VLEACHSM 2.0

Vadose zone *LEACH*ing and
Saturated zone *M*ixing model

Version 2.0
(for Heterogeneous Layers)

Developed for:

The United States Environmental Protection Agency
Robert S. Kerr Environmental Research Center
P.O. Box 1198
Ada, Oklahoma 74820

Prepared by:

Samuel Sangwon Lee
Civil and Environmental Engineering
University of Rhode Island
Kingston, Rhode Island 02881-0805

June 1998

FINAL REPORT

By Samuel Sangwon Lee

TASK 4: PREPARE A MODIFIED VLEACHSM MODEL MANUAL AND EXAMPLE PROBLEM

VLEACHSM 2.0:
Vadose zone LEACHing and Saturated zone Mixing model version 2.0

INTRODUCTION

This final report is in response to Dynamac Work Service Request SI-1-50 (EPA Contract 68-C4-0031) dated 8 July 1997 for modifications to an EPA computer program VLEACHSM 1.0a. This provides the final required deliverable item, task 4. The following tasks have been previously completed.

1) To rewrite the existing VLEACHSM 1.0a in C++,
2) To modify the program to incorporate vertical heterogeneity of vadose zone,
3) To add to the program capability to handle simulation results graphically, and
4) To add a Graphic User Interface (GUI) using Visual Basic 5.0.

The complete manual of VLEACHSM 2.0 and install disks are attached to this final report. Therefore, we will only mention about some key features of the program in this short report.

INTERFACE IMPROVEMENT

Data input and output of VLEACHSM 1.0a were based on ASCII text files, and the user was expected to create such an input file with an editor, execute the program, and import the output files into a graphic software to plot the simulation results. Because creating such an input file with an editor requires careful attention in positioning each parameter value within the specified columns, it was desirable that an easy-to-use preprocessor with Graphic User Interface (GUI) be provided to create such an input data file. Also, because importing the simulation output results into another graphic software requires laborious reformatting of the data structure, it was also desirable that a postprocessor be provided to plot the output results in x-y graphs using GUI.

CAPABILITY OF SIMULATING VERTICAL HETEROGENEITY OF VADOSE ZONE

The demonstration calculations by VLEACHSM 2.0 clearly demonstrate that total soil porosity and its water-filled porosity have strong effects on the liquid-phase contaminant concentration profile. Vertical heterogeneity of the soil columns affects not only the contaminant concentration profiles in the vadose zone but also the contaminant mixing rate into the groundwater. Addition of this capability will possibly make VLEACHSM 2.0 very useful in its real-world applications.

SUMMARY

With this upgrade from version 1.0a to 2.0, VLEACHSM program has been made very easy to use thanks to its extensive use of GUI and capable of simulating actual situations better by allowing vertical heterogeneity of vadose zone. The extent of its usefulness in studying the actual soil and groundwater contamination problems is yet to be demonstrated in future research projects.

ATTACHMENTS

1. VLEACHSM 2.0 USER'S MANUAL
2. INSTALLATION DISKS FOR VLEACHSM 2.0

TABLE OF CONTENTS

PAGE

TABLE OF CONTENTS ---------- i

LIST OF FIGURES ---------- iii

LIST OF TABLES ---------- v

1.0 Introduction ---------- 1

2.0 Model Conceptualization, Assumptions, and Limitations ---------- 4

2.1 Vadose Zone Leaching ---------- *4*

2.2 Saturated Zone Mixing ---------- *9*

3.0 Theoretical Background ---------- 13

3.1 Vadose Zone Transport ---------- *13*

3.1.1 Initial & Boundary Conditions ---------- *17*

3.2 Saturated Zone Mixing ---------- *19*

3.2.1 Solute Penetration Depth ---------- *20*

3.2.2 Assumptions & Limitations of Penetration Depth ---------- *22*

3.2.3 Adjustment of the Estimated Penetration Depth ---------- *24*

4.0 Numerical Implementation ---------- 26

4.1 Vadose Zone Leaching ---------- *26*

4.1.1 Initial & Boundary Conditions ---------- *28*

4.1.2 Numerical Stability ---------- *31*

4.2 Saturated Zone Mixing ---------- *32*

5.0 Model Operation ---------- 35

5.1 Input Data Format ---------- *37*

5.2 System Requirements ---------- 42

5.3 Installation ---------- 42

5.4 Using the Program ---------- 44

6.0 Example Problems ---------- 55

6.1 Problem Description ---------- 55

6.2 Results ---------- 58

6.3 Discussion ---------- 59

7.0 Summary ---------- 64

References Cited ---------- 65

APPENDIX A C++ Source Code of VLEACHSM 2.0

APPENDIX B Input Parameter Information

APPENDIX C Finite Difference Form of the Third Type Top Boundary Condition

APPENDIX D Simplified Estimation for Depth of Penetration

APPENDIX E Thomas Algorithm for VLEACHSM 2.0

APPENDIX F List of Symbols

LIST OF FIGURES

FIGURE PAGE

2.1 Schematic diagram illustrating the contaminant pathways in the vadose zone simulated by VLEACHSM. ---------- 5

2.2 Schematic diagram illustrating the discretization of columns in vadose zone and the saturated zone mixing aligned parallel to groundwater flow direction. ---------- 7

2.3 Schematic diagram illustrating model conceptualization as soil columns in the vadose zone and the mixing zone in the saturated medium. The soil columns are arranged transverse to the groundwater flow direction. ---------- 10

2.4 Schematic diagram illustrating model conceptualization as soil columns in the vadose zone and the mixing zone in the saturated medium. The soil columns are arranged parallel to groundwater flow direction. ---------- 11

3.1 Schematic diagram, for parallel soil column arrangement case, illustrating the saturated zone mixing. The contaminant penetration depths underneath each soil column (Fig. 3.1b) were adjusted to produce a composite total depth (Fig. 3.1c). ---------- 21

4.1 Illustration of saturated zone mixing mechanism for parallel column arrangement. ---------- 34

5.1 Flow chart of VLEACHSM 2.0 for computer programming ---------- 36

5.2 Opening Screen of New VLEACHSM 2.0 ---------- 44

5.3 Open File Screen ---------- 45

5.4 Main Menu ---------- 45

5.5 Simulation Data Screen ---------- 46

5.6 Soil Properties I Screen ---------- 47

5.7 Soil Properties II Screen ---------- 48

5.8 Save File Screen ---------- 49

5.9 Overwrite Warning Screen ---------- 49

5.10 Run Main Menu ---------- 50

5.11 Output File Screen ---------- 51

5.12 Initial Graph Screen ---------- 52

5.13 Example Plots of Graphed Data ---------- 53

5.14 Printed Graph of Liquid-Phase Contaminant Concentration (mg/L) vs. Depth from Ground Surface (ft) ---------- 54

6.1 Profile of Transverse Column Example Problem in VLEACHSM 2.0 -------- 56

6.2 Profile of Parallel Column Example Problem in VLEACHSM 2.0 ---------- 57

6.3 Simulation results of Vertically Homogeneous Soil Column I, 1st type of Boundary Condition at 10, 20, 30, and 40 years ---------- 60

6.4 Simulation results of Vertically Homogeneous Soil Column II, 3rd type of Boundary Condition at 10, 20, 30, and 40 years ---------- 61

6.5 Simulation results of Vertically Heterogeneous Soil Column I, 1st type of Boundary Condition at 10, 20, 30, and 40 years ---------- 62

6.6 Simulation results of Vertically Heterogeneous Soil Column II, 3rd type of Boundary Condition at 10, 20, 30, and 40 years ---------- 63

LIST OF TABLES

FIGURE PAGE

5.1 INPUT DATA FORMAT FOR NEW VLEACHSM 2.0 ---------- 38

5.2 INPUT PARAMETERS FOR NEW VLEACHSM 2.0 ---------- 39

5.3 VLSM1A.INP (VERTICALLY HOMOGENEOUS COLUMN): NEW FORMAT FOR VLSM1.INP IN VERSION 1.0a OF VLEACHSM --------- 40

5.4 VLSM1B.INP (VERTICALLY HETEROGENEOUS COLUMN): NEW FORMAT FOR VLSM1.INP IN VERSION 1.0a OF VLEACHSM ---------- 41

1.0 INTRODUCTION

VLEACHSM, Vadose-zone LEACHing and Saturated-zone Mixing model, is a computer code for simulating pollutant migration through the vadose zone and subsequent mixing within the saturated zone. A one-dimensional finite difference scheme was employed for solving the vadose zone leaching equation, and a mass-balance principle was used for the mixing calculation within the saturated aquifer underneath the soil columns. The model can be applied to either one time ("slug") contaminant spill or to a continuous contaminant input. The model was initially developed by Dr. Sam Lee at Dynamac Co. for the U. S. Environmental Protection Agency in 1996.

Although several sophisticated computer codes are available which can incorporate the heterogeneity of the soil properties (*e.g.*, volumetric water content and corresponding infiltration rate), often many sites do not have the same degree of sophistication in the field-measured data. Even when there is a site with a reasonable amount of field data, a screening level of estimation is often necessary before conducting a comprehensive simulation. The VLEACHSM 2.0 model was designed to meet these needs by combining a one dimensional heterogeneous vadose zone transport model and saturated zone mixing model.

The vadose zone module of VLEACHSM 2.0 simulates the transport processes including liquid-phase advection, liquid- and vapor-phase dispersion, sorption, and decay of the contaminant. The one-dimensional simulator is an extended version of the existing VLEACH (Ravi and Johnson, 1993) model in that the liquid-phase dispersion and decay terms are added by Lee (1996) in VLEACHSM version 1.0a. In VLEACH the vadose zone transport process in the liquid and gaseous phases are simulated separately for each

time step. At the end of each time step, the concentration in each phase is re-estimated by letting the contaminant reach equilibrium among the liquid, solid, and gaseous phases.

In VLEACHSM 2.0, the transport processes in all the phases are combined together in one equation using the linear equilibrium partitioning relationships. This not only provides a more consistent formulation, but is computationally more efficient since only one differential equation needs to be solved. Furthermore, VLEACHSM 2.0 allows the specification of two different types (Dirichlet's and Cauchy's) of boundary conditions at the top of each layer of the soil column.

Using a simple mass-balance technique, the saturated zone module estimates the concentration of contaminants following the mixing of leachate from the vadose zone with groundwater. This module uses the effluent concentration at the bottom of the soil column, which is estimated from the vadose zone module. A complete mixing of the leachate with the groundwater is assumed.

Although VLEACHSM 2.0 employs a number of major assumptions, it can be useful in making preliminary assessments of the potential impacts of contaminants in the subsurface. The model input parameter values for the hydrogeologic properties and chemical properties are somewhat simplified by assuming them to be constant throughout the simulation, but are made to reflect vertical heterogeneity of the soil columns.

The user is responsible for evaluating whether the model is suitable for the site-specific hydrogeologic conditions and corresponding modeling objective(s). Also, it is the user's responsibility to conceptualize the site properly and determine the appropriate model input parameter values.

This User's Manual provides essential information on such important aspects as model conceptualization, model theory, assumptions and limitations, determination of input parameters, model operation, analysis of results, and result comparisons with simple cases.

2.0 MODEL CONCEPTUALIZATION, ASSUMPTIONS, AND LIMITATIONS

Any mathematical model involves a simplification of real-world systems with the degree of simplification being inversely proportional to the sophistication of the model. The simpler model requires more significant idealization of the real-world system. VLEACHSM 2.0 is a simple model intended primarily for screening level estimation of solute transport phenomena in the subsurface. In this section, the assumptions and limitations are separately discussed for the vadose zone and saturated zone modules.

2.1 Vadose-Zone Leaching

The movement of an organic contaminant in the vadose zone can be described within and among three different phases: (1) as a solute dissolved in water, (2) as a gas in the vapor phase, and (3) as an adsorbed compound in the solid phase (see Figure 2.1). Equilibration among the phases occurs according to distribution coefficients defined by the user. Contaminant decay was incorporated in the model assuming a uniform decay rate for the three different phases of the user selected chemical. VLEACHSM 2.0 simulates vertical transport through vertically heterogeneous soil layers by advection and dispersion in the liquid phase and by gaseous diffusion in the vapor phase.

These processes are conceptualized as occurring in a number of distinct, user-defined soil columns, which are vertically divided into a series of user-defined cells within each layer. The soil columns and layers may differ in soil properties, recharge

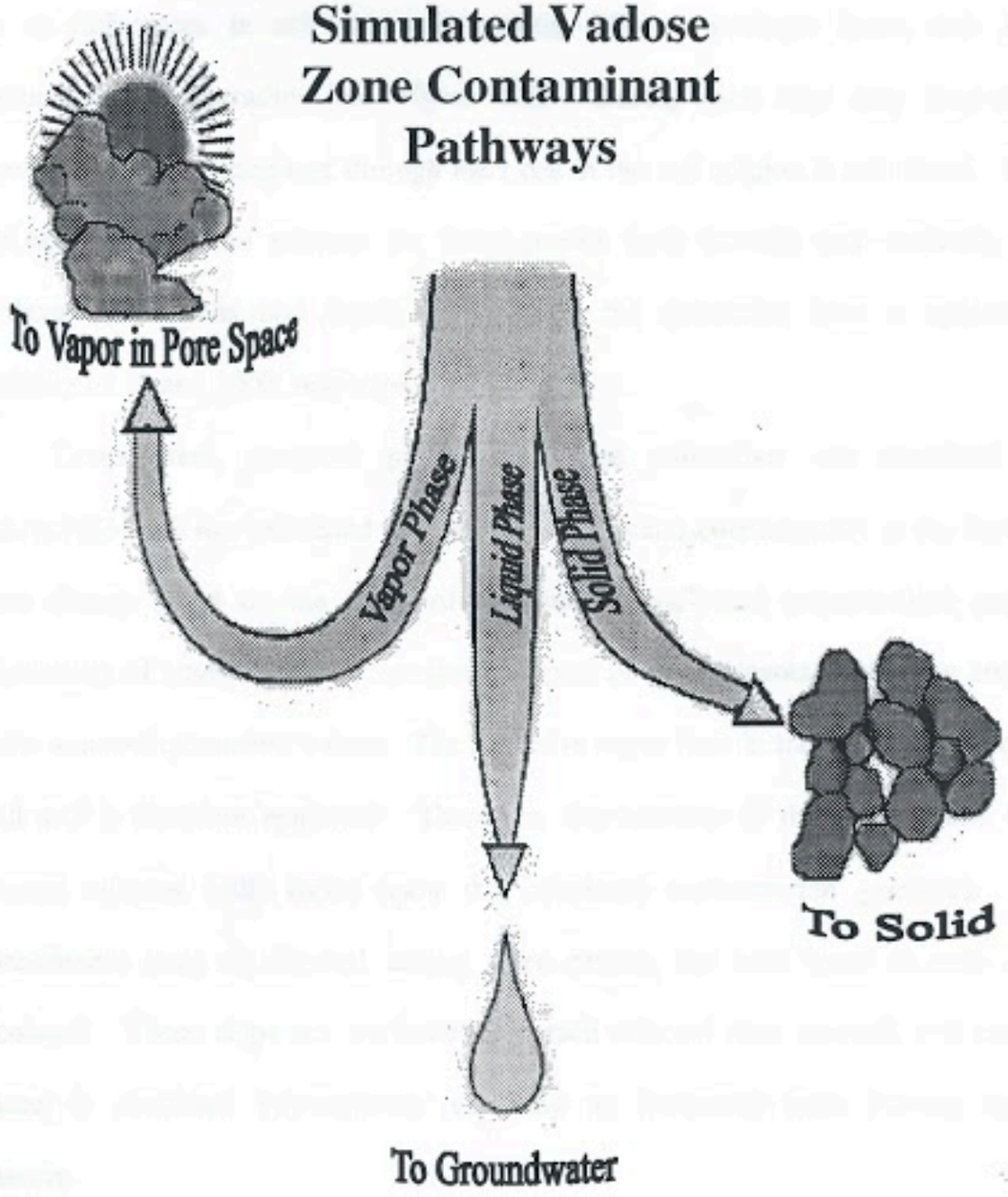


Figure 2.1 Schematic diagram illustrating the contaminant pathways in the vadose zone simulated by VLEACHSM (from Ravi and Johnson, 1993)

rate, and depth to water. Within each soil column heterogeneous conditions are assumed due to differences in soil properties among different geologic layers and also in contaminant concentration (see Figure 2.2). During each time step intervals the migration of the contaminant through each cell in the soil column is calculated. Hence, VLEACHSM 2.0 can account for heterogeneity both laterally and vertically. The numbers of columns and layers employed in the simulation have a mathematical capability of 50 and 1000, respectively.

Contaminant transport processes in the subsurface are simulated after VLEACHSM 2.0 has calculated contaminant equilibrium concentration in the liquid and vapor phases based on the user provided initial solid-phase concentration (mg/Kg). Calculations of advective and dispersive transport of contaminants in solution are based on the assumed parameter values. The advective vapor flow in the soil is assumed to be small and is therefore neglected. Therefore, contaminants in the vapor phase diffuse between adjacent cells based upon the calculated concentration gradients. After contaminants have equilibrated among three phases, the total mass in each cell is calculated. These steps are conducted for each selected time interval, and each soil column is simulated independently assuming no horizontal mass transfer between columns.

For computational purposes each soil column is divided into a series of cells. When developing a model simulation, it is important to fully understand the implications of the VLEACHSM 2.0 conceptualization. The following assumptions are made in the conceptualization of VLEACHSM 2.0 in the vadose zone.

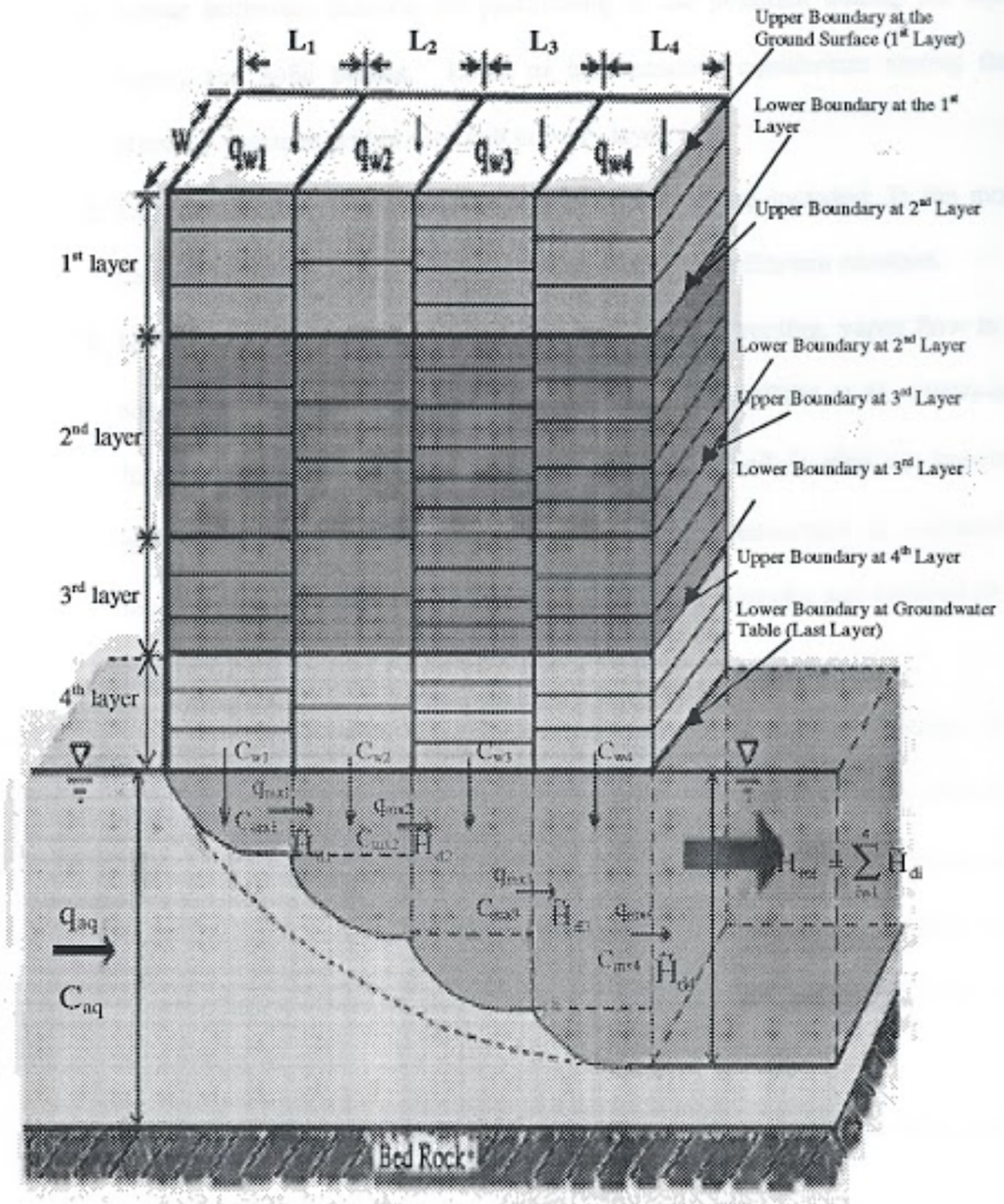


Figure 2.2 Schematic diagram illustrating the discretization of soil columns in vadose zone and the saturated zone mixing aligned parallel to groundwater flow direction.

1. Linear isotherms describe the partitioning of the pollutant among the liquid, vapor, and solid phases. Local or instantaneous equilibrium among these phases is assumed within each cell in every layer.
2. Liquid phase dispersion and vapor phase diffusion are included. In the model the two terms are combined as one via the Henry's equilibrium constant.
3. Unless there is a special remedial action such as soil venting, vapor flow in the subsurface is negligible. Although some researchers (Falta et al., 1989) have found that density driven adjective flow in the soil is also an important mechanism for gas movement, the vapor-phase advection is assumed as negligibly small compared to the infiltrating water velocity and omitted in this screening level model.
4. Organic contaminants, especially hydrocarbons, generally undergo some degree of degradation in the vadose zone. The module uses two decay constants to describe the first order decay; one for the contaminant decay in the soil, and another for the contaminant source reduction. The finite initial source concentration decreases due to either degradation or flushing by fresh water infiltration.
5. There is no capillary fringe. In most permeable soils this is very small in proportion to the column length and can be neglected.

