IL, 21 Oct. 2001.

[40] Hudson, R.J., Ph.D. 2001d. Personal communication. Urbana, IL, Nov. 2001.

[41] Huyakorn, P.S., S.D. Thomas, and B.M. Thompson. 1984. Techniques for Making Finite Elements Competitive in Modeling Flow in Variably Saturated Porous Media. *Water Resources Research*, *20*(8): 1099-1115.

[42] Johnsson, H., L. Bergstrom, and P.E. Jansson. 1987. Simulated Nitrogen Dynamics and Losses in a layered Agricultural Soil. *Agricultural, Ecosystems and Environment*, *18*: 333-356.

[43] Kalita, P. 1999. Transient Finite Element Method Solution of Oxygen Diffusion in Soil. *Ecological Modeling, 118*: 227-236.

[44] Kahn, L.A., and P.L.-F. Liu. 1995. An Operator Splitting Algorithm for coupled One-Dimensional Advection-Diffusion-Reaction Equations. *Computer Methods in Applied Mechanics and Engineering, 127*: 181-201.

[45] Kahn, L.A., and P.L.-F. Liu. 1998a. Numerical Analyses of Operator-Splitting Algorithms for the Two-Dimensional Advection-Diffusion Equation. *Computer Methods in Applied Mechanics and Engineering, 152*: 337-359.

[46] Kahn, L.A., and P.L.-F. Liu. 1998b. An Operator Splitting Algorithm for the Three-Dimensional Advection-Diffusion Equation. *International Journal for Numerical Methods in Fluids, 28*: 461-476.

[47] Kinzelbach, W.K.H., P.J. Dillon, and K.H. Jensen. 1990. State of the Art of Existing Numerical Groundwater Quality Models of the Saturated Zone and Experience with their Application in Agricultural Problems, Report No. ARS-81. In *Proceedings of the International Symposium on Water Quality Modeling of Agricultural Non-Point Sources, Part I*, ed. D.D. DeCoursey. Beltsville, MD: Agricultural Research Service, U.S. Department of Agriculture: 307-325.

[48] Kinzelbach, W., W. Schafer, and J. Herzer. 1991. Numerical Modeling of Natural and Enhanced Denitrification Processes in Aquifers. *Water Resources Research, 27*(6): 1123-1135

[49] Knisel, W.G. (ed.). 1980. CREAMS: a Field-Scale Model for Chemicals, Runoff, and Erosion from Agricultural Management Systems, Conservation Research Report No. 26. Washington, DC: U.S. Department of Agriculture.

[50] Knisel, W.G. (ed.). 1993. GLEAMS: Groundwater Loadings Effects of Agricultural Management Systems. Tifton, GA: Biological and Agricultural Engineering Department, Coastal Plain Experiment Station, University of Georga.

[51] Korom, Scott F. 1992. Natural Denitrification in the Saturated Zone: A Review. *Water Resources Research, 28*(6): 1657-1668.

[52] Leij, F.J., and S.A. Bradford. 1994. 3DADE: A Computer Program for Evaluating Three-Dimensional Equilibrium Solute Transport in Porous Media, Research Report No. 134. Riverside, CA: U.S. Salinity Laboratory Agricultural Research Service, U.S Department of Agriculture.

[53] Leonard, B.P. 1979. A Stable and Accurate Convective Modeling Procedure Based on Quadratic Upstream Interpolation. *Computer Methods in Applied Mechanics and Engineering*, *19*: 59-98.

[54] Liu, J., G.A. Pope, and K. Sepehrnoori. 1995. A High-Resolution Finite-Difference Scheme for Non-Uniform Grids. *Applied Mathematical Modelling*, *19*: 162-172.

[55] Leonard, B.P. 1991. The ULTIMATE Conservative Difference Scheme Applied to Unsteady One-Dimensional Advection. *Computer Methods in Applied Mechanics and Engineering*, *88*: 17-74.

[56] Logan, T.J., D.J. Eckert, and D.G. Beak. 1991. Tillage, Crop and Climatic Effects on Runoff and Tile Drainage Losses of Nitrate and Four Herbicides. *Soil Tillage Research*, *30*: 75-103.