2.2 Saturated-Zone Mixing

The leachate from the vadose zone will reach groundwater in the saturated zone and will be mixed. In VLEACHSM 2.0, this mixing process is simulated by employing a mass-balance principle.

For the purpose of mixing calculation in the saturated zone, the conceptual vadose zone soil columns are assumed to be arranged in one of two ways. To represent a contaminant source in the soil, a group of vertical soil columns can be arranged at either transverse or parallel alignment to the groundwater flow direction (see Figures 2.3 and 2.4). For the transverse arrangement (Figure 2.3), each soil column is treated independently both in the vadose and in the saturated zones. For the parallel case (Figure 2.4), only the vadose zone migration in each column is treated independently. For the saturated zone mixing, for the parallel case, the up-gradient mixed concentration is used as influent concentration for estimating the next down-gradient column's mixing (a detailed discussion is presented in the Section 3.2 and Figure 3.1 in that section). In both cases, the bottom of each soil column is assumed to be parallel to the water table.

For the screening level mixing calculation, following assumptions are made in the development of VLEACHSM 2.0.

1. The leachate is assumed to be completely mixed in the aquifer directly under the soil column.
2. The aquifer is homogeneous with steady state and uniform groundwater flow.
3. Groundwater mounding due to potential high infiltration is neglected.

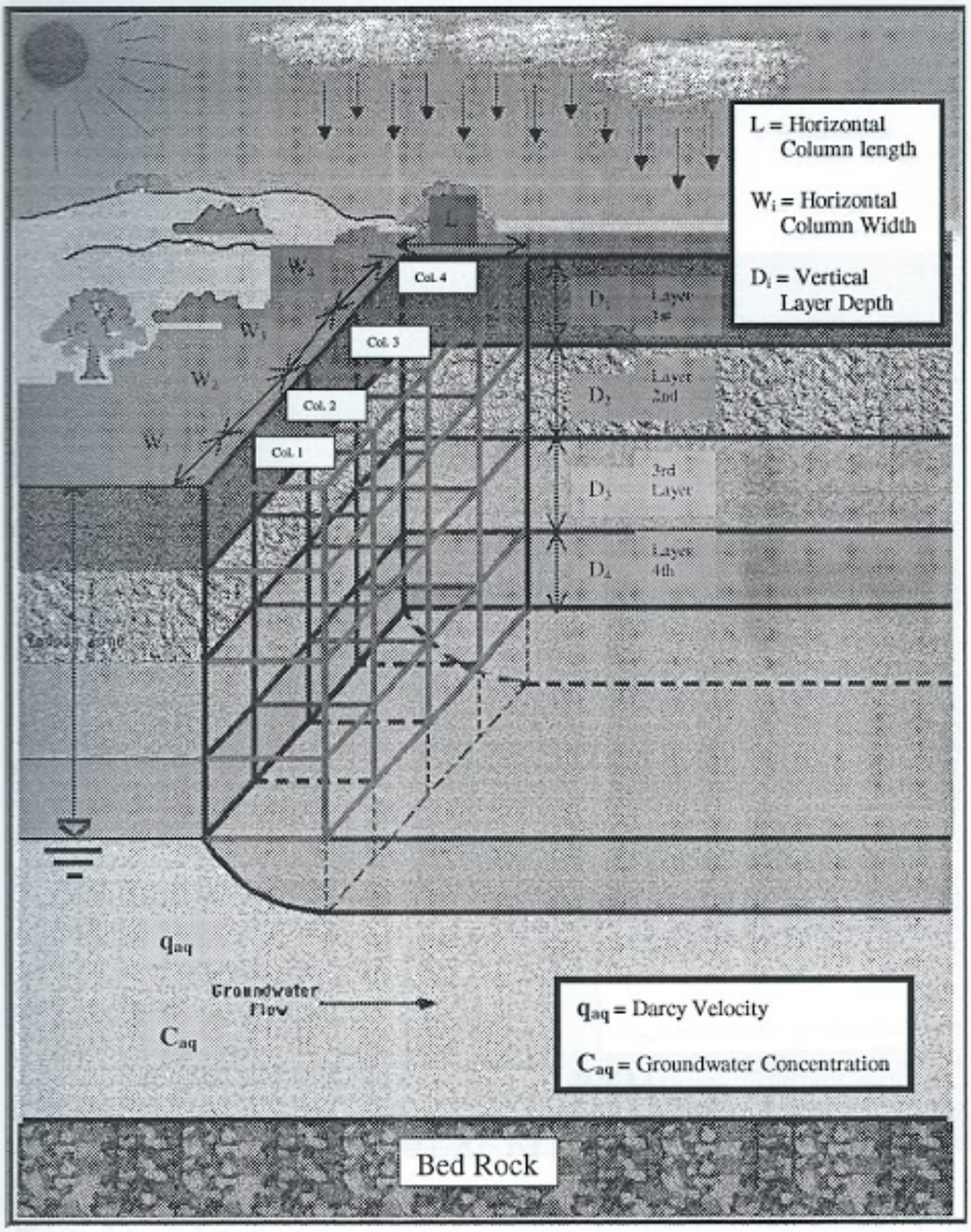


Figure 2.3 Schematic diagram illustrating model conceptualization as soil columns in the vadose zone and the mixing zone in the saturated medium. The soil columns are arranged transverse to the groundwater flow direction.

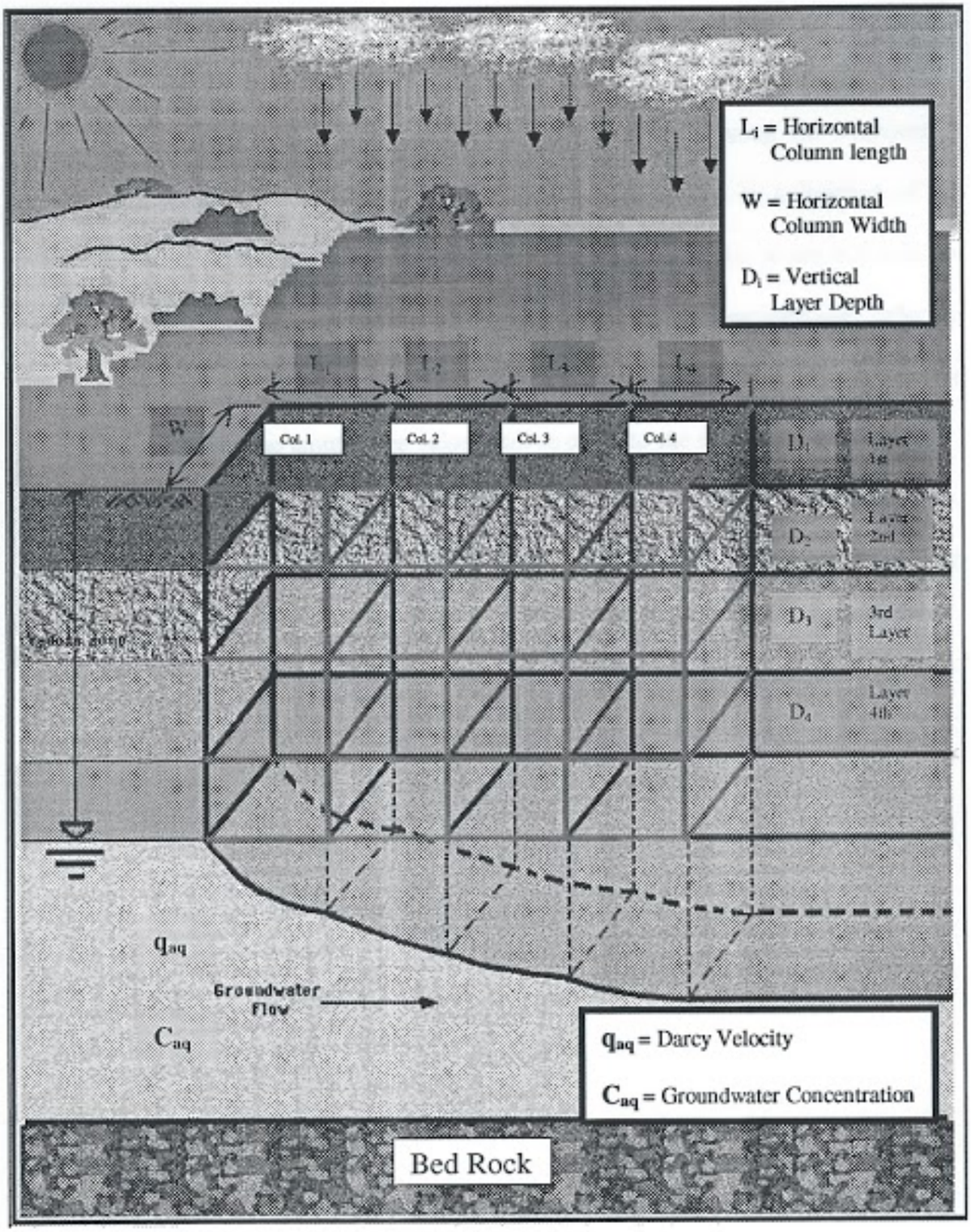


Figure 2.4 Schematic diagram illustrating conceptualization as soil columns in the vadose zone and the mixing zone in the saturated medium. The soil columns are arranged parallel to groundwater flow direction.

4. The decay of the solute in the saturated zone is conservatively assumed to be small and negligible.
5. The leachate penetration in the aquifer is estimated using the relationship presented in EPACML (EPA, 1990a). It is introduced as a function of the vertical infiltration velocity, horizontal length of the source in the soil column, vertical transverse dispersivity in the aquifer, horizontal groundwater velocity, and the aquifer thickness. When the estimated penetration is greater than the thickness of the aquifer, the value is set as equal to the thickness of the aquifer.
6. Contaminant spreading in the horizontal transverse direction (*i.e.*, perpendicular to the groundwater flow) is neglected. This assumption may cause somewhat higher mixed concentration than would occur naturally.

3.0 THEORETICAL BACKGROUND

In this section, a brief theoretical development of the VLEACHSM 2.0 (multi-layer version) is presented for vadose zone transport and saturated zone mixing.

3.1 Vadose Zone Transport

If we consider three equilibrium phases of pollutants in an unsaturated soil column, its one-dimensional governing transport equation is expressed as follows:

$$\theta_w \frac{\partial C_w}{\partial t} + \theta_a \frac{\partial C_a}{\partial t} + \rho_b \frac{\partial C_s}{\partial t} = \frac{\partial}{\partial z}\left(\theta_w D_w \frac{\partial C_w}{\partial z}\right) + \frac{\partial}{\partial z}\left(\theta_a D_a \frac{\partial C_a}{\partial z}\right)$$

$$-\frac{\partial}{\partial z}(q_w C_w) - \mu_w \theta_w C_w - \mu_a \theta_a C_a - \mu_s \theta_b C_s \qquad (3.1)$$

where

C_w Concentration of a contaminant in liquid (water) phase [mg/L].

C_a Concentration of a contaminant in vapor (air) phase [mg/L].

C_s Concentration of a contaminant in solid phase [mg/Kg].

θ_w Volumetric water content (volume of water / total volume) [ft^3/ft^3].

θ_a Air-filled porosity (volume of air / total volume) [ft^3/ft^3].

Note that the total porosity (n) equals the sum of water filled porosity and air filled porosity.

q_w Water flow velocity (recharge rate) [ft/yr]. The air flow velocity (q_a) is assumed to be zero in this study.

D_w Dispersion coefficient for the liquid phase contaminant in the pore water [ft^2/yr].

D_a Gaseous phase diffusion coefficient in the pore air [ft^2/yr].

μ_w First order decay rate of a contaminant in water phase [1/yr].

μ_a First order decay rate of a contaminant in gaseous phase [1/yr].

μ_s First order decay rate of a contaminant in solid phase [1/yr].

In this model, for simplicity, it is assumed that $\mu_w = \mu_a = \mu_s = \mu$.

ρ_b Bulk density of the soil [gr/cm^3].

z vertical coordinate with positive being downward.

t Time [yr].

Instantaneous equilibrium (partitioning) of the contaminant among the phases is assumed to be realized according to the following linear relationships:

$$C_s = K_d C_w \quad \text{(Liquid-solid phase equilibrium)} \tag{3.2}$$

$$C_a = HC_w \quad \text{(Liquid-gas phase equilibrium)} \tag{3.3}$$

where K_d [ml/g] is the distribution coefficient between the solid phase and liquid phase, and H ($H = K_H/(RT)$, [dimensionless]) is the partition coefficient between the air phase and water phase. Using the empirical relationship, K_d can be expressed as $K_d = K_{oc} \cdot f_{oc}$,

where K_{oc} [ml/g] is the organic carbon-water partition coefficient and f_{oc} [g/g] is the fraction organic carbon of the soil.

The dimensionless form of the Henry's partition coefficient, H, can be determined from the more common form having the units of atmospheres-cubic meters per mole (atm-m^3/mol) using the following equation

$$H = K_H/(RT) \tag{3.4}$$

where K_H [atm-m^3/mol] is the dimensional form of Henry's Law constant, R is the universal gas constant ($R = 8.2 \times 10^{-5}$ atm-m^3/mol·K), and T is the absolute temperature (°K = 273.16 + °C).

The dispersion coefficient in the unsaturated zone is regarded as a linear function of the pore water velocity as:

$$D_w = \alpha_L \left(\frac{q_w}{\theta_w} \right) \tag{3.5}$$

where α_L is the longitudinal dispersivity [feet] of the vadose zone. Based on data from Gelhar *et al.* (1985) on the field measurements of dispersion in the unsaturated zone, Unlu *et al.* (1992) used the following relationship in their model to describe the scale-dependent dispersivity:

$$\ln \alpha_L = -4.933 + 3.811 \ln L_a \qquad (L_a \leq 2 \text{ meter}) \tag{3.6a}$$

$$\ln \alpha_L = -2.727 + 0.584 \ln L_u \qquad (L_u > 2 \text{ meter}) \tag{3.6b}$$

where L_u is vertical distance from the source to the observation point (*i.e.*, the water table). The equations (3.6a) and (3.6b) can be rewritten for the English unit as:

$$\alpha_L \text{ (feet)} = 2.554 \times 10^{-4} L_u^{3.811} \qquad (L_u \leq 6.562 \text{ feet}) \tag{3.6c}$$

$$\alpha_L \text{ (feet)} = 1.072 \times 10^{-1} L_u^{0.584} \qquad (L_u > 6.562 \text{ feet}) \tag{3.6d}$$

It should be recognized that dispersion remains a significant uncertainty in the modeling effort. Considerable debate exists regarding the appropriate practical treatment of field-scale dispersion. In general, field tests are the only methods by which reasonable estimates for dispersion can be obtained.

The gas phase diffusion coefficient (D_a) in the porous medium is calculated by modifying the free air diffusion coefficient using the Millington model (Millington, 1959):

$$D_a = D_{air} \frac{\left(n - \theta_w\right)^{7/3}}{n^2} \tag{3.7}$$

where D_{air} is the diffusion coefficient of the contaminant in the free air, and n is the total porosity of the soil.

By substituting equation (3.2) and (3.3) into (3.1), and neglecting the air flow velocity ($q_a = 0$), the governing transport equation can be simplified as:

$$\left(\theta_w + \theta_a H + \rho_b K_d\right)\frac{\partial C_w}{\partial t} = \frac{\partial}{\partial z}\left\{\left(\theta_w D_w + \theta_a D_a H\right)\frac{\partial C_w}{\partial z}\right\}$$

$$-\frac{\partial}{\partial z}\left(q_w C_w\right) - \mu\left(\theta_w + \theta_a H + \rho_b K_d\right)C_w \qquad (3.8)$$

If we set as

$$\theta \equiv \theta_w + \theta_a H + \rho_b K_d \qquad (3.9)$$

$$D \equiv \theta_w D_w + \theta_a D_a H \qquad (3.10)$$

By substituting equations (3.9) and (3.10) into (3.8)

$$\theta\frac{\partial C_w}{\partial t} = \frac{\partial}{\partial z}\left(D\frac{\partial C_w}{\partial z}\right) - \frac{\partial}{\partial z}\left(q_w C_w\right) - \mu\theta C_w \qquad (3.11)$$

The equation (3.11) can be rewritten as:

$$\theta\frac{\partial C_w}{\partial t} = D\frac{\partial^2 C_w}{\partial z^2} + \left(\frac{\partial D}{\partial z} - q_w\right)\frac{\partial C_w}{\partial z} - \left(\frac{\partial q_w}{\partial z} + \mu\theta\right)C_w \qquad (3.12)$$

3.1.1 Initial and Boundary Conditions

To solve equation (3.12), initial and boundary conditions are required as follows.

$$C_w(z,t)\big|_{t=0} = \frac{C_s(z,0)}{K_d} \qquad (K_d > 0) \tag{3.13a}$$

$$C_w(z,t)\big|_{t=0} = C_w(z,0) \qquad (K_d = 0) \tag{3.13b}$$

where C_s(z,0) is the initial solid-phase concentration specified by the user. When the distribution coefficient ($K_d = K_{oc} \cdot f_{oc}$) is zero, liquid-phase concentration must be entered as an initial concentration to avoid the program run-time error (division by zero).

The most common type of boundary condition to be applied at the top of the soil column is either the first type (Dirichlet's) or the third type (Cauchy's) as shown below.

$$C_w\big|_{z=0} = C_0(z=0)\exp(-\gamma t) \quad (t \le t_0) \tag{3.14a}$$

$$C_w\big|_{z=0} = 0 \qquad (t > t_0) \tag{3.14b}$$

or

$$-D\frac{\partial C_w}{\partial z} + q_w C_w\Big|_{z=0} = q_w C_0 \exp(-\gamma t) \quad (t \le t_0) \tag{3.15a}$$

$$-D\frac{\partial C_w}{\partial z} + q_w C_w\Big|_{z=0} = 0 \qquad (t > t_0) \tag{3.15b}$$

where C_0 is the liquid phase solute concentration in the infiltration water, γ is the decay rate [1/yr] of the solute source due to either degradation or flushing by the infiltration, and t_0 is the duration of solute release [yr] which can be selected to simulate either "slug" or continuos input.

At the bottom of the soil column, the second type boundary condition (Neuman's) is applied.

$$\frac{\partial C_w}{\partial z} = 0 \qquad (z = \infty) \tag{3.16}$$

In applying this boundary condition, equation (3.16) is actually implemented at a finite column length (*i.e.*, $z \neq \infty$). To reduce the finite length effect, dummy cells are added at the bottom of the soil column automatically in the numerical calculation in the model. After evaluation of $C_w(z,t)$, the total contaminant mass per unit volume of the soil is calculated as:

$$M(z,t) = M_a + M_w + M_s = \left[\theta_a H + \theta_w + \rho_b K_d\right] C_w = \theta C_w \tag{3.17}$$

3.2 Saturated Zone Mixing

After estimating the liquid phase solute concentration (C_w) at the bottom of the soil column, the mixed concentration in the aquifer can be calculated using a mass-balance technique as below (USPEA, 1989 and Summers *et al.*, 1980):

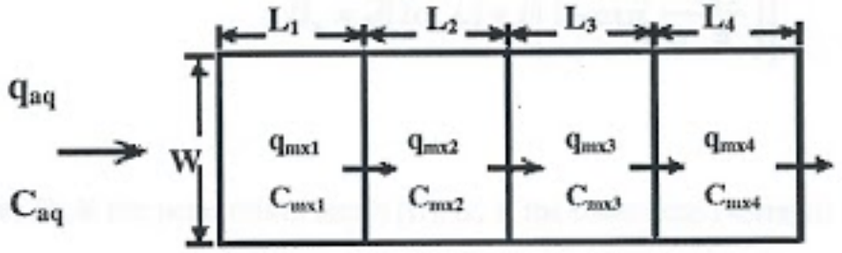
$$C_{mix} = \frac{\left(C_{aq} q_{aq} A_{aq} + C_w q_w A_{soil}\right)}{\left(q_{aq} A_{aq} + q_w A_{soil}\right)} \tag{3.18}$$

where C_{aq} is the concentration of horizontal groundwater influx, q_{aq} is the Darcy velocity in the aquifer, A_{aq} is the cross-sectional aquifer area perpendicular to the groundwater flow direction, and A_{soil} is the cross-sectional area perpendicular to the vertical infiltration in the soil column. The aquifer area (A_{aq}) is determined by multiplying the horizontal width of the soil column with the vertical solute penetration depth.

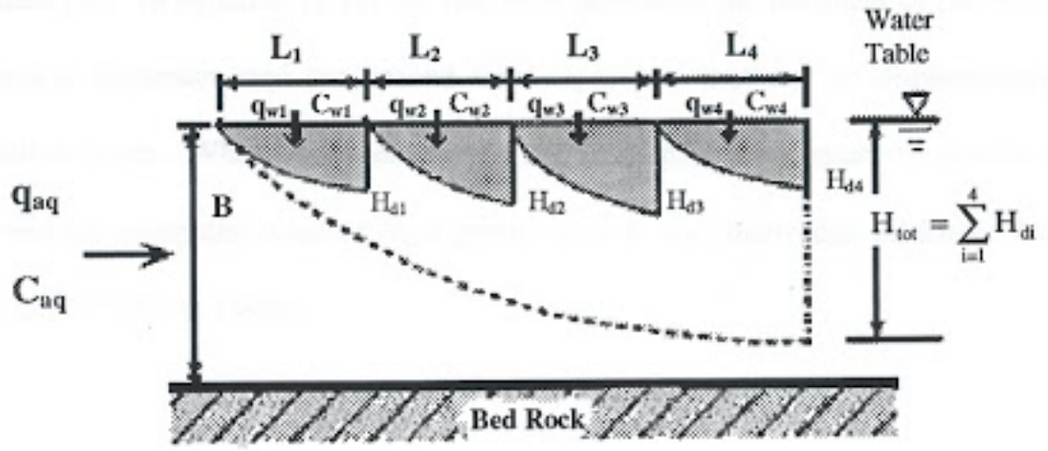
As discussed in Section 2.2, Saturated Zone Mixing, the procedure for the mixing calculation is different depending on the type of soil column arrangement. In the case of the transverse (right angle) arrangement, the mixing calculation is straight forward: simply apply equation (3.18) at the each mixing element underneath the soil columns. For the parallel arrangement case, however, the mixed concentration at the upgradient cell is considered as an influx concentration to the next cell (see Figure 3.1). The mixing concentration at the next cell is estimated by reapplying the equation (3.18) using the two inflow concentrations: one from the effluent from the upgradient cell and the other from the leaching from the soil column.