[57] MacQuarrie, K.T.B., E.A. Sudicky, and E.O. Frind. 1990. Simulation of Biodegradable Organic Contaminants in Groundwater: 1. Numerical Formulation in Principal Directions. *Water Resources Research*, *26*(2): 207-222.

[58] MacQuarrie, K.T.B, and E.A. Sudicky. 2001. Multicomponent Simulation of Wastewater-Derived Nitrogen and Carbon in Shallow Unconfined Aquifers I. Model Formulation and Performance. *Journal of Contaminant Hydrology*, *47*: 53-84.

[59] Marchetti, R., M. Donatelli, and P. Spallacci. 1997. Testing Denitrification Functions of Dynamic Crop Models. *Journal of Environmental Quality*, *26*: 394-401.

[60] McDonald, M.G., and A.W. Harbaugh. 1988. A Modular Three-Dimensional Finite-Difference Ground-Water Flow Model. *Techniques of Water Resources Investigations of the U.S. Geological Survey, Book 6.*

[61] Millington, R.J., and J.P. Quirk. 1961. Permeability of Porous Solids. *Transactions of the Faraday Society*, *15*: 1200-1207.

[62] Mills, W.B., R.A. Johns, C. White, B. Lester, and D.S. Ward. 1991. Review and Evaluation of Approaches to Simulate Flow and Solute Transport through Macroporous Media, Tetra Tech, Inc. Lafayette, CA.

[63] Mitchell, A.R. 1984. Recent Developments in the Finite Element Method. In *Computational Techniques and Applications: CTAC-83*, ed. J. Noye and C.A.J Fletcher. North-Holland, Elsevier.

[64] Mohanty, B.P., R.S. Bowman, J.M.H. Hendrickx, J. Simunek, and M.T. van Genuchten. 1998. Preferential Transport of Nitrate to a Tile Drain in an Intermittent-Flood-Irrigated Field: Model Development and Experimental Evaluation. *Water Resources Research*, *34*(5): 1061-1076.

[65] Morita, M., and B.C. Yen. 2000. Numerical Methods for Conjunctive Two-Dimensional Surface and Three-Dimensional Sub-Surface Flows. *International Journal for Numerical Methods in Fluids*, *32*: 921-957.

[66] Morita, M., and B.C. Yen. 2000. Modeling of Conjunctive Two-Dimensional Surface-Three-Dimensional Subsurface Flows. *Journal of Hydraulic Engineering*, *128*(2): 184-200.

[67] Neden, S. 1990. *Effect of High Ammonium Concentration on Nitrification Rate* (Hebrew), M.S. thesis. Israel: Faculty of Agricultural Engineering, Technion - IIT.

[68] Parton, W.J., B. McKeown, V. Kirchner, and D.S. Ojima. 1992. CENTURY Users Manual, NREL Publication. Fort Collins, CO: Colorado State University.

[69] Rabalais, N.N., R.E. Turner, W.J. Wiseman, Jr., and D.F. Boesch. 1991. A Brief Summary of Hypoxia on the Northen Gulf of Mexico Continental Shelf: 1985-1988. In *Modern and Ancient Continental Shelf Anoxia: Special Publication 58* ed. R.V. Tyson and T.H. Pearson, p. 35-47. Geological Society, London.

[70] Ray, C. 1994. *Modeling Transport of Agricultural chemicals in a Dual Porosity System Resulting from Macropores*, Ph.D. dissertation, University of Illinois, Urbana-Champaign.

[71] Roe P.L. 1984. Generalized Formulation of TVD Lax-Wendroff Schemes, ICASE Report 84-53, NASA CR-172478. NASA Langley Research Center.

[72] Shaffer, M.J., A.D. Halvorson, and F.J. Pierce. 1991. Nitrate Leaching and Economic Analysis Package (NLEAP): Model Description and Application. In *Managing Nitrogen for Groundwater Quality and Farm Profitability*, eds. R.F. Follet, D.R. Keeney, and R.M. Cruse. Madison, WI: SSSA, Inc.: 285-322.

[73] Skaggs, R.W. 1980. A Water Management Model for Artificially Drained Soils, Technical Bulletin No. 267. Raleigh, NC: North Carolina Agricultural Research Service, North Carolina State University.