3.2.1 Solute Penetration Depth

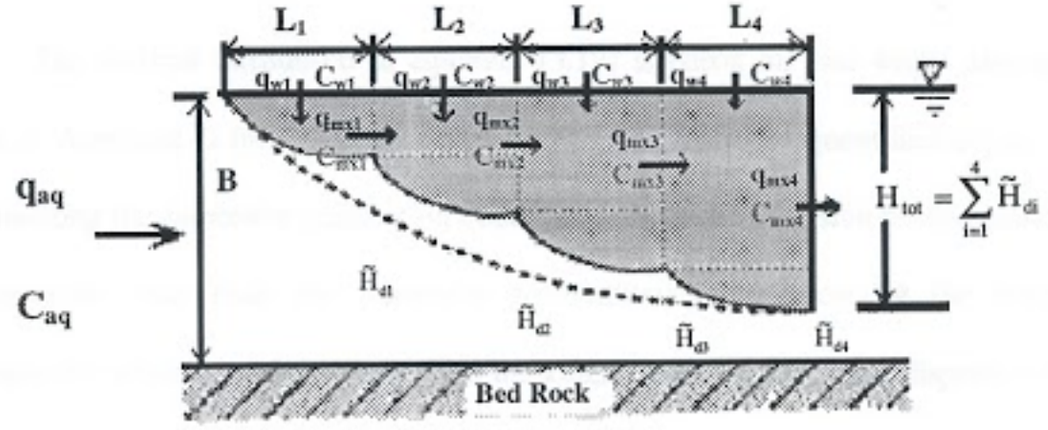
The solute penetration depth is the mixing thickness of the contaminant in the aquifer beneath the soil column. An estimation of the plume thickness in the aquifer can be made using the relationship below (adapted from USEPA, 1990a):



a) plan view (constant view (W) is assumed)



b) cross-sectional view – the penetration depth was estimated individually



c) cross-sectional view – the adjusted penetration depth

Figure 3.1 Schematic diagram, for parallel soil column arrangement case, illustrating the saturated zone mixing. The contaminant penetration depths underneath each soil column (Fig. 3.1b) were adjusted to produce a composite total depth (Fig. 3.1c).

$$H_d = \sqrt{(2\alpha_v L)} + B\left[1 - \exp\left(\frac{-Lq_w}{q_{aq}B}\right)\right] \qquad (3.19)$$

where, H_d is the penetration depth [ft], α_v is the transverse (vertical) dispersivity [ft] of the aquifer, L is the horizontal length dimension of the waste [ft], and B *is* the aquifer thickness [ft]. In equation (3.19) the first term represents the thickness of the plume due to vertical dispersion and the second term represents that due to displacement from infiltration water. When implementing this relationship, it is necessary to specify that in the event the computed value of H_d is greater than B, the penetration thickness, H_d is set equal to B (USEPA, 1990a).

3.2.2 Assumptions and Limitations Associated with the Penetration Depth Estimation

The method introduced in equation (3.19) is based on two major assumptions (refer to Appendix D for its derivation) and, therefore, has subsequent limitations. First, in estimating the dispersive penetration depth, there are two dispersion components in the mixing zone: one from the transverse (z-directional) dispersion by the horizontal groundwater velocity, and the other from the longitudinal (z-directional) dispersion by the infiltration velocity. In the first term of equation (3.19), however, the longitudinal dispersion due to infiltration is neglected by assuming that the horizontal groundwater velocity is much greater than the infiltration velocity. Therefore, when the horizontal

groundwater velocity is not substantially greater than the vertical infiltration velocity, the equation underestimates the penetration depth and overestimates the mixed concentration.

Secondly, the second term in equation (3.19), the advective penetration depth due to infiltration, is based on the assumption that the vertical (infiltration) velocity varies linearly with aquifer depth, with a maximum value at the top of the water table (*i.e.,* at the bottom of the soil column) and zero at the bottom of the aquifer. In case of either large aquifer thickness or higher horizontal velocity compared to the vertical, the vertical component of groundwater velocity becomes zero before the infiltration water reaches the bottom of the aquifer. In this case, the equation (3.19) overestimates penetration depth and, subsequently, underestimates the mixed concentration.

Due to the assumptions and limitations discussed above, the user must be careful when using the equation (3.19). Whenever field data are available, the measured solute penetration depth should be used instead of the equation. Otherwise, the input parameter values must be checked to see whether the assumptions are valid. For example, compare the horizontal and vertical groundwater velocities to ensure the horizontal velocity is much larger than the vertical one (*e.g.,* 10 times greater). In the VLEACHSM 2.0 program, an option is provided to choose either the measured penetration depth or to estimate the depth based on the aquifer parameter values.

3.2.3 Adjustment of the Estimated Penetration Depth

No adjustment of the penetration depth is required when either the depth is measured in the field or the soil columns are arranged at right angle to the horizontal groundwater flow direction. For the parallel arrangement, however, when the penetration depth needs to be estimated by the equation (3.19), the estimated depth is adjusted as follows. As shown in the example (Fig. 3.1), the total source dimension (L) is subdivided into four segments (L_1 through L_4) and the penetration depth for each soil column is calculated as:

$$H_{di} = \sqrt{(2\alpha_v L_i)} + B\left[1 - \exp\left(\frac{-L_i q_w}{q_{aq} B}\right)\right] \quad (i = 1 \text{ to m}) \tag{3.20}$$

where the subscript i represents the column 1 through 4. Since the relationship between the L and H_d is not linear in the equations (3.19 and 3.20), the sum of individual penetration depths based on the segment length is different from the overall penetration depth (*i.e.*, $H_d \neq H_{d1} + H_{d2} + H_{d3} + H_{d4}$). Note that the overall penetration depth (H_d) is based on the total contaminant length dimension (L) and the arithmetic average values of α_v, B, q_{aq}, and q_w for all columns. Therefore, the individual penetration depth is adjusted as:

$$\tilde{H}_{di} = H_{di} \frac{H_d}{H_{d1} + \cdots + H_{dm}} \quad (i = 1 \text{ to m}) \tag{3.21}$$

where $\tilde{H}_{di}$ is the adjusted penetration depth for each column.

Consequently, the relationship between the adjusted and individual penetration depth is expressed as:

$$H_d = \sum_{i=1}^{m} \tilde{H}_{di} \neq \sum_{i=1}^{m} H_{di} \tag{3.21}$$

4.0 NUMERICAL IMPLEMENTATION

4.1 Vadose Zone Leaching

The governing solute transport equation (3.12) is solved using the finite difference method. Differential equations dealing with liquid contaminant concentration C_w as a function of time and depth are converted into the finite difference equations dealing with the corresponding variable C_i^k centered on time between two time steps:

$$C_w \rightarrow \frac{\left(C_i^{k+1} + C_i^k\right)}{2}, \quad \frac{\partial C_w}{\partial t} \rightarrow \frac{\left(C_i^{k+1} - C_i^k\right)}{\Delta t} \tag{4.1}$$

where Δt is the time increment, the subscript i refers to the discretized soil column cell and the superscript k refers to the time level. The subscript w is dropped for simplicity. The first derivative by depth z at time step k becomes

$$\left(\frac{\partial C_w}{\partial z}\right)_k \rightarrow \frac{1}{2}\left(\frac{C_{i+1}^k - C_i^k}{\Delta z} + \frac{C_i^k - C_{i-1}^k}{\Delta z}\right) = \frac{1}{2\Delta z}\left(C_{i+1}^k - C_{i-1}^k\right) \tag{4.2}$$

which is converted into the form centered on time between time steps k and k+1:

$$\frac{\partial C_w}{\partial z} = \frac{1}{2}\left\{\left(\frac{\partial C_w}{\partial z}\right)_k + \left(\frac{\partial C_w}{\partial z}\right)_{k+1}\right\} \rightarrow \frac{1}{4\Delta z}\left(C_{i+1}^k - C_{i-1}^k + C_{i+1}^{k+1} - C_{i-1}^{k+1}\right) \tag{4.3}$$

The second derivative by depth z becomes

$$\left(\frac{\partial^2 C_w}{\partial z^2}\right)_k = \frac{\partial}{\partial z}\left(\frac{\partial C_w}{\partial z}\right)_k \rightarrow$$

$$\frac{1}{\Delta z}\left(\frac{C_{i+1}^k - C_i^k}{\Delta z} - \frac{C_i^k - C_{i-1}^k}{\Delta z}\right) = \frac{1}{(\Delta z)^2}\left(C_{i+1}^k - 2C_i^k + C_{i-1}^k\right) \tag{4.4}$$

In the same way, this is converted into the form centered between two time steps:

$$\frac{\partial^2 C_w}{\partial z^2} = \frac{1}{2}\left\{\left(\frac{\partial^2 C_w}{\partial z^2}\right)_k + \left(\frac{\partial^2 C_w}{\partial z^2}\right)_{k+1}\right\} \rightarrow$$

$$\frac{1}{2(\Delta z)^2}\left(C_{i+1}^k - 2C_i^k + C_{i-1}^k + C_{i+1}^{k+1} - 2C_i^{k+1} + C_{i-1}^{k+1}\right) \tag{4.5}$$

From equation (4.2),

$$\frac{\partial D}{\partial z} \rightarrow \frac{1}{2\Delta z}\left(D_{i+1} - D_{i-1}\right), \quad \frac{\partial q_w}{\partial z} \rightarrow \frac{1}{2\Delta z}\left(q_{i+1} - q_{i-1}\right) \tag{4.6}$$

By substituting equations (4.1), (4.3), (4.5), and (4.6) in equation (3.12),

$$\frac{\theta_i}{\Delta t}\left(C_i^{k+1} - C_i^k\right) = \frac{D_i}{2(\Delta z)^2}\left(C_{i+1}^{k+1} - 2C_i^{k+1} + C_{i-1}^{k+1} + C_{i+1}^k - 2C_i^k + C_{i-1}^k\right)$$

$$+\left\{\frac{D_i}{2\Delta z}\left(D_{i+1}-D_{i-1}\right)-q_i\right\}\frac{1}{4\Delta z}\left(C_{i+1}^{k}-C_{i-1}^{k}+C_{i+1}^{k+1}-C_{i-1}^{k+1}\right)$$

$$-\left\{\frac{1}{2\Delta z}\left(q_{i+1}-q_{i-1}\right)+\mu\theta_i\right\}\frac{1}{2}\left(C_i^{k+1}+C_i^{k}\right)$$

$$\text{for } 2 \le i \le \text{n-1, and k} = 1, 2, \ldots \tag{4.7}$$

If we define the dimensionless constants M_i, $M_i^{'}$, N_i, $N_i^{'}$, and L_i as:

$$M_i \equiv \frac{\Delta t}{2(\Delta z)^2}\frac{1}{\theta_i}D_i, \quad M_i^{'} \equiv \frac{\Delta t}{2(\Delta z)^2}\frac{1}{\theta_i}\frac{D_{i+1}-D_{i-1}}{4}$$

$$N_i \equiv \frac{\Delta t}{4\Delta z}\frac{1}{\theta_i}q_i, \quad N_i^{'} \equiv \frac{\Delta t}{4\Delta z}\frac{1}{\theta_i}\left(q_{i+1}-q_{i-1}\right)$$

$$L_i \equiv \frac{\Delta t}{2}\mu \tag{4.8}$$

equation (4.7) can be rewritten as:

$$\left(-M_i+M_i^{'}-N_i\right)C_{i-1}^{k+1}+\left(1+2M_i+N_i^{'}+L_i\right)C_i^{k+1}+\left(-M_i-M_i^{'}+N_i\right)C_{i+1}^{k+1}$$

$$=\left(M_i-M_i^{'}+N_i\right)C_{i-1}^{k}+\left(1-2M_i-N_i^{'}-L_i\right)C_i^{k}+\left(M_i+M_i^{'}-N_i\right)C_{i+1}^{k} \tag{4.9}$$

4.1.1 Initial and Boundary Conditions

By specifying the appropriate boundary conditions at both ends of the soil column, the n or n-1 simultaneous equations can be solved for the value of C at each node for the future time level t^{k+1}. The finite difference form of the initial condition for the liquid phase solute concentration is

$$C_i^1 = \frac{\left(C_s^1\right)_i}{K_d} \qquad (K_d > 0,\ 2 \le i \le \text{n-1}) \tag{4.10a}$$

or

$$C_i^1 = \left(C_w^1\right)_i \qquad (K_d = 0,\ 2 \le i \le \text{n-1}) \tag{4.10b}$$

The finite difference forms of the top boundary conditions (equations 3.14 and 3.15) for the soil column are:

First Type Top Boundary Condition,

$$C_i^k = \frac{C_s^k(z=0)}{K_d}\exp\left\{-\gamma(k-1)\Delta t\right\} \qquad (k = 1, 2, \ldots)$$

$$C_s^k \neq 0, \qquad \text{when } t \le t_0$$

$$C_s^k = 0, \qquad \text{when } t > t_0 \tag{4.11}$$

Third Type Top Boundary Condition (see Appendix C for derivation),

$$C_1^{k+1} - \frac{\Psi'}{\Phi'}C_2^{k+1} = -\frac{\Omega'}{\Phi'}C_1^k + \frac{\Psi'}{\Phi'}C_2^k + \frac{q_w C_0}{\Phi'}\exp\left(-\gamma t\right) \tag{4.12}$$

where

$$\Phi = \frac{D(2M+L+1)}{4(\Delta z)(M+N)} + \frac{q_w}{2}, \quad \Psi = \frac{DM}{4(\Delta z)(M+N)}, \quad \Omega = \frac{D(2M+L-1)}{4(\Delta z)(M+N)} + \frac{q_w}{2},$$

and M, N, and L were defined in equation (4.8).

The second type bottom boundary condition is used in this model as follows:

$$\frac{C_n^{k+1} - C_{n-1}^{k+1}}{\Delta z} = 0 \tag{4.13}$$

Note that as indicated in the section 3.1, equation (4.13) is supposed to be applied at the distance of z = ∞. This finite column length will cause somewhat higher concentration estimation at longer times near the end of the column. To reduce this finite boundary effect, dummy cells can be added at the bottom of the column either by changing input file or the source code. When the source code is modified, estimated concentration at the water table (n-dummy cells) should be used in calculation of saturated zone mixing.

The finite difference forms of simultaneous equations (4.5 through 4.13) are computer coded in C++ to solve for the value of C_i^{k+1} by the Thomas algorithm (Press et al., 1992).

4.1.2 Numerical Stability

Often, the efficiency of a numerical technique is limited due to the instability, oscillation, and mass-balance problems. Several methods have been proposed to determine the stability criteria of finite difference calculation (*e.g.,* Fourier expansion method, matrix method, and other, [Hirsch, 1989]). The Fourier expansion method, developed by von Neumann, relies on a Fourier decomposition of the numerical solution in space neglecting boundary conditions. It provides necessary conditions for stability of constant coefficient problems regardless of the type of boundary condition (Mitchel and Griffiths, 1980). The matrix method, however, takes eigenvectors of the space-discretization operator, including the boundary conditions, as a basis for the representation of the spatial behavior of the solution (Ames, 1977; Hirsch, 1989).

Based on the von Neumann method, Crank-Nicolson scheme of finite difference equation (Equations 4.1 through 4.7) can be derived as:

$$\Delta z < \frac{2D}{q_w} \tag{4.14}$$

$$\Delta t < \frac{2\theta(\Delta z)^2}{\sqrt{(2D)^2 - (q_w \Delta z)^2}} \tag{4.15}$$

According to the stability criteria expressed in equations (4.14) and (4.15), it is clear that the combined dispersion coefficient (D) must be greater than zero. In natural soil conditions, it is rare to have a value of D as zero or close to zero. Specifically, if there

is downward infiltration in the vadose zone, hydrodynamic dispersion of contaminant is inevitable. In addition, the air diffusion coefficients of selected organic compounds (in Appendix B) are must greater than zero.

Note that the numerical criteria presented in equations (4.14) and (4.15) are general guidelines and may not work for all situations. To ensure a stable result, the output should be checked thoroughly. If the results show any oscillation or negative concentration, a smaller time step and/or cell size than the values from the above equations should be tried. After having "stable" results, additional runs are recommended to check convergence of the simulation results. In this case, even a smaller time step and/or cell size can be used to check whether the solution converges or not.

4.2 Saturated Zone Mixing

The saturated zone mixing equation (3.18) is computer coded in the VLEACHSM 2.0 program. The calculation results versus time will be showed in a graphic outputs on VLEACHSM 2.0 program.

Based on the mass balance principle of equation (3.18) and Figure 4.1, the mixed concentration is estimated as:

$$C_{mx1} = \frac{q_{w1} L_1 C_{w1} + q_{aq} \tilde{H}_{d1} C_{aq}}{q_{mx1} \tilde{H}_{d1}} \qquad (4.16a)$$

$$C_{mx2} = \frac{q_{w2} L_2 C_{w2} + q_{aq} \tilde{H}_{d1} C_{aq} + q_{mx1} \tilde{H}_{d1} C_{mx1}}{q_{mx1} \left(\tilde{H}_{d1} + \tilde{H}_{d2} \right)} \qquad (4.16b)$$

Note that the last term in the above equation $\left(\frac{q_{mx1} \tilde{H}_{d1} C_{mx1}}{q_{mx2}} \right)$ is the same as $C_{mx(i)}$ in "Mixed Concentration in Groundwater" output data of VLEACHSM 2.0 program. Similarly, mixed concentrations at the subsequent elements can be estimated easily.

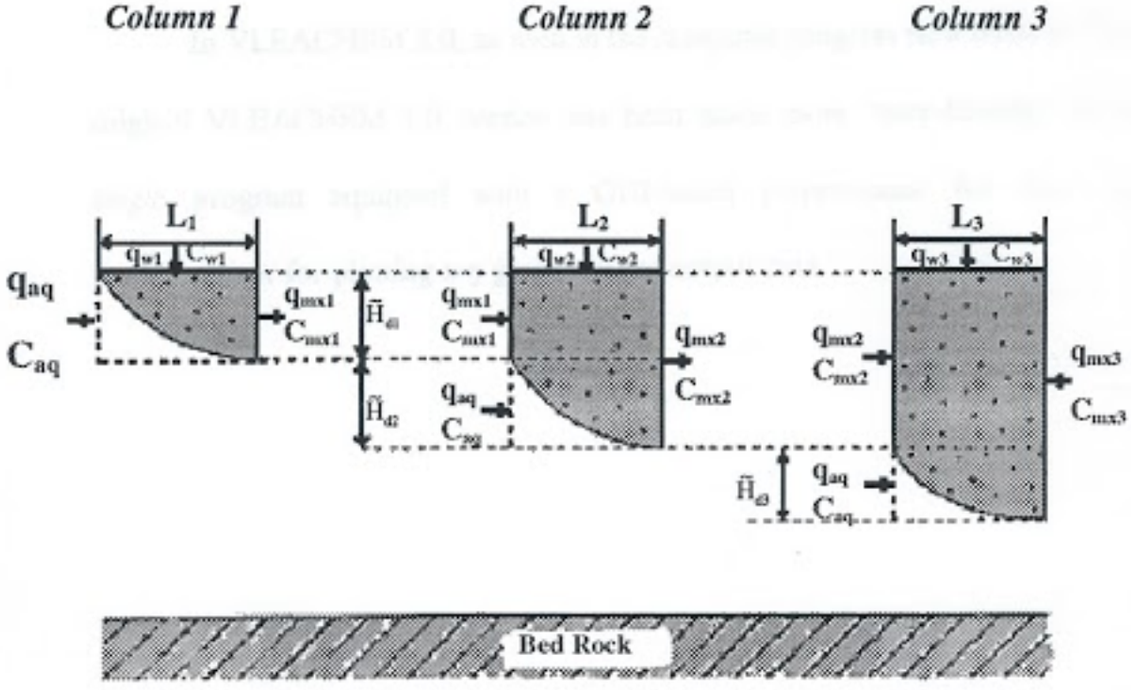


Figure 4.1 Illustration of saturated zone mixing mechanism for parallel column arrangement

5.0 MODEL OPERATION

In VLEACHSM 2.0, as seen in the computer program flow chart in Figure 5.1, the original VLEACHSM 1.0 version has been made more "user-friendly" by providing a single program equipped with a GUI-based preprocessor for data input and a postprocessor for plotting x-y graphs of the output data.