[74] Strang, G. 1968. On the Construction and Comparison of Difference Schemes. *SIAM Journal of Numerical Analysis, 5*(3): 506-517.

[75] Thibodeaux, L.J. 1979. *Chemodynamics: Environmental Movement of Chemicals in Air, Water, and Soil.* New York: Wiley.

[76] Tindall, J.A., and J.R. Kunkel. 1999. *Unsaturated Zone Hydrology for Scientists and Engineers.* New Jersey: Prentice Hall.

[77] Turner, R.E., N. Qureshi, N.N. Rabalais, Q. Dortch, D. Justice, R.F. Shaw, and J. Cope. 1998. Accuracy of Operator Splitting for Advection-Dispersion-Reaction Problems. *Water Resources Research, 28*(5): 1471-1476.

[78] Valocchi, A.J., and M. Malmstead. 1992. Accuracy of Operator Splitting for Advection-Dispersion-Reaction Problems. *Water Resources Research, 28*(5): 1471-1476.

[79] Valocchi, A.J., C.J. Werth, J.J. Decker, and G. Hammond. 2001. *Interactive Models for Groundwater Flow and Solute Transport.* Retrieved April 19, 2002 from University of Illinois, Department of Civil and Environmental Engineering web site: http:// www.cee.uiuc.edu/transport/models.html

[80] VanderKwaak, J.E., P.A. Forsyth, K.T.B. MacQuarrie, and E.A. Sudicky. 1995. WATSOLV Sparse Matrix Iterative Solver Package: User's Guide Version 1.01. Waterloo Ontario: Waterloo Centre for Groundwater Research,University of Waterloo.

[81] van Genuchten, M.T. 1980. A Closed Form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils. *Soil Science Society of America Journal, 44*: 892-898.

[82] Van Leer, B. 1974. Towards the Ultimate Conservative Difference Scheme:II. Monotonicity and Conservation Combined in a Second Order Scheme. *Journal of Computational Physics*, *14*:361-370.

[83] Vitousek, P.M., D. Aber, R.W. Howarth, G.E. Likens, R.A. Matson, D.W. Schindler, W.H. Schlesinger, and D.G. Tilaman. 1997. Human Alteration of the Global Nitrogen Cycle: Sources and Consequences. *Ecological Applications*, *7*(3): 737-750.

[84] Vogel, T., and M. Cislerova. 1988. On the Reliability of Unsaturated Hydraulic Conductivity Calculated from the Moisture Retention Curve. *Transport in Porous Media*, *3*(1): 1-15.

[85] Ward M.H., S.D. Mark, K.P. Cantor, D.D. Weisenburger, A. Correa-Villaseñor, and S.H. Zahm. 1996. Drinking Water Nitrate and the Risk of non-Hodgkin's Lymphoma. *Epidemiology*, *7*: 465-471.

[86] Wang, F., J. Bear, and A. Shaviv. 1998. A N-Dynamics Model for Predicting N-Behavior Subject to Environmentally Friendly Fertilization Practices: II-Numerical Model and Model Validation. *Transport in Porous Media*, *33*: 309-324.

[87] Zheng, C., and G.D. Bennett. 1995. *Applied Contaminant Transport Modeling: Theory and Practice*. New York: Van Nostrand Reinhold.

[88] Zheng, C., and P.P. Wang. 1999. MT3DMS: A Modular Three-Dimensional Multi-Species Transport Model for Simulation of Advection, Dispersion and Chemical Reactions of Contaminants in Groundwater Systems; Documentation and User's guide, U.S. Army Engineer Research and Development Center Contract Report SERDP-99-1. Vicksburg, MS.

Appendix A

Input Files

Before detailing the format of the input files, it is important to note that the OSRTM does not specify a system of units a priori. For this reason, any units for length, mass, and time can be used; however, this functionality comes at the price of requiring the user to be conscious of unit consistency. For example, if decimeters, grams, and minutes are chosen for the units of length, mass, and time, then concentration must be specified in g/dm^2, which is equivalent to g/liter.