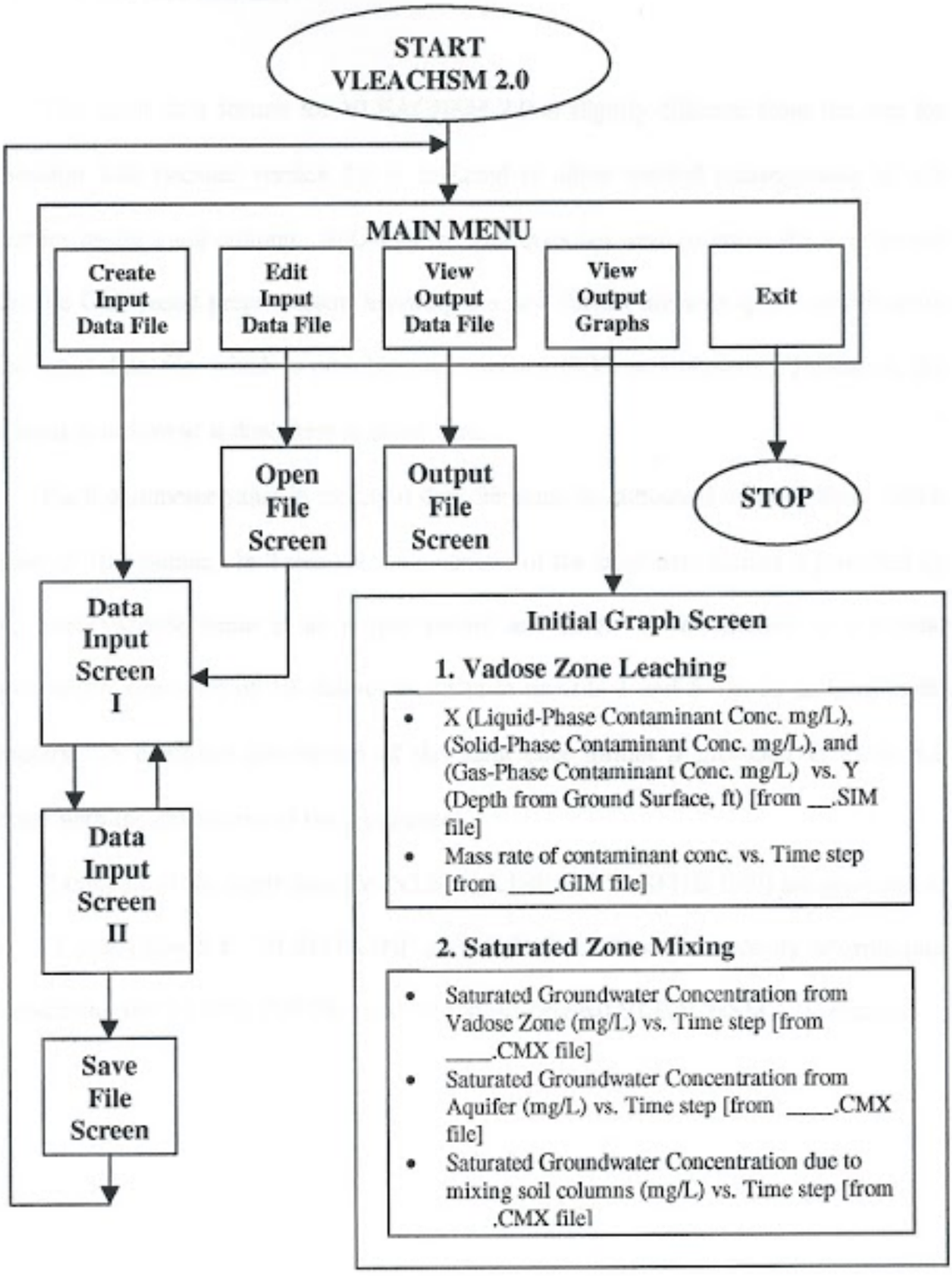


Figure 5.1 Flow chart of VLEACHSM 2.0 for computer programming

5.1 INPUT DATA FORMAT

The input data format for VLEACHSM 2.0 is slightly different from the one for the version 1.0a because version 2.0 is designed to allow vertical heterogeneity of soil properties inside a soil column. Although the user does not need to know the new format to use the GUI-based preprocessor, knowing the new format allows a quick modification of the input data file, which is possible even under a DOS environment. Therefore, the new input data format is described in detail here.

Each parameter value in the input data file must be contained within a field, which consists of 10 columns. In Table 5.1, an overview of the input data format is provided by listing each variable name in its proper record and field. As exceptions to a normal column width limitation of 10 characters, titles in records 1 and 5 can be as long as 80 characters. A complete description of the input data format is provided in Table 5.2 together with the definitions of the parameters.

Examples of the input data file (VLSM1A.INP and VLSM1B.INP) are provided in Table 5.3 and Table 5.4. VLSM1A.INP and VLSM2A.INP were made by reformatting and modifying the VLSM1.INP file contained in the original VLEACHSM 1.0a manual.

Table 5.1 INPUT DATA FORMAT FOR NEW VLEACHSM 2.0

RECORD	FIELD							
	1	2	3	4	5	6	7	8
1	TITLE							
2	NSCOL							
3	DELT	STINE	TIMPL	PTIME	PRTIME	ANGL		
4	KOC	KH	CMAX	DAIR				
5	TITLE1							
6	NCELL	DELZ	WIDTH	LENGTH	TYPTBC	VALTBC	ALDISP	
7	PLT	PLTIME	DECY1	DECY2				
8	NLAYER							
9	J1	J2	RECHRG	RHOB	POR	THETA	FOC	XCON
	:	:	:	:	:	:	:	
	:	:	:	:	:	:	:	
	:	:	:	:	:	:	:	
10	FLXAQF	CINAQF	PDEPTH	PCHOIC	ADEPTH	AVDISP		
5	TITLE2							
	:	:	:	:	:	:	:	

Table 5.2 INPUT PARAMETERS FOR NEW VLEACHSM 2.0

CARD	COLUMN	FORMAT	VARIABLE	DEFINITION
SIMULATION DATA				
1	1-80	A80	TITLE	Description of the Problem/Simulation
2	1-10	I10	NSCOL	Number of soil columns
3	1-10	F10.3	DELT	Time step [year]. ($\Delta z \leq \min\{\theta^*\Delta z/q_w, 1/\mu\}$)
	11-20	F10.3	STINE	Simulation Time [year]
	21-30	F10.3	TIMPL	Duration of pulse release of the source [year]
	31-40	F10.3	PTIME	Output time interval [year]
	41-50	F10.3	PRTIME	Profile time interval [year]
	60	A1	ANGL	Angle of the soil arrangement (T or P)
4	1-10	F10.3	KOC	Organic carbon distribution coefficient (K_{OC}) [ml/g]
	11-20	F10.3	KH	Henry's partition coefficient [dimensionless]
	21-30	F10.2	CMAX	Maximum water solubility of the contaminant [mg/L]
	31-40	F10.5	DAIR	Free air diffusion coefficient [cm^2/sec]
Soil Column-specific Data (this set is repeated NSCOL times)				
5	1-70	A70	TITLE1	Description of the soil column
6	1-10	I10	NCELL	Total number of cells (Vertical Column Length / DELZ + 1)
	11-20	F10.2	DELZ	Vertical length increment [ft]. ($\Delta z \leq 2^*D / q_w$)
	21-30	F10.2	WIDTH	Horizontal width of the soil column [ft]
	31-40	F10.2	LENGTH	Horizontal length of the soil column [ft]
	41-50	F10.2	TYPTBC	Type of top boundary condition (1 or 3, for 1st or 3rd type)
	51-60	F10.2	VALTBC	Value of the top boundary condition [mg/L]
	61-70	F10.3	ALDISP	Longitudinal dispersivity in the vadose zone [ft]
7	10	A1	PLT	Plotting option (y/n) to save a plot file (soil concentration profile and groundwater impact information)
	11-20	F10.2	PLTIME	Plot frequency in the plot file [year]
	21-30	F10.5	DECY1	Decay rate of the contaminant in the soil [1/year]
	31-40	F10.5	DECY2	Reduction rate of the contaminant in the source [1/year]
8	1-70	A70	NLAYER	Number of the soil column layers
Initial Soil Concentration (CARD 9 must be repeated until each cell has been described)				
9	1-10	I10	J1	Upper cell number
	11-20	I10	J2	Lower cell number. If J2<NCELL, repeat CARD 9.
	21-30	F10.4	RECHRG	Groundwater recharge rate [ft/year]
	31-40	F10.3	RHOB	Bulk density of the soil [g/cm3]
	41-50	F10.3	POR	Total (effective) porosity of the soil [dimensionless]
	51-60	F10.3	THETA	Volumetric Water Content (water filled porosity) of the soil, V_W/V_T, [dimensionless]
	61-70	F10.4	FOC	Soil organic carbon content [g/g]
	71-80	F10.3	XCON	Initial contaminant concentration in the soil column [mg/Kg] for solid-phase concentration, or [mg/L] for liquid-phase concentration depending on the values of KOC and FOC.
Aquifer-specific Data (Saturated Mixing Zone)				
10	1-10	F10.2	FLXAQF	Horizontal groundwater flux ($q_{aq}=K^*i$) [ft^3/yr/ft^2].
	11-20	F10.2	CINAQF	Concentration of incoming horizontal groundwater [mg/L]
	21-30	F10.2	PDEPTH	Measured (M)/Estimated (E) penetration depth [ft]
	40	A1	PCHOIC	Option (M/E) for penetration depth.
	41-50	F10.2	ADEPTH	Thickness of the aquifer [ft]
	51-60	F10.2	AVDISP	Vertical (transverse) dispersivity in the aquifer (α_v) [ft]

Table 5.3 VLSM1A.INP (VERTICALLY HOMOGENEOUS COLUMN): NEW FORMAT FOR VLSM1.INP IN VERSION 1.0a OF VLEACHSM

```
VLSM2 Result Comparison, TIMPL=20yr, 1st & 3rd type b. c.; VLSM1A.INP
2
0.1         20          20          10          10          P
100         0.4         1100        0.08102
Soil Column I: Case A  Zero Initial Soil Concentration
51          1           100         100         1           100         0
Y           10          0           0
4
1           10          1           1.6         0.4         0.3         0.005       0
11          20          1           1.6         0.4         0.3         0.005       0
21          30          1           1.6         0.4         0.3         0.005       0
31          51          1           1.6         0.4         0.3         0.005       0
10          1           15          M           15          2
Soil Column II: Case B  Zero Initial Soil Concentration
51          1           100         100         3           100         0
Y           10          0           0
4
1           10          1           1.6         0.4         0.3         0.005       0
11          20          1           1.6         0.4         0.3         0.005       0
21          30          1           1.6         0.4         0.3         0.005       0
31          51          1           1.6         0.4         0.3         0.005       0
10          1           21.22       E           50          1
```

Table 5.4 VLSM1B.INP (VERTICALLY HETEROGENEOUS COLUMN): NEW FORMAT FOR VLSM1.INP IN VERSION 1.0a OF VLEACHSM

VLSM2 Result Comparison, TIMPL=20yr, 1st & 3rd type b. c.; VLSM1B.INP							
2							
0.1	20	20	10	10	P		
100	0.4	1100	0.08102				
Soil Column I: Case A Zero Initial Soil Concentration							
51	1	100	100	1	100	0	
Y	10	0	0				
4							
1	10	1	1.387	0.48	0.26	0.005	0
11	20	1	1.493	0.44	0.28	0.005	0
21	30	1	1.6	0.4	0.3	0.005	0
31	51	1	1.76	0.34	0.33	0.005	0
10	1	15	M	15	2		
Soil Column II: Case B Zero Initial Soil Concentration							
51	1	100	100	3	100	0	
Y	10	0	0				
4							
1	10	1	1.387	0.48	0.26	0.005	0
11	20	1	1.493	0.44	0.28	0.005	0
21	30	1	1.6	0.4	0.3	0.005	0
31	51	1	1.76	0.34	0.33	0.005	0
10	1	21.22	E	50	1		

5.2 SYSTEM REQUIREMENTS

System requirements for VLEACHSM 2.0 are a PC with 80386 or higher processor (80486 or higher recommended) running Microsoft Windows 95 or Windows NT, a 3.5-inch high-density floppy drive (for installation), 8 megabytes of available memory (16 megabytes recommended), a VGA monitor (over 800 x 600 pixels of desktop area set up recommended for printing graphs), and a mouse. Optional hardware is an 80387 (or better) math coprocessor.

5.3 INSTALLATION

You can use the setup program (SETUP.EXE) to install VLEACHSM 2.0 on your computer. To run the setup program, click Start button on Taskbar, then click Run on the menu. Alternatively, you can open My Computer folder, double-click 3.5 Floppy (A:) to display the contents of Disk 1, and double-click Setup icon. After starting the setup program, follow the instructions on the screen.

After the installation, this program folder will be located in "C:\Program Files\VLEACHSM20" on your computer. In order to execute this program, click Start button, choose Programs, and click on VLEACHSM20.

This program can be uninstalled using the Add/Remove program in the standard Windows 95 system. To uninstall this program, click Start button on Taskbar, then click Control Panel in Settings on the menu. After double click the Add/Remove program in Settings folder, the user should choose VLEACHSM20 and click Add/Remove button.

You may get a Windows 95 system message "Program not completely removed". In that case, you simply delete this program folder (under C:\Program Files\VLEACHSM20) containing some output files for completely removing the program.

5.4 USING THE PROGRAM

The procedures of inputting data, viewing the output data, and plotting the output data are described in the following Figures 5.2 to 5.13. Simulation calculation is automatically done when the input data are saved. Many types of output data plots (liquid, solid, and gaseous phases contaminant concentration vs. depth, mass concentration vs. time, and time vs. saturated groundwater concentration from mixing soil column) are provided because they are the most important information for groundwater contamination studies.

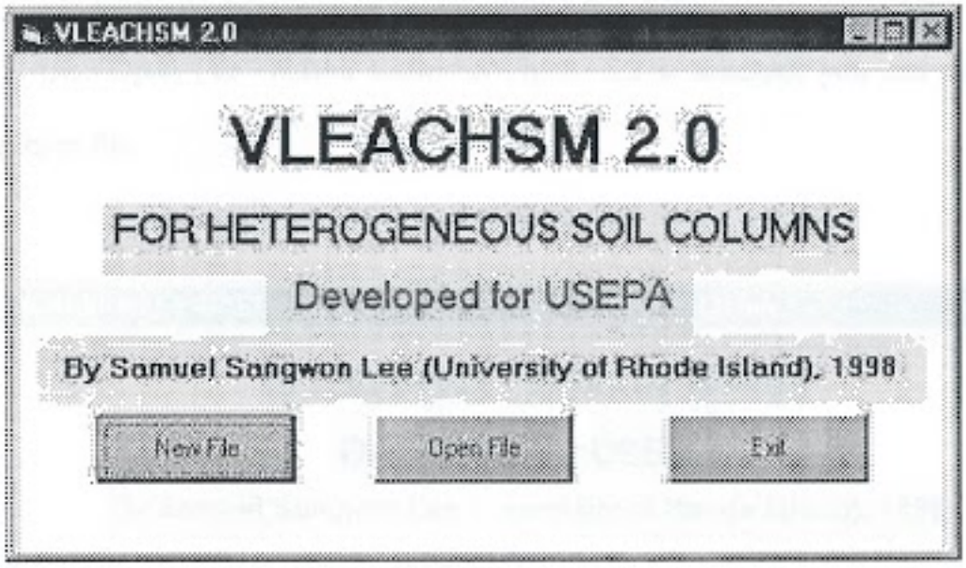


FIGURE 5.2 Opening Screen of New VLEACHSM 2.0

The "New File" button is for creating new input file. The "Open File" button is for using an existing input file.

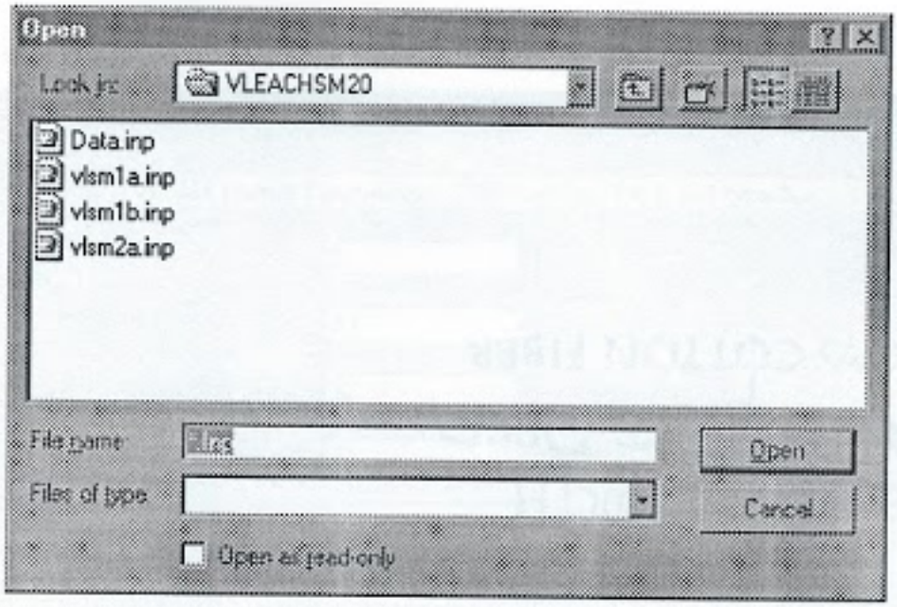


FIGURE 5.3 Open File Screen

If the "Open File" button shown in Figure 5.2 is selected, you can pick up an existing input file.

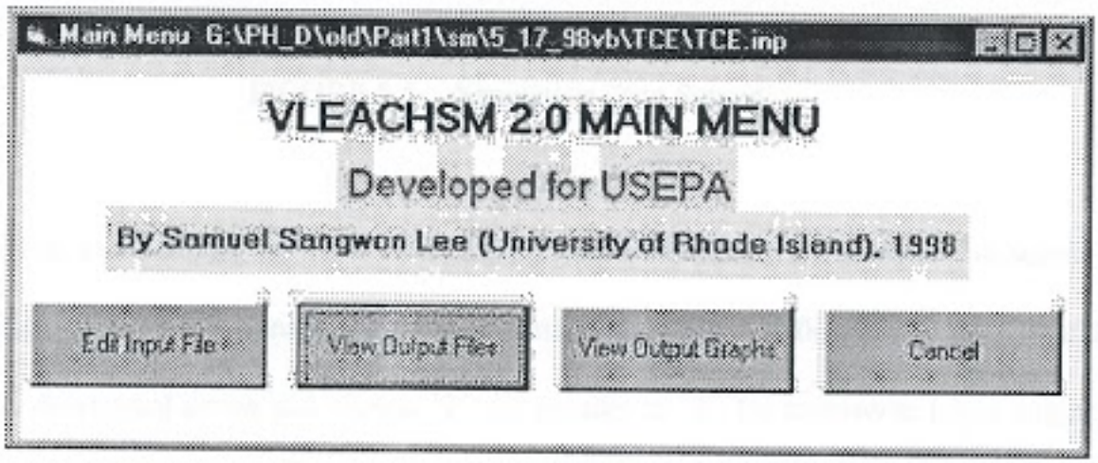


FIGURE 5.4 Main Menu

For a new simulation, you should choose "Edit Input File". Viewing output files or graphs can be done only after the simulation has been accomplished.

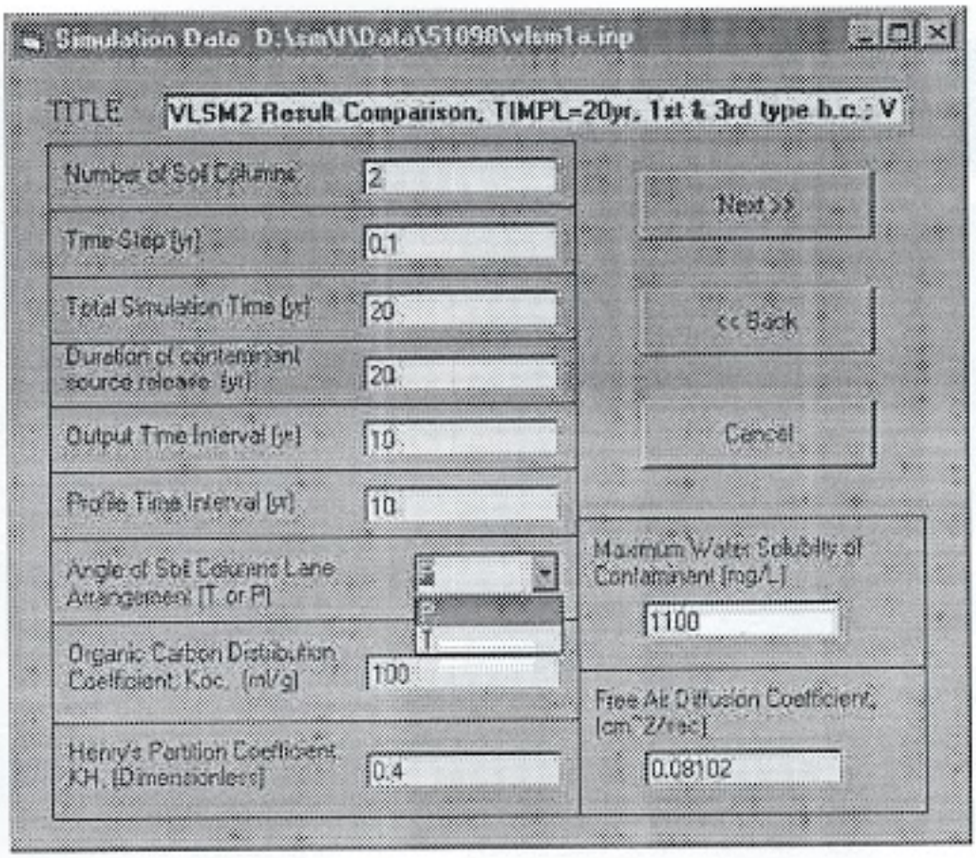


FIGURE 5.5 Simulation Data Screen

You can edit the input file by changing numerical values in the boxes on screen. For "Angle of Soil Columns Lane Arrangement to groundwater flow" box, you should click the downward arrow and choose "P" for parallel or "T" for transverse (right angle). You can go to the next menu by clicking "Next" when finished, or click "Back" to go back pick another file.

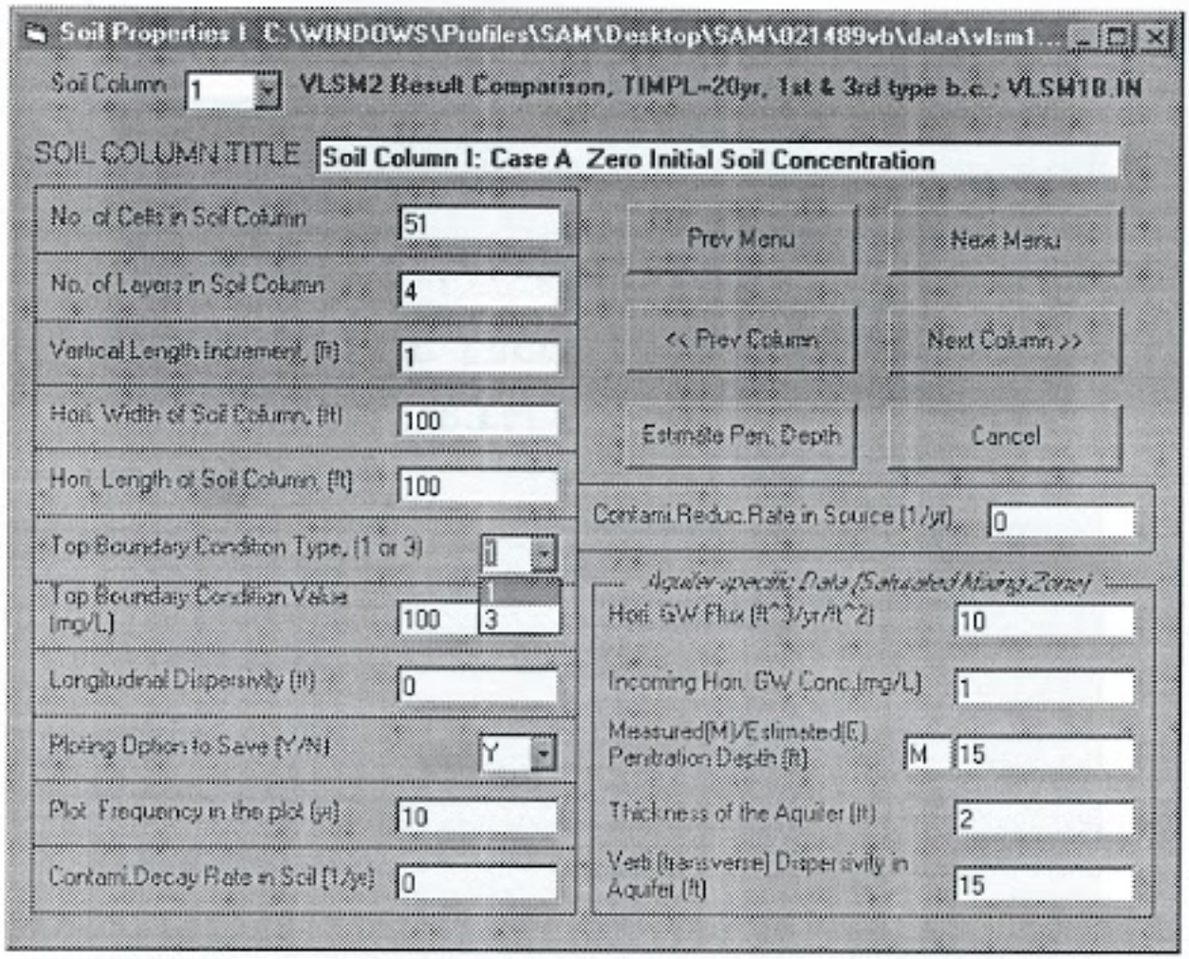


FIGURE 5.6 Soil Properties I Screen

Type or select input data in the boxes. If the penetration depth needs to be estimated, click "Estimated Pen. Depth" button. Then click "Next Column >>" to continue. When finished, click "Next Menu" button.

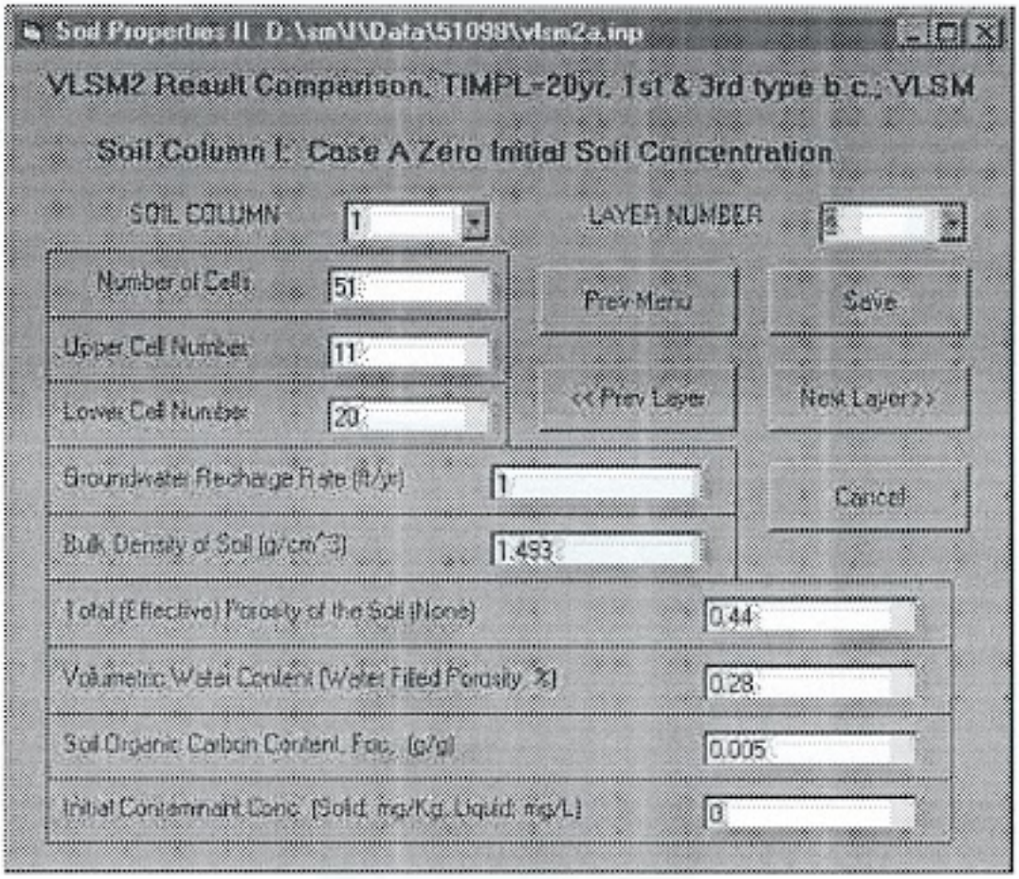


FIGURE 5.7 Soil Properties II Screen

Type or select input data in the boxes. Then click "Next Layer >>" to continue. When finished, click "Save".

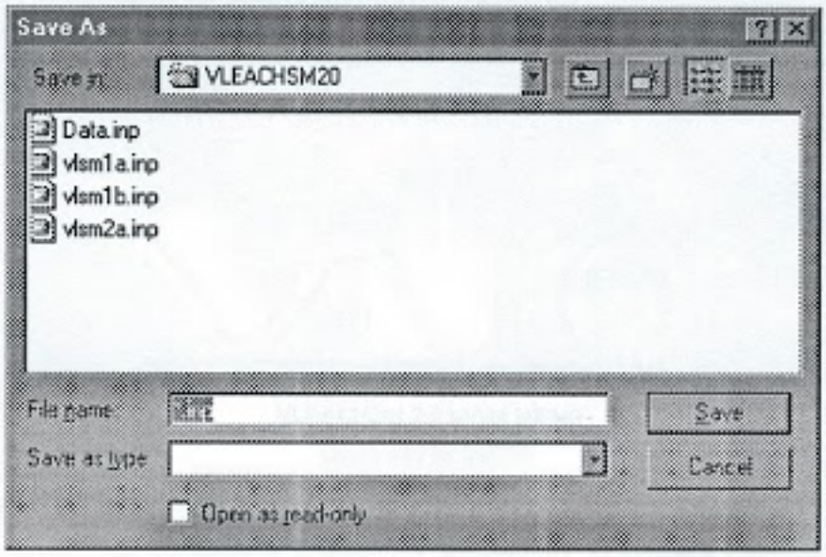


FIGURE 5.8 Save File Screen

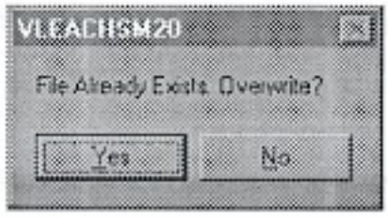


FIGURE 5.9 Overwrite Warning Screen

Type a new file name or overwrite onto an exiting file. Then, click "Save."

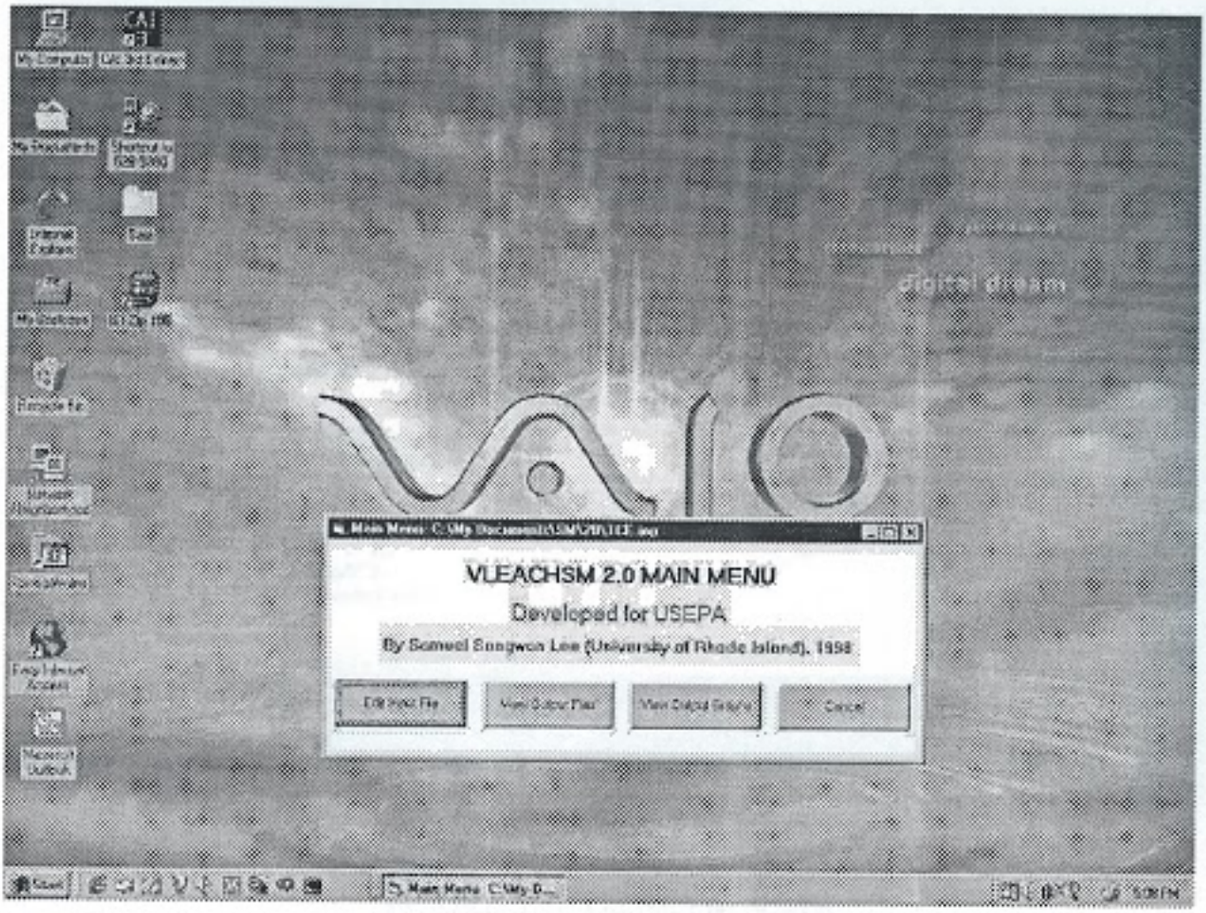


FIGURE 5.10 Run Main Menu

If you see a small "Vad" task in the taskbar on the bottom of the screen, wait until it disappears (in a second for a 100 MHz computer machine). Click "View Output Files" to view the contents of simulation output files or click "View Output Graphs" to plot 'Liquid-Phase Contaminant Concentration vs. Depth from Ground Surface', 'Mass Concentration vs. Time-Step', and so on. Click "Edit Input File" to modify the input data.

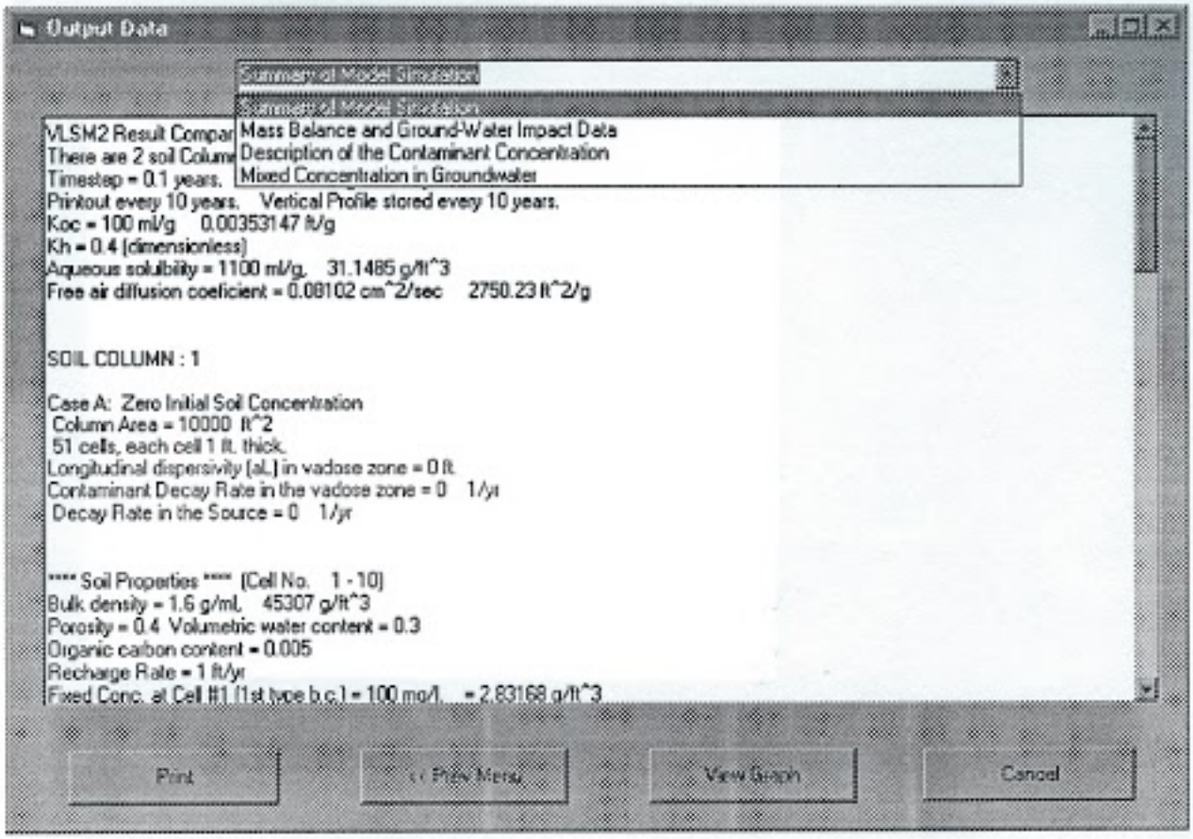


FIGURE 5.11 Output File Screen

Any one of four simulation output files can be selected by clicking the downward arrow next to the top box "Summary of Model Simulation." You can have print the output data shown the screen by clicking the "Print" button. The graphs of the output data can be viewed by clicking the "View Graph" button.

FIGURE 5.12 Initial Graph Screen

You can plot any one of seven kinds of graphs

"Liquid-P. Contaminant Conc. (mg/L) vs. Depth from Ground-Surface (ft)",

"Solid-P. Contaminant Conc. (mg/L) vs. Depth from Ground-Surface (ft)",

"Gas-P. Contaminant Conc. (mg/L) vs. Depth from Ground-Surface (ft)",

"Time (yr) vs. Mass Conc. (mg/L) in Vadose Zone",

"Time (yr) vs. Vadose Zone Leach. Conc. (mg/L) to GW",

"Time (yr) vs. Aquifer Conc. (mg/L) in Groundwater", or

"Time (yr) vs. Mixed Conc. (mg/L) in Groundwater"

by clicking the respective button.

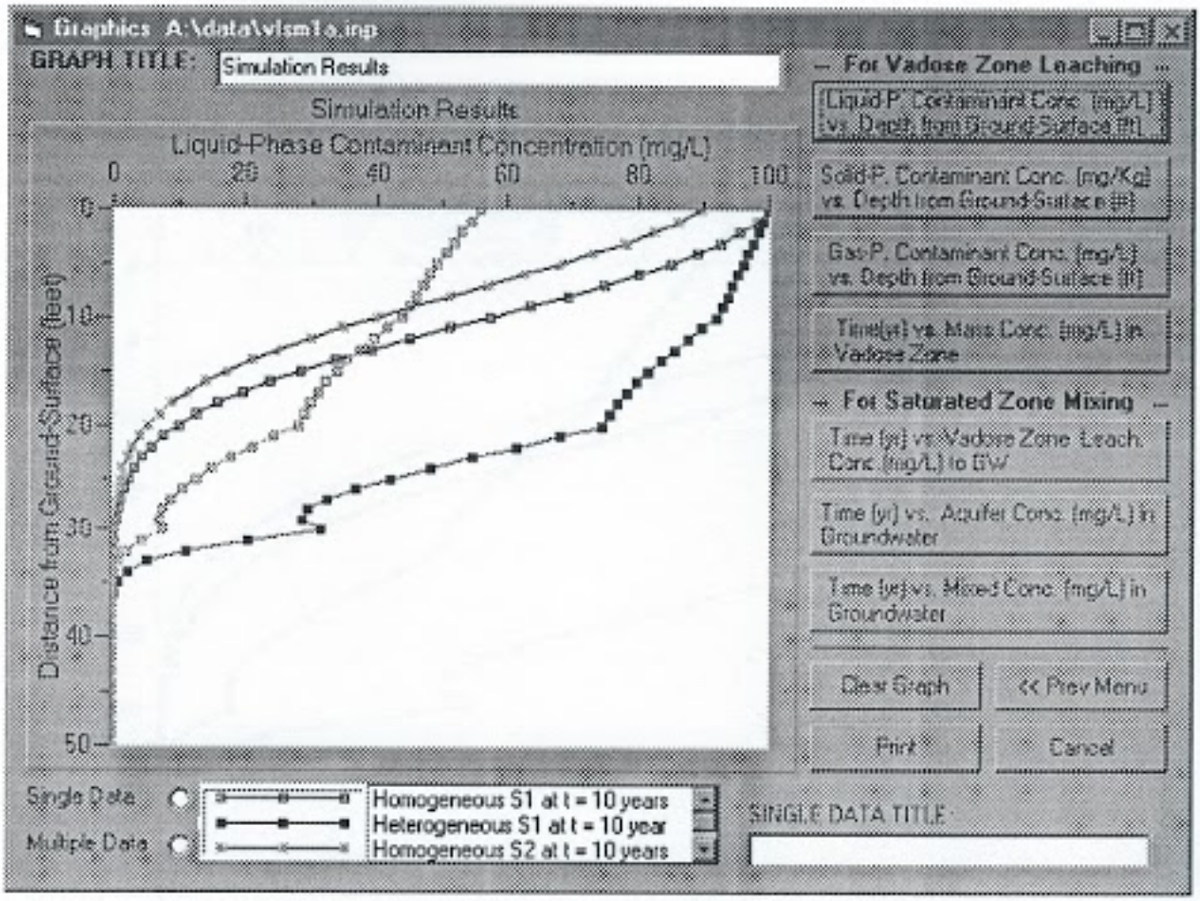


FIGURE 5.13 Example Plots of Graphed Data

The example plots (Figure 5.13) show liquid-phase contaminant concentration profiles for VLEACHSM 2.0 Case A and B, which are modified from those VLEACHSM 1.0a manual. You can put a title to the graph by typing in "Graph Title" box. If you want to plot only one data set, click in "Single Data" circle, and choose one data set title in the box on the right. If you want to plot multiple data sets, click in "Multiple Data" circle, and choose your desired data set titles one by one in the box on the right. The graph will remain on screen during any desired title changes. The graphs can be printed as desired by clicking the "Print" button.

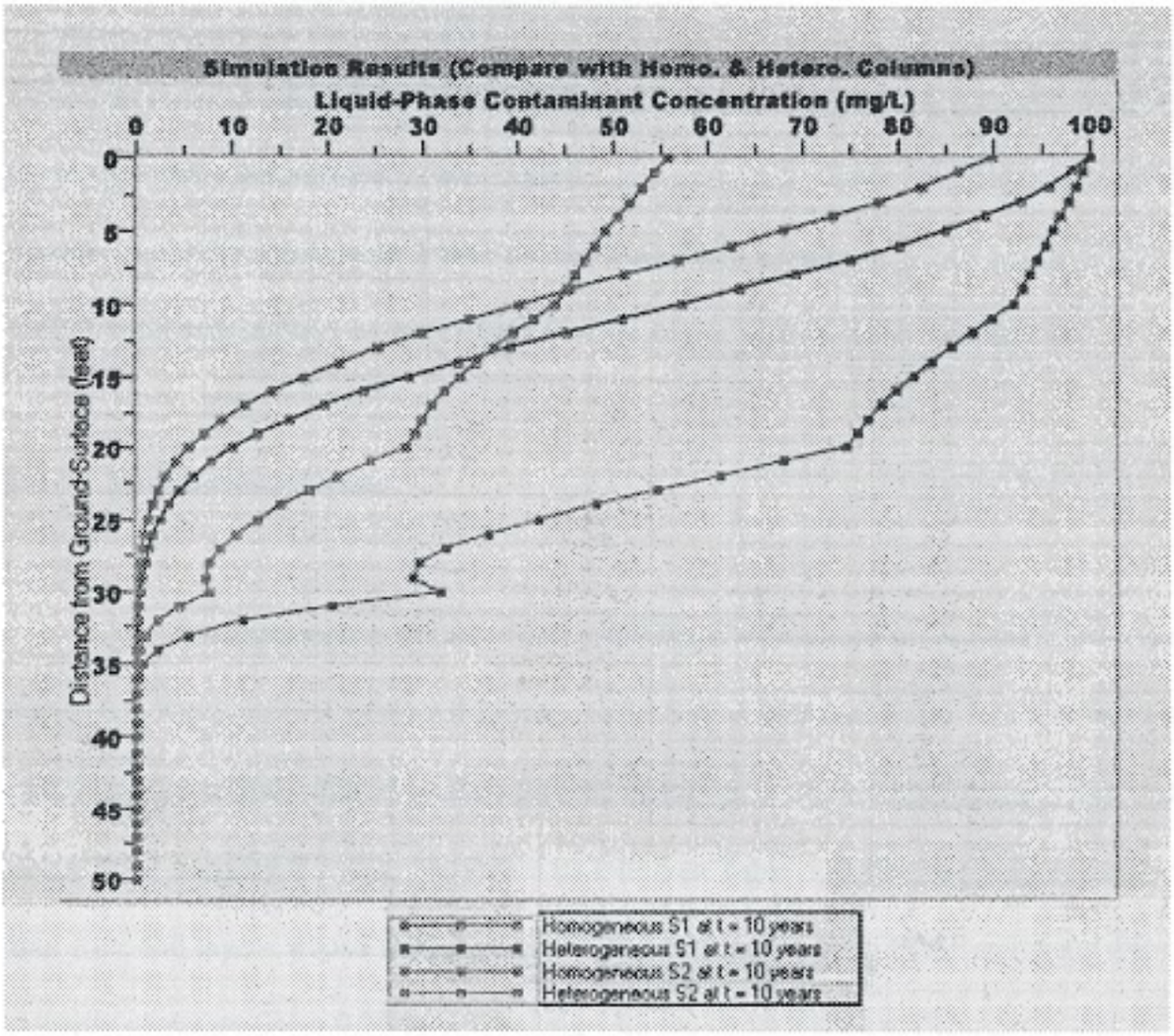


FIGURE 5.14 Printed Graph of Liquid-Phase Contaminant Concentration (mg/L) vs. Depth from Ground Surface (ft)

If you click the "Print" button in Figure 5.13, you can get the above hard copy of graph.