A.1 PARAM

The `PARAM` input file is used by the OSERTM to specify model parameters such as time and space discretization sizes, number of chemicals, etc. This input file uses the Fortran 90 namelist construct to specify parameter assignments. The first line of the file must contain the text `&PARAMETERS` and end with a forward slash (/) character. Also, a comma must end each assignment statement. An example of this input file is shown in Figure A.1. In this figure, data types have been substituted for parameter values.

A.2 FILESIN

The `FILESIN` input file contains a list of the filenames of the input and output files used in the simulation. The lines beginning with the capital letter C are comment lines, and are not read. The other lines contain the partial path and file names of the `CHEM`, `ICOND`, `PATCH`, `STYPE`, `SPARAM`, `MOISTURE`, `SPDIS`, `DRAIN`, `VEC`, `CDRAIN`, `MDITCH`, and `PROFILES` files, respectively. These file names must be less than 40 characters in length. Figure A.2 shows an example of this input file.

```
&PARAMETERS
ALPHA_L          =     double float,
ALPHA_TH         =     double float,
ALPHA_TV         =     double float,
CHEM             =     integer,
CHOICE           =     integer,
Dstar            =     double float,
DELTA_i          =     double float,
DELTA_j          =     double float,
DELTA_k          =     double float,
DELTA_T          =     double float,
FILESIN          =     character string,
INCREMENTA       =     integer,
INCREMENTD       =     integer,
M                =     integer,
N                =     integer,
NUMDRAIN         =     integer,
O                =     integer,
STOPTIME         =     integer,
trans            =     (0 or 1),
/
```

Figure A.1: Example of `PARAM` input file.

A.3 CHEM

The `CHEM` input file is used by the OSERTM to define the chemicals that will be tracked during the simulation. The first line is a comments line and is not read. The second line contains the subsurface temperature in degrees Celsius. This value should be input as a real number. The third line is a comments line. The remaining lines contain a table of chemical species and their properties. Each row contains the information specific to one chemical. The first column of this table contains the chemical name. The program does not use this information; it is meant to be merely an aid to the user. The second column contains an integer identification code, which must range from 1 to the number of chemicals in the simulation, in order. The third, fourth, and fifth columns contain the soil-water partition coefficient, the Henry's Law coefficient in atmospheres per mole fraction, and the coefficient of gaseous phase molecular diffusion. These parameters should be input as real numbers. For example, the input file shown in Figure A.4 describes three chemicals: ammonium, oxygen, and nitrate.

```
C chemical parameter files
23/23chem.txt
23/23icond.txt
23/23patch.txt
C
C soil parameter files
flow/110501/113soiltype.txt
flow/110501/sparam.txt
C
C flow parameter files
flow/110501/113M.dat
flow/110501/113V.dat
flow/110501/113D.dat
C
C
23/output/23VEC.dat
23/output/23Cdrain.dat
23/output/23Mout.dat
23/output/23cgms1.dat
23/output/23cgms2.dat
23/output/23cgms3.dat
23/output/23cgms4.dat
```

Figure A.2: Example of `FILESIN` input file.

```
temperature in degrees C
15.0
 name        id   Kd       H(atm/X)   Dg
 ammonium    1    1.0d-2   0.00       0.00
 oxygen      2    0.00     43800.0    34.70
 nitrate     3    0.00     0.00       0.00
```

Figure A.3: Example of `CHEM` input file.

s	NT	mini	maxi	minj	maxj	mink	maxk	C(Mass/L3)
1	3	1	2	1	2	1	1	1.0
1	0	4	4	4	4	4	4	1.0
3	0	1	2	1	2	1	1	0.5
0	0	0	0	0	0	0	0	0.0

Figure A.4: Example of `ICOND` input file.

A.4 ICOND

The `ICOND` input file is used by the OSERTM to initialize the concentration profiles of the chemicals being tracked by the model. This file has the form of a table, where each column is 10 characters wide. The input type for the first eight columns is integer, while the last column takes a real value. The first row contains a comments field that is not read by the program. The remaining rows contain the min and max node index in the x-, y-, and z- directions, for chemicals of species s with a common concentration C. The first column of each row contains the integer chemical index used in the `CHEM` file to identify which chemical species is being described. The second column contains the maximum number of time steps during which this condition is true as an integer value. The third through eighth columns demarcate the boundaries of the uniform concentration regions, and the last column indicates the concentration of that region. The end of the file is indicated by a row of zeros. For example, in Figure A.4, for the first three time steps, the concentration of chemical 1 in nodes (1,1,1), (1,2,1), (2,1,1), and (2,2,1) will be 1.0 mass/volume, while the concentration of chemical 3 in those same nodes will be 0.5 mass/volume only at time zero. Also, chemical 1 will have an initial concentration of 1 mass/volume at cell node (4,4,4).