6.0 EXAMPLE PROBLEMS

In order to demonstrate VLEACHSM 2.0's ability to simulate actual soil columns, by incorporating heterogeneity, two examples are presented. These are similar to the soil column 1 and the soil column 2 with the same boundary conditions which are the 1st type and 3rd type, respectively, in the VLSM1.INP file of the VLEACHSM 1.0a manual.

6.1 Problem Descriptions

The data input equivalent to VLSM1.INP file is reproduced as VLSM1A.INP (see Table 5.3) in the VLEACHSM 2.0 format. In VLSM1A.INP (see Table 5.3) file, there is only one kind of layer with 51 pixels with bulk density $\rho_b = 1.6$, total porosity $n = 0.4$, and water-filled porosity $\theta = 0.3$. In VLSM1B.INP (see Table 5.4) file, the layer is divided into four soil layers whose total porosity decreases (from 0.44 to 0.38) and water-filled porosity increases (from 0.26 to 0.32) along with the depth shown in Figures 6.1 and 6.2. Figure 6.1 shows Case A where two soil columns are arranged transversely to the groundwater flow direction. Figure 6.2 shows Case B where two soil columns are arranged parallel to the groundwater flow direction. The contaminant concentration profiles in the vadose zone are the same between Case A and Case B, where only the saturated mixing zones are different. The bulk density is adjusted according to the total porosity change (contribution from the water content change is disregarded). The soil column 1 has 1st type top boundary condition. The soil column 2 (Case B) has 3rd type top

Case A: Transverse Profile

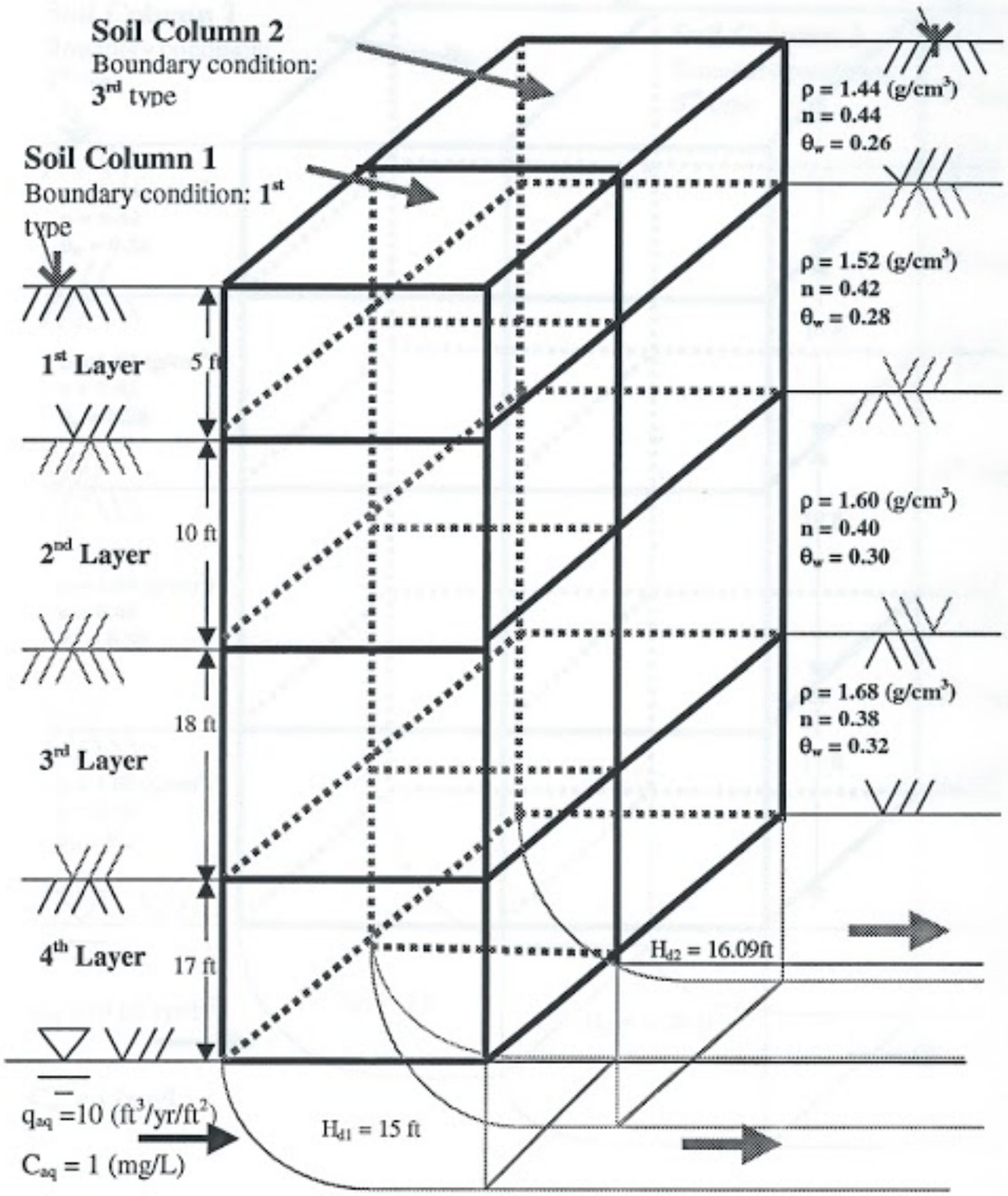


FIGURE 6.1 Profile of Transverse Column Example Problem in VLEACHSM version 2.0 (where Soil Column 1 is the 1st type of boundary condition. Soil Column 2 is the 3rd type boundary condition.)

Case B: Parallel Profile

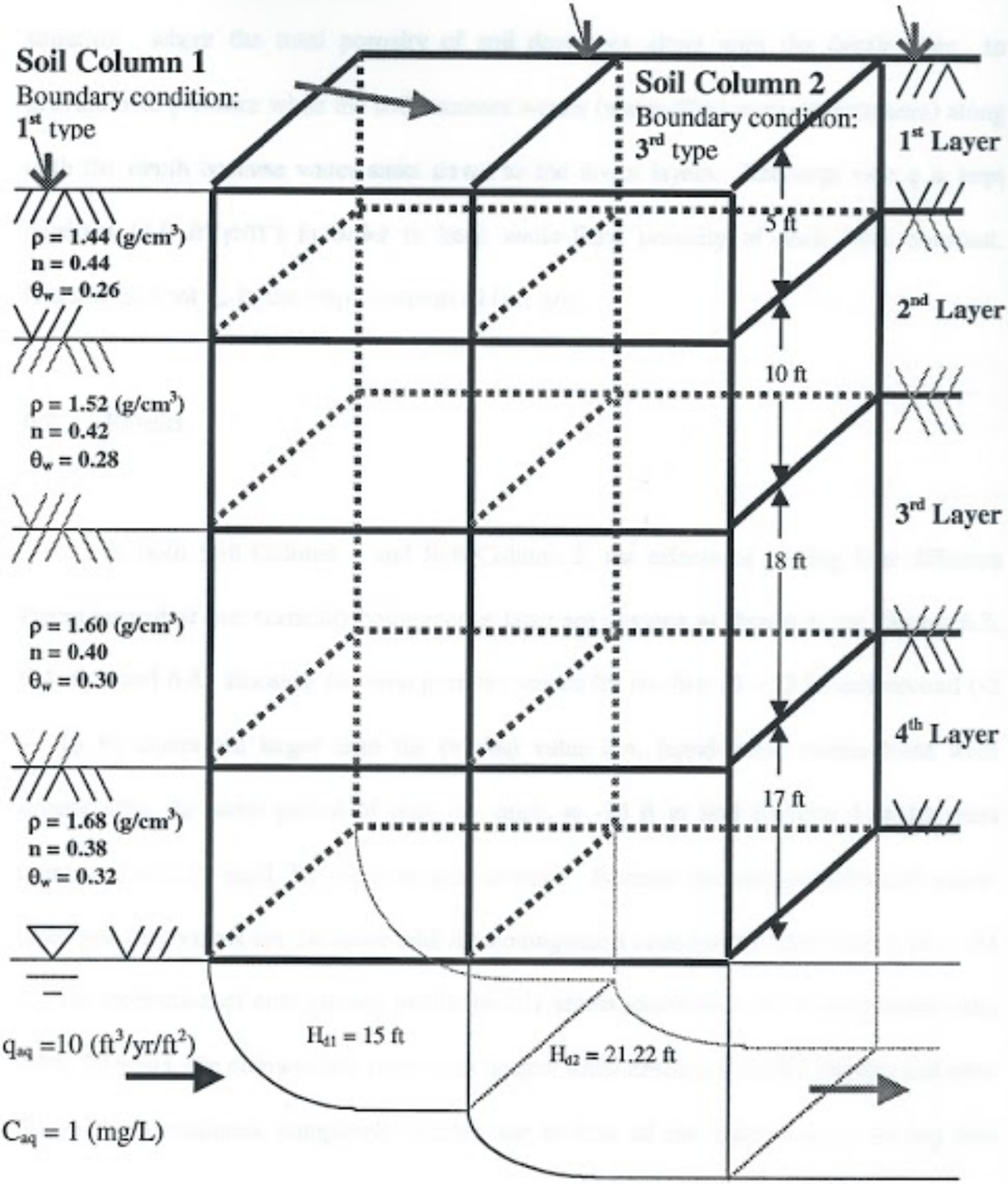


FIGURE 6.2 Profile of Parallel Column Example Problem in VLEACHSM version 2.0 (where Soil Column 1 is the 1st type of boundary condition. Soil Column 2 is the 3rd type boundary condition.)

boundary condition. This assumed set of parameters are derived from a geologic situation where the total porosity of soil decreases along with the depth due to gravitational pressure while the soil becomes wetter (water-filled porosity increases) along with the depth because water sinks down to the lower layers. Recharge rate q is kept constant (1.0 $ft^3/yr/ft^2$) in order to keep water-filled porosity of each layer constant. Organic content f_{oc} is also kept constant (0.005 g/g).

6.2 Results

In both Soil Column 1 and Soil Column 2, the effects of putting four different layers instead of one vertically homogenous layer are obvious as shown in see Figures 6.3, 6.4, 6.5, and 6.6. Because the total porosity values for the first (0 ~ -5 ft) and second (-5 ~ -15 ft) layers are larger than the original value 0.4, liquid-phase contaminant went deeper after the same period of time (71 mg/L at -15 ft in Soil Column 1 at 10 years compared with 28 mg/L for the homogenous case). Because the total porosity and water-filled porosity values are the same with the homogenous ones for the third layer (-15 ~ -33 ft), the inclination of contaminant profile mostly seems identical to the homogenous case. After 20 years, the contaminant penetrates deeper while keeping a similar profile, and after 30 years, contaminant completely reaches the bottom of the vadose zone, mixing into groundwater.

6.3 Discussion

These calculations clearly demonstrate that total soil porosity and its water-filled porosity have a strong effect on the liquid-phase contaminant concentration profile. VLEACHSM 2.0 can simulate such porosity change by assigning different porosity values for each layer. These simulations got same type of graphs in different shorter time steps. However, the program can also assign different values for recharge rate and organic carbon content, which may also affect the results, although there is no clear justification to make the recharge rate different among the soil layers because water-filled porosity is assumed to be constant.

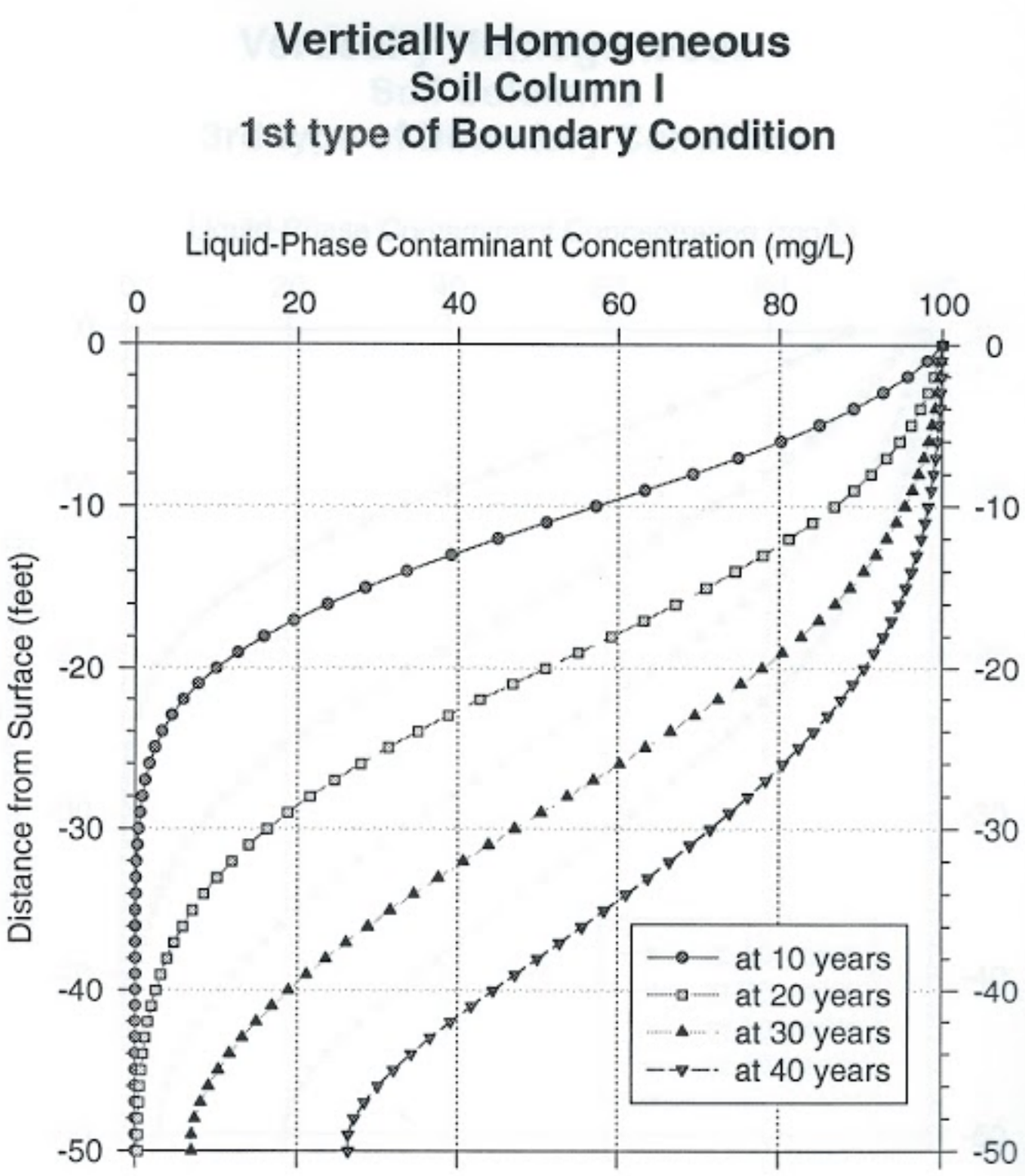


FIGURE 6.3 Simulation Results of Vertically Homogeneous Soil Column I, 1st type of Boundary Condition at 10, 20, 30, and 40 years of Liquid-Phase Contaminant Concentration (mg/L) vs. Depth from Ground Surface (ft)

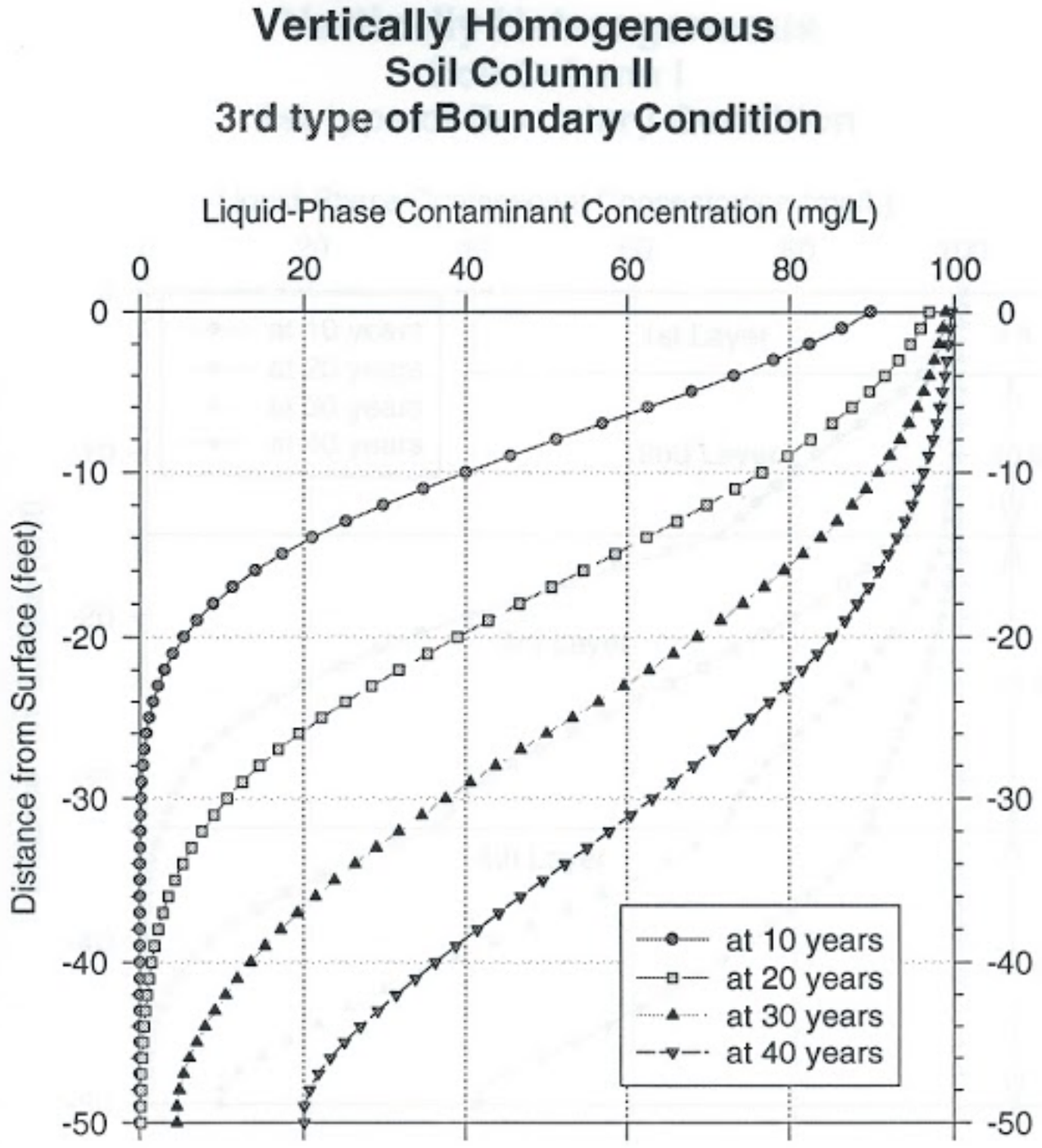


FIGURE 6.4 Simulation Results of Vertically Homogeneous Soil Column II, 3rd type of Boundary Condition at 10, 20, 30, and 40 years of Liquid-Phase Contaminant Concentration (mg/L) vs. Depth from Ground Surface (ft)

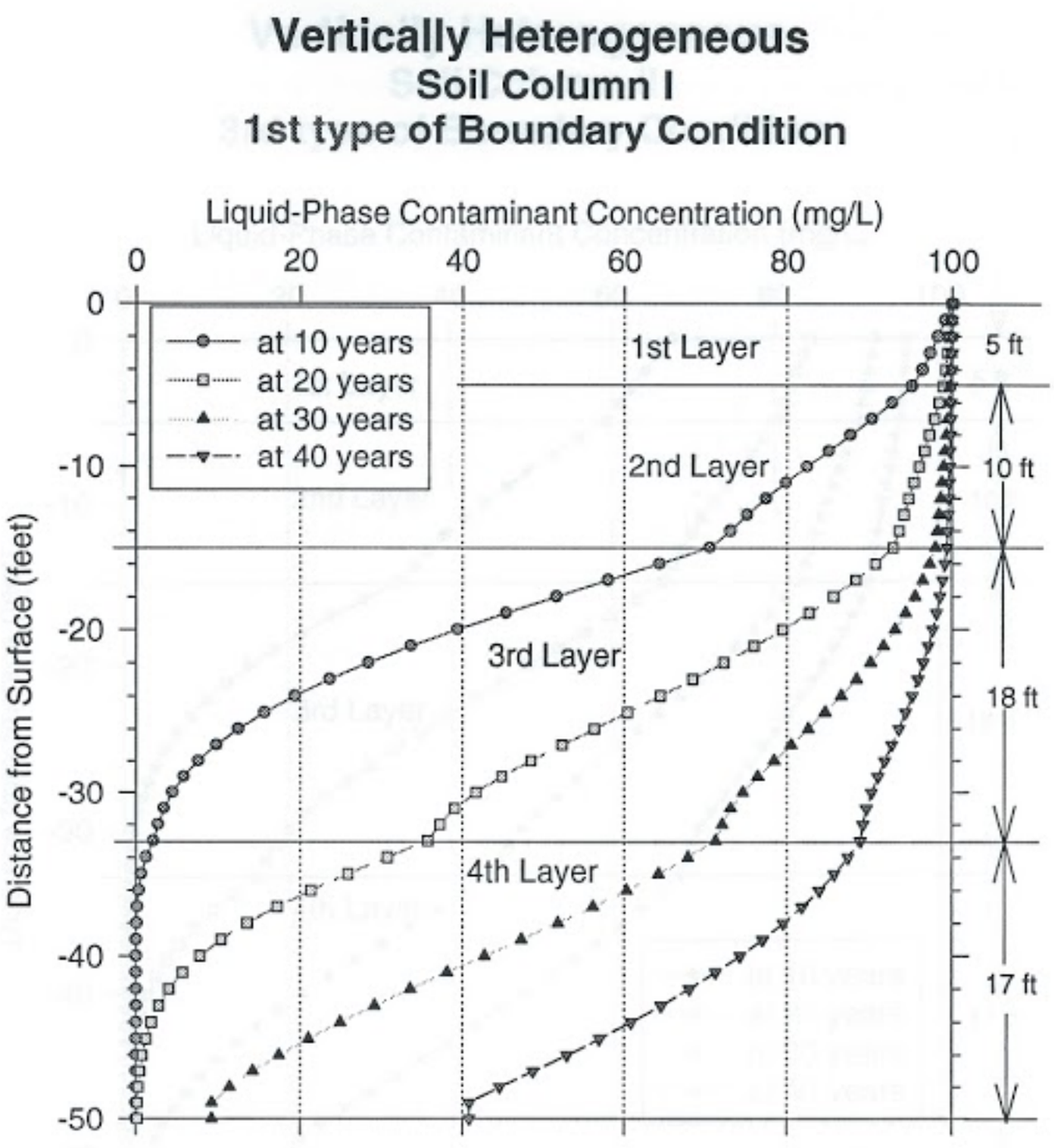


FIGURE 6.5 Simulation Results of Vertically Heterogeneous Soil Column I, 1st type of Boundary Condition at 10, 20, 30, and 40 years of Liquid-Phase Contaminant Concentration (mg/L) vs. Depth from Ground Surface (ft)

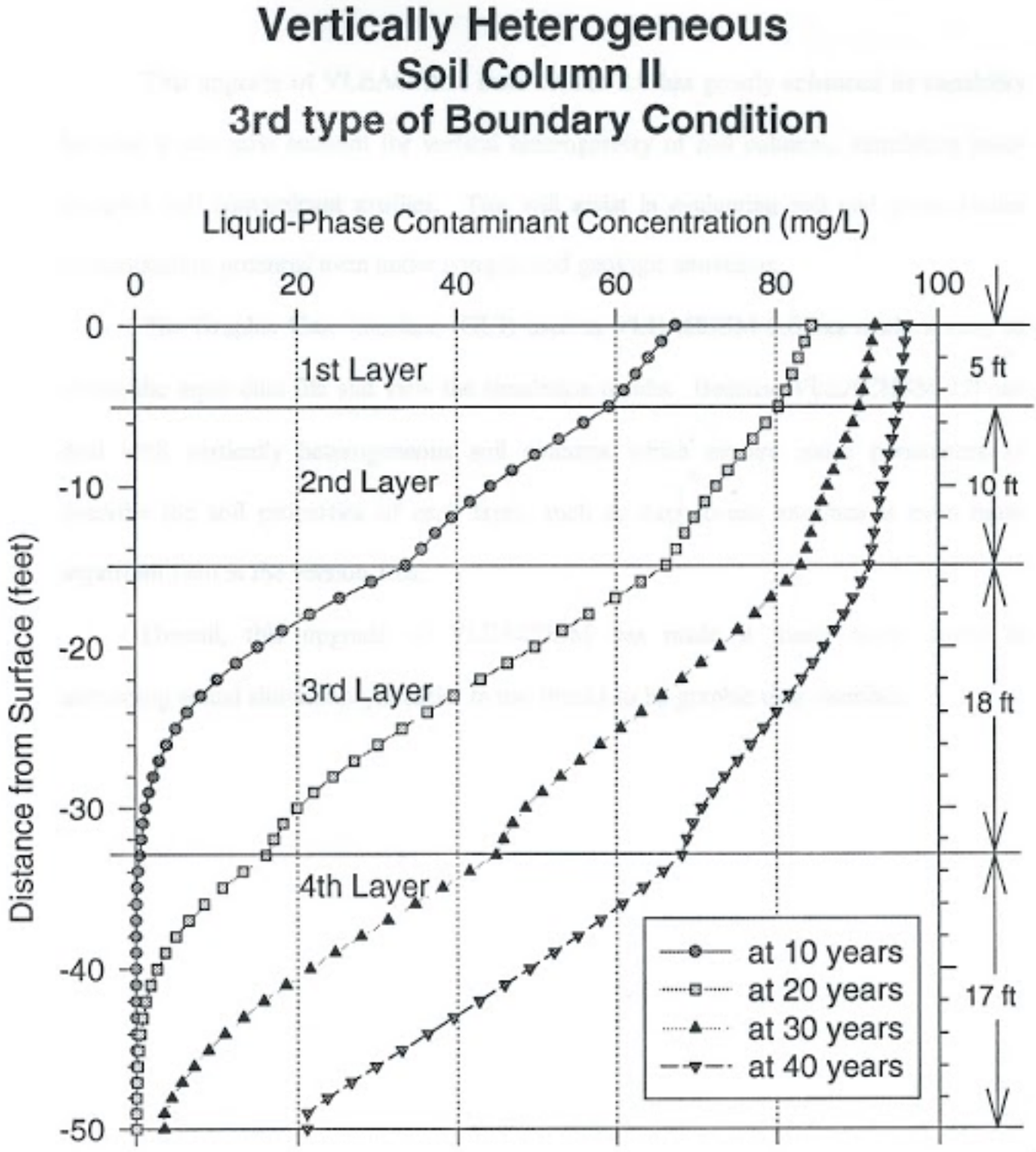


FIGURE 6.6 Simulation Results of Vertically Heterogeneous Soil Column II, 3rd type of Boundary Condition at 10, 20, 30, and 40 years of Liquid-Phase Contaminant Concentration (mg/L) vs. Depth from Ground Surface (ft)

7.0 SUMMARY

This upgrade of VLEACHSM from 1.0a to 2.0 has greatly enhanced its capability because it can now account for vertical heterogeneity of soil columns, simulating more complex soil contaminant profiles. This will assist in evaluating soil and groundwater contamination potential even under complicated geologic situations.

The Graphic User Interface (GUI) used in VLEACHSM 2.0 has made it easy to create the input data file and view the simulation results. Because VLEACHSM 2.0 can deal with vertically heterogeneous soil columns which require more parameters to describe the soil properties of each layer, such an easy-to-use interface is even more important than in the version 1.0a.

Overall, this upgrade of VLEACHSM has made it much more useful in simulating actual situations, yet easier to use thanks to its graphic user interface.

REFERENCES CITED

Ames, W. F., 1997, Numerical Methods for Partial Differential Equations, 2nd Ed., Academic Press, New York.

Beljin, M. S. and P. K. M. ven der Heijde, 1993, SOLUTE: A Program Package of Analytical Models for Solute Transport in Ground-Water, International Ground Water Modeling Center, Golden, CO.

Carsel, R. F. and R. S. parrish, 1988, "Variation within Texture Classes of Soil Water Characteristics," Water Res. Research. Vol. 24, pp. 755-769.

Cleary, R. W. and M. J. Ungs, 1994, PRINCE: Princeton Analytical Models of Flow and Transport, User Guide, Version 3.0, Waterloo Hydrogeologic Software, Waterloo, Ontario, Canada.

Dragun, J., 1988, The Soil Chemistry of Hazardous materials, Hazardous Materials Control Research Institute, Silver Spring, MD.

Falta, R. W.. I. Javandel, K. Pruess, and P. A. Witherspoon, 1989, " Density-driven Flow of Gas in the Unsaturated Zone Due to the Evaporation of Volatile Organic Compounds," Water Res. Research, Vol 25, pp. 2159-2169.

Gelhar, L. W., A. Mantoglou, C. Welty, and K. R. Rehfeldt, 1985, A Review of Field Scale Physical Solute Transport Processes in Saturated and Unsaturated Porous Media. Rep. EA-4190, Elec. Power Res. Inst., Palo Alto, Calif.

Hirsch, C., 1989, Numerical Computation of Internal and External Flows, Vol. 1, Fundamentals of Numerical Discretization. John Wiley & Sons, New York.

Howard, P. H., R. S. Boethling, W. F. Jarvis, W. M. Meylan, and E. M. Michalenko, 1991, Handbook of Environmental Degradation Rates, Lewis Publishers.

Jury, W. A., 1986, in Vadose Zone Modeling of Organic Pollutants, Chapter 11, eds. S. C. Hern and S. M. Melancon, Lewis Publishers, Chelsea, MI.

Lee, S. B., 1995, VLEACHSM: Vadose zone LEACHing and Saturated zone Mixing model (Version 1.0), Developed for USEPA, Robert S. Kerr Environmental Research Lab., Ada, OK.

Lee, S. B., 1996, A Screening Level Model for Estimation of Vadose Zone Leaching and Saturated Zone Mixing: VLEACHSM, North American Water and Environment Congress, ASCE.

Li, E. A., V. O. Shanholtz, and E. W. Carson, 1976, Estimating Saturated Hydraulic Conductivity and Capillary Potential at the Wetting Front, Department of Agricultural Engineers, Virginia Polytechnic Institute and State University, Blacksburg.

McCuen, R. H., W. J. Rawls, and D. L. Brakensiek, 1981, "Statistical Analysis of the Brooks-Corey and the Green-Ampt Parameter Across Soil Textures," Water Resources Research, vol. 17, no. 4, pp. 1005-1013.

Mitchel, A. R., D. F. Griffiths, 1980, The Finite Difference Method in Partial Differential Equations, John Wiley & Sons, New York.

Millington, R. J., 1959, "Gas Diffusion in Porous Media," Science, vol. 130, pp. 100-102.

Perry, R. H., D. W. Green, and J. O. Malony, 1984, Perry's Chemical Engineer's Handbook, 6th Ed., McGraw-Hill, Inc.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, 1992, Numerical Recipies in FORTRAN, 2nd Ed. Cambridge Univ. Press, NY.

Ravi, V. and J. A. Johnson, 1993, VLEACH: An One-Dimensional Finite Difference Vadose Zone Leaching Model, Developed for USEPA, R. S. Kerr Res. Lab., Ada, OK.

Ravi, V. and S. B. Lee, 1996, "One-Dimensional Analytical Solution for Contaminant Transport in the Subsurface," Poster presentation at the Emerging Technologies in Hazadous Waste Management VIII, I&EC Division, American Chemical Society, September 9-11, Birmingham, AL.

Rawls, W. J., 1983, "Estimating Bulk Density from Particle Size Analysis and Organic Matter Content," Soil Science, vol. 135, no. 2, pp. 123-125.

Sims, R. C., J. L. Sims, and S. G. Hansen, 1991, Soil Transport and Fate Database 2.0 and Model Management System, Dept. of Civil and Environmental Engineering, Utah State Univ., Logan, UT., Prepared for USEPA, Robert S. Kerr Environmental Research Laboratort, Ada, OK.

Summers, K., S. Gherini, and C. Chen, 1980, Methodology to Evaluate the Potential for Ground Water Contamination for Geothermal Fluid Releases, Prepared by Tetra Tech Inc. for USEPA/IERL, Cincinnati, OH, EPA-600/7-80-117.

Ünlü, K., M. W. Kemblowski, J. C. Parker, D. Stevens, P. K. Chong and I. Kamil, 1992, "A Screening Model for Effects of Land-disposed Wastes on Ground-water Quality,"

USEPA, 1989, Determining Soil Response Action Levels Based on Potential Contaminant Migration to Ground Water: A Compendium for Examples, Office of Emergency and Remedial Response, Washington D.C., EPA/540/2-89/057.

USEPA, 1990a, Background Document for EPA's Composite Model for Landfills (EPACML), Prepared by Woodward-Clyde Consultants for USEPA, Office of Solid Waste, Wash. D.C.

USEPA, 1990b, Subsurface Contamination Reference Guide, Office of Emergency and Remedial Response, Washington D.C., EPA/540/2-90/011.

USEPA, 1993, Compilation of Ground-Water Models, Prepared by P. van der Heijde and O.A. Elnawawy for Robert S. Kerr Environmental Research Laboratort, Ada, OK.

USEPA, 1994, Evaluation of Unsaturated/Vadose Zone Models for Superfund Sites, Office of Research and Development, Washington D.C., EPA/600/R-93/184.