A.5 PATCH

The `PATCH` input file is used by the OSERTM to define the flux boundary condition of the model. This file has the form of a table, where each column is 10 characters wide. The input type for the first seven columns is integer, while the last column takes a real value. The first row contains a comments field that is not read by the program. The remaining rows contain the min and max coordinates in the x- and y-directions for chemicals of species s with a common concentration C. The first column of each row contains the integer chemical index used in the `CHEM` file to identify which chemical species is being described. The time step numbers during which the flux

s	beginTime	endTime	mini	maxi	minj	maxj	C(Mass/L3)
1	3	5	1	10	1	1	2.40d-2
3	1	10	1	10	1	1	1.00d-3
0	0	0	0	0	0	0	0

Figure A.5: Example of `PATCH` input file.

condition begins and ends are indicated in the second and third columns. The fourth through seventh columns contain node indices that demarcate the boundaries of regions with uniform concentration. Note that because it is assumed that mass only enters through the soil surface, only the x and y limits are needed. Finally, the last column indicates the concentration of that region, as calculated from Equation (2.13). The end of the file is indicated by a row of zeros. For example, in Figure A.5, for the first ten time steps there will be a flux of chemical 3 through the soil surface into nodes (1,1,1) through (10,1,1). At time steps three through five, there will be a flux of chemical 1 through the soil surface into those same nodes.

A.6 STYPE

The `STYPE` input file is used by the OSERTM to initialize the soil type within the model domain. The first line of this file is a comments line and is not read. The second line contains the number of unique soil types as an integer value. The third line is a comments line. The remaining lines are formatted as a table, with 10-character-wide columns. The first row of this table contains a header and is not read by the program. The remaining rows contain the min and max node indices in the x-, y-, and z- directions, which demarcate the boundaries for regions of homogenous soil types. The first column of each of these rows contains the integer index used to identify the soil type. The remaining columns contain the minimum and maximum node index in each direction of each homogenous region. The end of the file is indicated by a row of zeros. Because the OSERTM initializes the soil profile to contain soil type 1, the boundaries of soil type 1 need not be specified. For example, in Figure A.6, there are 2 soil types. Nodes (20,1,1), (20,2,1), (21,1,1), and (21,2,1) have soil type 2, while the remaining nodes in the model domain have soil type 1.

numtypes
2

TYPE	MINX	MAXX	MINY	MAXY	MINZ	MAXZ
2	20	21	1	2	1	1
0	0	0	0	0	0	0

Figure A.6: Example of `STYPE` input file.

TYPE	poros	bulk density
1	0.33	0.0
2	0.5	0.0

Figure A.7: Example of `SPARAM` input file.

A.7 SPARAM

The `SPARAM` input file is used by the OSERTM to define the properties for the soil types declared in the `STYPE` input file. This file is formatted as a table with three columns that are 10 characters wide. The first row of this table is a header row that is not read by the program, while the remaining rows contain the soil parameters for each soil type. The first column contains the integer index of the soil type, while the remaining two columns contain the soil porosity and bulk density as real numbers. The soil types must be listed in order from 1 to the number of soil types. Figure A.7 shows an example of this input file where there are two soil types. Notice that the bulk density may contain a zero value. The bulk density is only needed by the OSERTM to calculate sorbtion from the linear isotherm model described in Equation (2.3). If there are no chemicals that sorb to the soil, then the bulk density need not be specified. In this case a value of zero can be used.

A.8 SPDIS

The `SPDIS` input file is used by the OSERTM to define the x-, y-, and z-directional components of the specific discharge through each of the cell faces. The data should be written such that each time level is written as a column vector, and in each vector, the x-directional data comes before the y-directional data, which comes before the z-directional data. This data should also be able to be associated with the correct node when the cells are accessed, first in increasing j, then i, and finally k index. A header that indicates the time level precedes the column of data for each time level. The end of the data set is indicated by the text `ENDDS`. If the simulation

```
TS 0 100.00
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
          ⋮
TS 0 200.00
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
          ⋮
ENDDS
```

Figure A.8: Example of `SPDIS` input file.

has transient flow conditions, then data for each of the model time steps is necessary; however, if the simulation has steady state flow, then data is only needed at one time level. An example of the `SPDIS` data file is shown in Figure A.8.

A.9 MOISTURE

The `MOISTURE` input file is used by the OSERTM to define the moisture content at each node. The data should be written in such a way that each time level is written as a column vector, which, when reading the data from top to bottom can be associated with the correct node, if the cells are accessed first in increasing j, then i, and finally k index. The entire data set is preceded by a six-line header, which is necessary for the file to be read by GMS. A header that indicates the time level precedes the column of data for each time level. The end of the data set is indicated by the text `ENDDS`. If the simulation has transient flow conditions then data for each of the model time steps is necessary; however, if the simulation has steady state flow, then data is only needed at one time level. An example of the `MOISTURE` data file is shown in Figure A.9. Note that the header of this file is required so that this data may be imported into GMS for visualization. The value that follows ND and NC on the fourth and fifth lines of this header is the number of nodes in the block centered grid used for the simulation, and is equal to $m * n * o$.

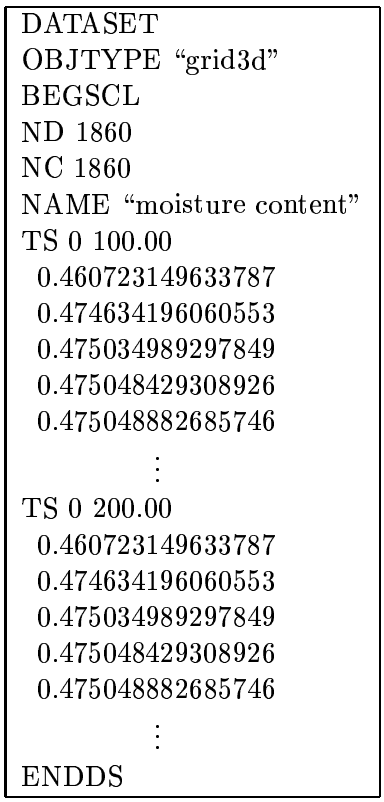

Figure A.9: Example of `MOISTURE` input file.

```
TS 0 100.00
  1   1   1   31   21   0.2615380729
TS 0 200.00
  1   1   1   31   21   0.2615380729
```

Figure A.10: Example of `DRAIN` input file.

A.10 DRAIN

The `DRAIN` input file is used by the OSERTM to determine the location of any drain segments that pass through cells in the model domain. If the simulation does not include any drains, then this file is not necessary. Data for each time level is written as a 6-column table, where columns 1 through 5 contain integer values and column 6 contains a real value. The first column contains the drain number, the second column contains the segment number, the third, fourth, and fifth columns contain the j, i, and k index of the node though which that drain segment passes, and the sixth column contains the flow rate [L^3/T] of that drain during that time step. If the simulation has transient flow conditions then data for each of the model time steps is necessary; however, if the simulation has steady state flow, then data is only needed at one time level. An example of the `DRAIN` input file is shown in Figure A.10.

Appendix B

Output Files

B.1 VEC

The `VEC` output file is designed so that the specific discharge field, defined by the `SPDIS` input file, can be imported into GMS for visualization purposes. As mentioned in Chapter 3, GMS requires that the directional components of the specific discharge vectors be defined at the nodes, not at the cell boundaries. Linear averaging is used to convert the data in the `SPDIS` file into this format. At each time step where data is collected, the elapsed time of the simulation will be written out in the format `TS 0 <elapsed time>`, followed by the specific discharge data [L/T]. The header of this file is required so that this data may be imported into GMS for visualization. The value that follows ND and NC on the fourth and fifth lines of this header is the number of nodes in the block-centered grid used for the simulation, and is equal to $m * n * o$.

B.2 CDRAIN

The `CDRAIN` output file is used to output the mass flux from the tile drain. Each row contains data taken at the elapsed time indicated in the first column. The remaining columns contain the mass flux [M/L^2] of chemical 1 through s in order, where s is the number of chemicals tracked during the simulation. Figure B.2 shows an example of this output file for three chemicals.

B.3 MDITCH

The `MDITCH` output file is used to output the mass flux through the model boundaries. Each row contains data taken at the elapsed time indicated in

```
DATASET
OBJTYPE "grid3d"
BEGVEC
ND 2760
NC 2760
NAME "specific discharge"
TS 0 100.00
 0.000000000   0.000211594   -0.000090808
 0.000000000   0.000014370   -0.000099330
 0.000000000   0.000000507   -0.000099976
 0.000000000   0.000000017   -0.000099999
 0.000000000   0.000000001   -0.000100000
 0.000000000   0.000000000   -0.000100000
 0.000000000   0.000000000   -0.000100000
 0.000000000   0.000000000   -0.000100000
 0.000000000   0.000000000   -0.000100000
 0.000000000   0.000000000   -0.000100000
      ⋮             ⋮             ⋮
TS 0 200.00
 0.000000000   0.000211594   -0.000090808
 0.000000000   0.000014370   -0.000099330
 0.000000000   0.000000507   -0.000099976
 0.000000000   0.000000017   -0.000099999
 0.000000000   0.000000001   -0.000100000
 0.000000000   0.000000000   -0.000100000
 0.000000000   0.000000000   -0.000100000
 0.000000000   0.000000000   -0.000100000
 0.000000000   0.000000000   -0.000100000
 0.000000000   0.000000000   -0.000100000
      ⋮             ⋮             ⋮
ENDDS
```

Figure B.1: Example of `VEC` output file.

0.00	0.000000000	0.000000000	0.000000000
13140.00	0.000000000	0.000241431	0.000000000
26280.00	0.000000000	0.000253830	0.000000000
39420.00	0.000000000	0.000278229	0.000000000
52560.00	0.000000000	0.000306930	0.000000000
65700.00	0.000000000	0.000333500	0.000000000
78840.00	0.000000000	0.000358209	0.000000000
91980.00	0.000000000	0.000381270	0.000000011
105120.00	0.000000000	0.000402483	0.000000121
118260.00	0.000000000	0.000425774	0.000000643

Figure B.2: Example of `CDRAIN` output file.

0.00	0.000000000	0.000000000	0.000000000
13140.00	0.000000000	0.000428971	0.000000000
26280.00	0.000000000	0.000403302	0.000000000
39420.00	0.000000000	0.000365887	0.000000000
52560.00	0.000000000	0.000347261	0.000000000
65700.00	0.000000000	0.000340931	0.000000000
78840.00	0.000000000	0.000340408	0.000000000
91980.00	0.000000000	0.000343895	0.000000000
105120.00	0.000000000	0.000350281	0.000000000
118260.00	0.000000000	0.000358457	0.000000000

Figure B.3: Example of `MDITCH` output file.

the first column. The remaining columns contain the mass flux [M/L^2] of chemical 1 through s in order, where s is the number of chemicals tracked during the simulation. Figure B.3 shows an example of this output file for three chemicals.

B.4 PROFILE

The `PROFILE` output file is designed so that the concentration profile of a specified chemical species at specified times during the simulation can be imported into GMS for visualization purposes. At each time step where data is collected, the elapsed time of the simulation will be written out in the format `TS 0 <elapsed time>`, followed by the concentration data [M/L^3]. The header of this file is necessary in order for these data to be imported into GMS for visualization. The value that follows ND and NC on the fourth and fifth lines of this header is the number of nodes in the block-centered grid used for the simulation, and is equal to $m * n * o$. Figure B.4 shows an

```
DATASET
OBJTYPE "grid3d"
BEGSCL
ND 1860
NC 1860
NAME "CHEM 1"
TS 0 100.00
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
          ⋮
TS 0 200.00
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
 0.000000000000000E+000
          ⋮
ENDDS
```

Figure B.4: Example of `PROFILE` output file.

example of this output file. Note that each `PROFILE` file only contains data for one chemical species. If the simulation contains many chemical species, a unique `PROFILE` file will be created for each species.