APPENDIX A

C++ SOURCE CODES FOR VLEACHSM 2.0

By Samuel Sangwon Lee

1998

```
// ************************************************************
//
// Main.cpp
// A part of NEW VLEACHSM 2.0
// By Samuel Sangwon Lee (URI), 1998
//
// ************************************************************

#include "VadClass.h"
void main(int x, char **FName)
{
        vadoseClass V;
        V.VadoseOutputFiles();
}//end main

// ************************************************************
// VadClass.cpp
//
// ************************************************************

#include "VadClass.h"
float Abs(float Num)
{
        if (Num < 0)
                Num = Num * -1;
        return (Num);
}//end Abs

vadoseClass::vadoseClass()
{
        Vad.ReadInput("Data.inp");
}//end default constructor

vadoseClass::vadoseClass(char *FileName)
{
        Vad.ReadInput(FileName);
}//end default constructor

void vadoseClass::VadoseOutputFiles()
{
        OpenFiles();
        Prm.Output(Vad);
        MainLoop();
        CloseFiles();
}//end VadoseOutputFiles

void vadoseClass::OpenFiles()
{
        char FileName[255];
        int Len, Cut;
        strcpy(FileName, Vad.OutFileName);
        Len = strlen(FileName);
        Cut = Len - 3;
        FileName[Cut] = '\0';
        strcat(FileName, "prm");
        Prm.OpenFile(FileName);
        FileName[Cut] = '\0';
        strcat(FileName, "out");
        Out.OpenFile(FileName);
        FileName[Cut] = '\0';
        strcat(FileName, "prf");
        Prf.OpenFile(FileName);
        FileName[Cut] = '\0';
        strcat(FileName, "sim");
        Sim.OpenFile(FileName);
        FileName[Cut] = '\0';
        strcat(FileName, "gim");
        Gim.OpenFile(FileName);
        FileName[Cut] = '\0';
        strcat(FileName, "cmx");
        Cmx.OpenFile(FileName);
```

```
}//end OpenFiles

void vadoseClass::CloseFiles()
{
        Prm.CloseFile();
        Out.CloseFile();
        Prf.CloseFile();
        Sim.CloseFile();
        Gim.CloseFile();
        Cmx.CloseFile();
}//end CloseFiles

//BEGINNING OF MAIN CALCULATIONS AND OUTPUT

void vadoseClass::MainLoop()
{
        int i,j; //loop controls
        Out.SimTitle(Vad.SimTitle);
        for (i=0; i < Vad.Ntime; ++i)
        {
                Gtotal[i] = 0;
        }//end for i
        Cmx.Title(Vad.Angl);
        for (i=0; i < Vad.Nscol; ++i)
        {
                if (Vad.Angl == 'T')
                Cmx.ColInfo(Vad.Angl, i+1);
                VadMain(i);
                for(j=0; j < Vad.Ntime; ++j)
                {
                        Gwim[i][j] = Gwimp[j];
                        Gtot[i][j] = Gtotal[j];
                }//end forj
        }//end for i
        Gcum = 0;
        Out.TotalGWTitle();
        for(i=0; i < Vad.Ntime; ++i)
        {
                Gwimtl = 0;
                Gtottl = 0;
                for(j=0; j < Vad.Nscol; ++j)
                {
                        Gwimtl += Gwim[j][i];
                        Gtottl += Gtot[j][i];
                }//end for j
        Gcum += Gtottl;
        Out.TotalGWImpact(Vad.Ntime, (i+1)*Vad.Ptime, Gwimtl, Gcum);
        }//end for i
        if (Vad.Angl == 'P')
        {
                Cmx.ColInfo(Vad.Angl, 1);
                Qsum[0] = Vad.Volfxc[0] + Vad.Pdepth[0] * Vad.Width[0] * Vad.Flxaqf[0];
                for(j=0; j <= Vad.Ntimest; ++j)
        {
                Time = (j+1) * Vad.Delt;
                Cmixe[0][j] = (Vad.Pdepth[0] * Vad.Width[0] * Vad.Flxaqf[0]
                        * Vad.Cinpaq[0] + Vad.Volfxc[0] * Cliqend[0][j])/Qsum[0];
                Cmx.PAngleOutput_1(Time, Cliqend[0][j] / CONVERT_2,
                        Vad.Cinpaq[0] / CONVERT_2, Cmixe[0][j] / CONVERT_2);
        }//end for j
        for(i=1; i < Vad.Nscol; ++i)
        {
                Cmx.ColInfo(Vad.Angl, (i+1));
                Qsum[i] = Vad.Volfxc[i] + Vad.Pdepth[i]
                                * Vad.Width[i] * Vad.Flxaqf[i] + Qsum[i-1];
                for(j=0; j <= Vad.Ntimest; ++j)
                {
                        Time = (j+1) * Vad.Delt;
                        Cmixe[i][j] = (Vad.Pdepth[i] * Vad.Width[i] * Vad.Flxaqf[i]
                                * Vad.Cinpaq[i] + Vad.Volfxc[i] * Cliqend[i][j]
                                + Cmixe[i-1][j] * Qsum[i-1]) / Qsum[i];
```

```
                        Cmx.PAngleOutput_2(Time, Cliqend[i][j] / CONVERT_2,
                                Vad.Cinpaq[i] / CONVERT_2, Cmixe[i-1][j] / CONVERT_2,
                                Cmixe[i][j] / CONVERT_2);
                }//end for j
        }//end for i
}//end if P
Sim.Terminate();
Gim.Terminate();
}//end MainLoop

void vadoseClass::VadMain(int Col)
{
        int i,j, Ncel;
        int count = 0;
        Nceldm = Vad.Ncell[Col];
        Ncel = Vad.Ncell[Col] - 1;
        for(i=0; i <= Ncel; ++i)
{
        Cgas[i] = 0;
        Cliq[i] = 0;
        Cs[i] = 0;
}//end for i
for(i=0; i<Vad.GrNum[Col]; ++i)
{
        for(j=(Vad.J1[Col][i]-1); j < Vad.J2[Col][i]; ++j)
        {
                if(Vad.Kd[Col][j] == 0)
                {
                        Cliq[j] = Vad.Xcon[Col][i] * CONVERT_2;
                        Cs[j] = 0;
                }//end if
                else
                {
                        Cs[j] = float(Vad.Xcon[Col][i] * 0.000001);
                        Cliq[j] = Cs[j] / Vad.Kd[Col][j];
                }//end else
        }//end for j
}//end for i
if (Vad.Typtbc[Col] == 1)
{
        Cs[0]   = Vad.Vatbc[Col] * Vad.Kd[Col][0];
        Cliq[0] = Vad.Vatbc[Col];
}//end if
//Call Equil function
VadEquil(Col);
Out.SoilTitle(Vad.SoilTitle[Col]);
if (Iflag > 0)
{
        Out.Warn(0, Iflag);
}//end if
Out.AtTime((Col+1), 0, Mtotal, Mgtot, Mltot, Mstot);
Mt0    = Mtotal;
Mtp    = Mtotal;
MdfSum = 0;
MatCum = 0;
MatInt = 0;
MabCum = 0;
MabInt = 0;
MdtCum = 0;
MdtInt = 0;
MdbCum = 0;
MdbInt = 0;
MdcCum = 0;
MdcInt = 0;
Itime  = 0;
Prf.Title(Vad.SoilTitle[Col], 0);
for(i=0; i < Nceldm; ++i)
{
        Depthz = i * Vad.Delz[Col];
        Prf.Output(Depthz, Cgas[i]/CONVERT_2, Cliq[i]/CONVERT_2, float(Cs[i]*(1e+6)));
```

```
}//end for i
Cofm = Vad.Delt / (2*Vad.Delz[Col]*Vad.Delz[Col]);
Cofn = Vad.Delt / (4*Vad.Delz[Col]);
if (Vad.Decy1[Col])
        Li = Vad.Delt / 2*Vad.Decy1[Col];
else
        Li = 0;
Mi[0] = Cofm / Vad.Theta[Col][0]*Vad.Disp[Col][0];
Ni[0] = Cofn / Vad.Theta[Col][0]*Vad.Q[Col][0];
Mip[0] = 0;
Nip[0] = 0;
for(i=1; i<(Nceldm-1); ++i)
{
        Cofmi = Cofm / Vad.Theta[Col][i];
        Cofni = Cofn / Vad.Theta[Col][i];
        Mi[i] = Cofmi * Vad.Disp[Col][i];
        Ni[i] = Cofni * Vad.Q[Col][i];
        Mip[i] = Cofmi*( Vad.Disp[Col][i+1] - Vad.Disp[Col][i-1] ) / 4.0;
        Nip[i] = Cofni*( Vad.Q[Col][i+1] - Vad.Q[Col][i-1] );
}//end for i
Mi[Ncel] = Cofm / Vad.Theta[Col][Ncel] * Vad.Disp[Col][Ncel];
Ni[Ncel] = Cofn / Vad.Theta[Col][Ncel] * Vad.Q[Col][Ncel];
Mip[Ncel] = 0;
Nip[Ncel] = 0;
for(i=1; i<(Nceldm-1); ++i)
{
        A[i] = -Mi[i] + Mip[i] - Ni[i];
        B[i] = 1 + 2*Mi[i] + Nip[i] + Li;
        C[i] = -Mi[i] - Mip[i] + Ni[i];
}//end for i
Areas = Vad.Width[Col] * Vad.Pdepth[Col];
if (Vad.Plt[Col] == 'Y')
        Gim.Title(Col);
Iptime  = 1;
Ipltime = 1;
Iprtim = 1;

//Loop Begins

do
{
        ++Itime;
        Time = Itime * Vad.Delt;
        if (Time > Vad.Timpl)
                Vad.Vatbc[Col] = 0;
//Top boundary conditions
        B[0] = 1.;
        Ddzmpn1 = Vad.Disp[Col][0] / ( Vad.Delz[Col]*( Mi[0] + Ni[0]) );
        Aaa1 = Ddzmpn1 * ( 2.*Mi[0] + Li + 1 ) / 4.;
        Bbb1 = Ddzmpn1 * Mi[0] / 2;
        Ccc1 = Ddzmpn1 * ( 2.*Mi[0] + Li - 1. ) / 4.;
        Phi = Aaa1 + Vad.Q[Col][0] / 2.;
        Psi = Bbb1;
        Omg = Ccc1 + Vad.Q[Col][0] / 2.;
        if(Vad.Typtbc[Col] == 1)
        {
                C[0] = 0.;
                Rhs[0] = Vad.Vatbc[Col] * float(exp(-Vad.Decy2[Col] * Time));
        }
        else if(Vad.Typtbc[Col] == 3)
        {
                C[0] = -Psi / Phi;
                Rhs[0] = (-Omg*Cliq[0]+Psi*Cliq[1]+Vad.Q[Col][0]
                        *Vad.Vatbc[Col]*float(exp(-Vad.Decy2[Col]*Time)))/Phi;
        }
        else
                cout << "Check type of top boundary condition !!!\n";
//Bottom boundary conditions type 2: Cn - Cn-1 = 0
                Valbbc = 0;
                Rhs[Ncel] = Valbbc;
                B[Ncel] = 1.;
```

```
                A[Ncel] = -1.;
//Values between the top and bottom
                for(i=1; i<Ncel; ++i)
        {
        Rhs[i] = -A[i] * Cliq[i-1] + ( 2. - B[i] ) * Cliq[i] - C[i]*Cliq[i+1];
        }//end for i
//Flux calculations
        Ddzmpnn = Vad.Disp[Col][Ncel] / ( Vad.Delz[Col] * ( Mi[Ncel] + Ni[Ncel] ) );
        Aaan = Ddzmpnn*(2.*Mi[Ncel] + Li + 1.) / 4.;
        Bbbn = Ddzmpnn*Mi[Ncel] / 2.;
        Cccn = Ddzmpnn*(2.*Mi[Ncel] + Li - 1.) / 4.;
        Mat00 = Vad.Q[Col][0] * Cliq[0];
        Mdt00 = Ccc1*Cliq[0] - Bbb1*Cliq[1];
        Mab00 = Vad.Q[Col][Ncel] * Cliq[Ncel];
        Mdb00 = Cccn*Cliq[Ncel-1] - Bbbn*Cliq[Ncel];
//Call to Tridia
        VadTridia(A, B, C, Rhs, Cliq, Ncel);
//Top b' dry flux calculations by Adevection
        Mat = Vad.Delt * (Vad.Q[Col][0] * Cliq[0] + Mat00) / 2.;
//Top b' dry flux calculations by Dispersion
        Mdt = Vad.Delt * (Mdt00 + Aaa1*Cliq[0] - Bbb1*Cliq[1]);
//Bottom b' dry flux calculations by Adevection
        Mab = -Vad.Delt * (Vad.Q[Col][Ncel]*Cliq[Ncel] + Mab00) / 2.;
//Bottom b' dry flux calculations by Dispersion
        Mdb = -Vad.Delt * (Mdb00 + Aaan*Cliq[Ncel-1] - Bbbn*Cliq[Ncel]);
//Mass equilibration step
        Mttold = Mtotal;
        VadEquil(Col);
        if (Iflag > 0)
        {
                Out.Warn(0, Iflag);
        }//end if
        Mdiff  = Mtotal - Mttold;
        MdfSum += Mdiff;
//Mass Decay in the soil column
        Mdc    = -Vad.Decy1[Col] * Mtotal * Vad.Delt;
//Mass Balance calculations
        MatCum += Mat;
        MabCum += Mab;
        MdtCum += Mdt;
        MdbCum += Mdb;
        MdcCum += Mdc;
        MatInt += Mat;
        MabInt += Mab;
        MdtInt += Mdt;
        MdbInt += Mdb;
        MdcInt += Mdc;
//Write Groundwater impact data to plot file
        if (Vad.Plt[Col] == 'Y')
        {
                Gim.NumOutput(Time, (-Vad.Areac[Col] * (Mab + Mdb) / Vad.Delt));
        }//end if
//Write soil concentration - depth data to plot file
        Pltim = Vad.Pltime[Col] * Ipltime;
        if ((Vad.Plt[Col] == 'Y') &&
                (Abs(Time-Pltim) < (1e-6)))
        {
                ++Ipltime;
                Sim.Title((Col+1), Time);
                for(i=0; i < Nceldm; ++i)
                {
                        Sim.Output(float(Cliq[i] / CONVERT_2), (-(i) * Vad.Delz[Col]));
                }//end for i
        }//end if
        Ptime1 =  int(Vad.Ptime * Iptime);
        if (Abs(Time - Ptime1) < (1e-6))
        {
                Gwimp[count]  = -Vad.Areac[Col] * (Mab + Mdb) / Vad.Delt;
                Gtotal[count] = -Vad.Areac[Col] * (MabInt + MdbInt);
        //Call to Output
                VadOutput(Col);
```

```
                ++Iptime;
                ++count;
                MdfSum = 0;
        }//end if
//Write vert. concent profiles
        Prtim = int(Vad.Prtime * Iprtim);
        if (Abs(Time-Prtim) < 1e-6)
        {
                ++Iprtim;
                Prf.Title(Vad.SoilTitle[Col], Time);
                for(i=0; i < Nceldm; ++i)
                {
                        Depthz = i * Vad.Delz[Col];
                        Prf.Output(Depthz,Cgas[i]/CONVERT_2, Cliq[i]/
                                CONVERT_2, float(Cs[i]   * (1e+6)));
                }//end for i
        }//end if
        if (Vad.Angl == 'T')
        {
                Qaqf   = Vad.Plxaqf[Col] * Areas;
                Cmixed = (Qaqf*Vad.Cinaqf[Col]+Cliq[Ncel]*Vad.Areac[Col]
                        *Vad.Q[Col][Ncel])/(Qaqf+Vad.Areac[Col]*Vad.Q[Col][Ncel]);
                Cmx.RAngleOutput(Time,Cliq[Ncel]/CONVERT_2,
                        Vad.Cinaqf[Col]/CONVERT_2, Cmixed / CONVERT_2);
        }//end if Angl==T
        else
        {
                Cliqend[Col][Itime-1] = Cliq[Ncel];
        }//end else
}while(Time < Vad.Stime);
//Loop Ends
Out.GWImpact((Col+1), Vad.Ntime, Vad.Ptime, Gwimp, Vad.Areac[Col]);
}//end VadMain
void vadoseClass::VadOutput(int Col)
{
float Del1, Del2;
Out.AtTime((Col+1), Time, Mtotal, Mgtot, Mltot, Mstot);
Out.SinceLast(Time - Vad.Ptime);
Del1 = Mtotal - Mtp;
Del2 = MatInt + MdtInt + MabInt + MdcInt + MdbInt;
Out.Change(Del1, MatInt + MdtInt, MabInt + MdbInt, MdcInt, Del2);
Out.SinceBegin(0);
Del1 = Mtotal - Mt0;
Del2 = MatCum + MdtCum + MabCum + MdbCum + MdcCum;
Out.Change(Del1, MatCum + MdtCum, MabCum + MdbCum, MdcCum, Del2);
MatInt = 0;
MdtInt = 0;
MabInt = 0;
MdbInt = 0;
MdcInt = 0;
Mtp    = Mtotal;
}//end VadOutput
void vadoseClass::VadEquil(int Col)
{
Mgtot  = 0;
Mltot  = 0;
Mstot  = 0;
Mtotal = 0;
Iflag  = 0;
for(int i=0; i < Nceldm; ++i)
{
        Ml      = Cliq[i] * Vad.Delz[Col] * Vad.Thetaw[Col][i];
        Cgas[i] = Cliq[i] * Vad.Kh;
        Mg      = Cgas[i] * Vad.Delz[Col] * Vad.Pora[Col][i];
        Cs[i]   = Cliq[i] * Vad.Kd[Col][i];
        Ms      = Cs[i] * Vad.Delz[Col] * Vad.Rhob[Col][i];
        Mgtot   += Mg;
        Mltot   += Ml;
        Mstot   += Ms;
        Mt      = Mg + Ml + Ms;
```

```
        if (i==1)
                Mtotal1 = Mt;
        Mtotal += Mt;
        if (Cliq[i] > Vad.Cmax)
                ++Iflag;
}//end for i
}//end VadEquil
void vadoseClass::VadTridia(vadNumType A[], vadNumType B[], vadNumType C[],
                vadNumType F[], vadNumType X[], int N)
{
//Variables
vadNumType Alpha,
                   Beta[1000],
                   Y[1000];
int i;
//Calculating beta and y arrays
Alpha = B[0];
Y[0] = F[0] / Alpha;
for(i=1; i<=N; ++i)
{
        Beta[i-1] = C[i-1] / Alpha;
        Alpha = B[i] - A[i]*Beta[i-1];
        Y[i] = ( F[i] - A[i] * Y[i-1] ) / Alpha;
}//end for i
//Back substitution to calculate x array
X[N] = Y[N];
for(i=(N-1); i>=0; --i)
        X[i] = Y[i] - Beta[i]*X[i+1];
}//end VadTridia

// ************************************************************
//
// VadInput.cpp
//
// ************************************************************

#include "VadInput.h"
vadoseInput::vadoseInput()
{
}//end default constructor
vadoseInput::vadoseInput(char *FileName)
{
        ReadInput(FileName);
}//end constructor
void vadoseInput::ReadInput(char *FileName)
{
        int i, j, k;
        char ch;
        float fl1,fl2;
        vadNumType  Qj,Rhobij,Rhobj,Porj,Poraj,Thetawj,Thetaj,Pocj,Dispj,Kdj,Darj;
        ifstream In(FileName);
        In.getline(OutFileName, MAX_CHAR - 1, '\n');
        In.getline(SimTitle, MAX_CHAR - 1, '\n');
        In >> Nscol;
        In >> Delt >> Stime >> Timpl >> Ptime >> Prtime >> Angl;
        In >> Koci >> Kh >> Cmaxi >> Dairi;
        Koc  = Koci / CONVERT_1;
        Cmax = Cmaxi * CONVERT_2;
        Dair = Dairi * CONVERT_3;
        Ntime   = int(Stime/Ptime);
        Ntimest = int(Stime/Delt);
//WRITING IN SOIL COLUMN DATA FOR EACH NSCOL
        for(i=0; i < Nscol; ++i)
        {
                In.get(ch);
                In.get(ch);
                In.getline(SoilTitle[i], MAX_CHAR - 1, '\n');
                In>>Ncell[i]>>Delz[i]>>Width[i]>>Length[i]>>Typtbc[i]>>
                        Valtbc[i]>>Aldisp[i];
                In >> Plt[i] >> Pltime[i] >> Decy1[i] >> Decy2[i];
                In >> GrNum[i];
```

```
                Areac[i] = Width[i] * Length[i];
                Vatbc[i] = Valtbc[i]  * CONVERT_2;
        //WRITING IN AQUIFER DATA FOR EACH NCELL
                for(j=0; j < GrNum[i]; ++j)
                {
                        In >> J1[i][j] >> J2[i][j] >> Qj >> Rhobij >> Porj >>
                                Thetawj >> Focj >>Xcon[i][j];
                        Rhobj = Rhobij * CONVERT_1;
                        Kdj = Koc * Focj;
                        Poraj = Porj - Thetawj;
                        Darj = float(Dair * pow(Poraj,7./3.) / (Porj*Porj));
                        Thetaj = Thetawj + Poraj*Kh + Rhobj*Kdj;
                        Dispj = Aldisp[i]*Qj + Poraj*Darj*Kh;
                        for(k=(J1[i][j]-1); k<J2[i][j]; ++k)
                        {
                                Rhob[i][k] = Rhobj;
                                Kd[i][k] = Kdj;
                                Por[i][k] = Porj;
                                Pora[i][k] = Poraj;
                                Foc[i][k] = Focj;
                                Thetaw[i][k] = Thetawj;
                                Dar[i][k] = Darj;
                                Theta[i][k] = Thetaj;
                                Disp[i][k] = Dispj;
                                Q[i][k] = Qj;
                        }//end for k
                }//end for j
                In >> Flxaqf[i] >> Cinaqfi[i] >> Pdepth[i] >> ch >> fl1 >> fl2;
                Cinaqf[i] = Cinaqfi[i] * CONVERT_2;
                Cinpaq[i] = Cinaqf[i];
                Volfxc[i] = Q[i][Ncell[i] - 1] * Areac[i];
        }//end for i
        In.close();
}//end ReadInput

// ************************************************************
//
// OutOutput.cpp
//
// ************************************************************

#include "OutOutput.h"
outClass::outClass()
{
}//end default constructor
outClass::outClass(char *FileName)
{
        OpenFile(FileName);
}//end constructor
void outClass::OpenFile(char *FileName)
{
        Out.open(FileName);
}//end OpenFile
void outClass::CloseFile()
{
        Out.close();
}//end CloseFile
void outClass::SimTitle(char *Title)
{
        Out << Title << endl;
}//end SimTitle
void outClass::Warn(float Time, float Iflag)
{
        Out << "WARNING!!!  At time " << Time << " aqueous solubility was exceeded in " <<
                Iflag << " cells." << endl;
}//end Warn
void outClass::SoilTitle(char *Title)
{
        Out << endl << endl;
        Out << Title << endl;
}//end SoilTitle
```

```
void outClass::AtTime(int SoilCol, float Time, float Mtotal, float Mgtot, float Mltot,
float Mstot)
{
       Out << endl << endl;
       Out << "Soil Column  " << SoilCol << endl;
    Out << "At time = " << Time << ", total mass in vadose zone = " << Mtotal << "
g/ft^2" << endl;
    Out << "Mass in gas phase = " << Mgtot << "    g/ft^2" << endl;
    Out << "Mass in liquid phase = " << Mltot << "    g/ft^2" << endl;
    Out << "Mass sorbed = " << Mstot << "    g/ft^2" << endl;
}//end AtTime
void outClass::SinceBegin(float Time)
{
       Out << endl << endl;
       Out << "Since beginning of run at time = " << Time << endl;
}//end SinceBegin
void outClass::SinceLast(float Time)
{
       Out << endl << endl;
       Out << "Since last printout at time = " << Time << endl;
}//end SinceLast
void outClass::Change(float Del1, float Mt, float Mb, float Mc, float Del2)
{
       Out << " Change in Total mass = " << Del1 << "    g/ft^2" << endl;
       Out << "  Mass Flux Across Top Boundary = " << Mt << "   g/ft^2" << endl;
       Out << "  Mass Flux Across Bottom Boundary = " << Mb << "    g/ft^2" << endl;
       Out << "  Mass Loss due to Decay in Column = " << Mc << "    g/ft^2" << endl;
       Out << " Net Mass Flux Through Boundaries = " << Del2 << "    g/ft^2" << endl;
       Out << " Mass discrepancy = " << Del1 - Del2 << "    g/ft^2" << endl;
}//end Change
void outClass::GWImpact(int SoilCol, int Ntime, float Ptime, float Gwimp[], float Areac)
{
       Out << endl << endl;
    Out <<
********************************************************************************* << endl;
    Out << "Ground-water IMPACT OF soil column " << SoilCol << endl;
    Out << endl;
    Out << "Time          Mass flux (g/yr/ft^2)          Total Mass (g/yr)" << endl;
       for(int i=0; i < Ntime; ++i)
       {
              Out << setprecision(3) << setw(9) << (i+1) * Ptime
                     << setprecision(8) << setw(22) << Gwimp[i] / Areac
                     << setprecision(5) << setw(31) << Gwimp[i] << endl;
       }//end for t
    Out <<
********************************************************************************* << endl;
       //Spaces
       Out << endl << endl;
}//end GWImpact
void outClass::TotalGWTitle()
{
       Out <<
********************************************************************************* << endl;
       Out << "TOTAL GROUND WATER IMPACT" << endl;
       Out <<
********************************************************************************* << endl;
       Out << endl;
       Out << "Time (yr)          Mass (g/yr)          Cumulative Mass (g)" << endl;
}//end TotalGWTitle
void outClass::TotalGWImpact(int Ntime, float Ptime, float Gwimtl, float Gcum)
{
       Out << setprecision(3) << setw(8)  << Ptime
              << setprecision(3) << setw(18) << Gwimtl
              << setprecision(3) << setw(24) << Gcum << endl;
}//end TotalGWImpact

// ************************************************************
//
// PrfOutput.cpp
//
// ************************************************************
```

```
#include "PrfOutput.h"
prfClass::prfClass()
{
}//end default constructor
prfClass::prfClass(char *FileName)
{
	OpenFile(FileName);
}//end constructor
void prfClass::OpenFile(char *FileName)
{
	Out.open(FileName);
}//end OpenFile
void prfClass::CloseFile()
{
	Out.close();
}//end CloseFile
void prfClass::Title(char *Title, float Time)
{
	Out << endl << endl;
	Out << Title << endl;
	Out << "Time : " << Time << endl;
	Out << "Dist(ft)	Cgas(mg/L)	Cliq(mg/L)	Cs(mg/Kg)" << endl;
}//end Title
void prfClass::Output(float Depthz, float Cgas, float Cliq, float Cs)
{
	Out << setprecision(3) << setw(9)  << Depthz
		<< setprecision(8) << setw(17) << Cgas
		<< setprecision(8) << setw(18) << Cliq
		<< setprecision(8) << setw(17) << Cs << endl;
}//end Output
void prfClass::Spaces()
{
	Out << endl << endl;
}//end Spaces

// ****************************************************************
//
// PrmOutput.cpp
//
// ****************************************************************

#include "PrmOutput.h"
prmClass::prmClass()
{
}//end default condtructor
prmClass::prmClass(char *FileName)
{
	OpenFile(FileName);
}//end condtructor
void prmClass::OpenFile(char *FileName)
{
	Out.open(FileName);
}//end OutputFile
void prmClass::CloseFile()
{
	Out.close();
}//end Close
void prmClass::Output(vadoseInput &Vad)
{
int i,j,J1;
//Space between each Out statement represent different lines of Output
Out << Vad.SimTitle << endl;
Out << "There are " << Vad.Nscol << " soil Columns." << endl;
Out<<"Timestep="<<Vad.Delt<<"years. Simulation length="<<Vad.Stime<<"years."<<endl;
Out<<"Printout every"<<Vad.Ptime<<"years.Vertical Profile stored every"<<
	Vad.Prtime<<" years." << endl;
Out << "Koc = " << Vad.Koci << " ml/g     ";
Out << Vad.Koc << " ft/g" << endl;
Out << "Kh = " << Vad.Kh << " (dimensionless)" << endl;
Out<<"Aqueous solulbility = "<<Vad.Cmaxi<<" ml/g,     "<<Vad.Cmax<<" g/ft^3"<<endl;
```

```
Out<<"Free air diffusion coeficient ="<<Vad.Dairi<<"cm^2/sec"<<Vad.Dair<<" ft^2/g"<<endl;
for(i=0; i < Vad.Nscol; ++i)
{
        //Spaces
        Out << endl << endl;
        Out << "SOIL COLUMN : " << (i+1) << endl << endl;
        Out << Vad.SoilTitle[i] << endl;
        Out << " Column Area = " << Vad.Areac[i] << "  ft^2" << endl;
        Out << " "<<Vad.Ncell[i]<<"cells, each cell"<<Vad.Delz[i]<<"ft. thick."<<endl;
        Out<<"Longitudinal dispersivity (aL)in vadose zone="<<Vad.Aldisp[i]<<"ft."<< endl;
        Out<<"Contaminant Decay Rate in the vadose zone ="<<Vad.Decy1[i]<<" 1/yr" << endl;
        Out << " Decay Rate in the Source = " << Vad.Decy2[i] << "    1/yr" << endl;
        //Spaces
        Out << endl << endl;
        for(j=0; j<Vad.GrNum[i]; ++j)
        {
                J1 = Vad.J1[i][j] - 1;
                Out<<""Soil Properties"(Cell No."<<Vad.J1[i][j]<<"-"<<
                        Vad.J2[i][j]<<")"<<endl;
                Out<<"Bulk density ="<<Vad.Rhob[i][J1]/CONVERT_1<<"g/ml,"<<
                        Vad.Rhob[i][J1] << " g/ft^3" << endl;
                Out<<"Porosity="<<Vad.Por[i][J1]<<"Volumetric water content="<<
                        Vad.Thetaw[i][J1] << endl;
                Out << "Organic carbon content = " << Vad.Foc[i][J1] << endl;
                Out << "Recharge Rate = " << Vad.Q[i][J1] << " ft/yr" << endl;
                if (Vad.Typtbc[i] == 1)
                {
                        Out << "Fixed Conc. at Cell #1 (1st";
                }//end if Tybtc=1
                else
                {
                        Out << "Fixed Flux at Cell #1 (3rd";
                }//end else
                Out<<"type b.c.)="<<Vad.Valtbc[i]<<"mg/l,="<<Vad.Vatbc[i]<<"g/ft^3"<<endl;
                //Spaces
                Out << endl << endl;
        }//end for j
        Out << "***** Aquifer Properties ***** << endl;
        Out << "Horizontal GW Flux (q=ki) = " << Vad.Flxaqf[i] << " ft/yr" << endl;
        Out << "Background Conc. of GW Flux = " << Vad.Cinaqfi[i] << " mg/L" << endl;
        Out << "Leachate Penetration Depth = " << Vad.Pdepth[i] << " ft" << endl;
}//end for i
}//end Output

// ****************************************************************
//
// SimOutput.cpp
//
// ****************************************************************

#include "SimOutput.h"
simClass::simClass()
{
}//end default constrructor
simClass::simClass(char *FileName)
{
        OpenFile(FileName);
}//end constructor
void simClass::OpenFile(char *FileName)
{
        Out.open(FileName);
}//end OpenFile
void simClass::CloseFile()
{
        Out.close();
}//CloseFile
void simClass::Title(int SoilCol, float Time)
{
        Out << "Soil Column " << SoilCol << " at t = " << Time << endl;
        Out << SoilCol << endl;
}//end Title
```

```
void simClass::Output(float Cs, float Delz)
{
        Out << Cs << ", " << Delz << endl;
}//end Output
void simClass::Terminate()
{
        Out << "9999" << endl;
}//Terminate

// ************************************************************
//
// CmxOutput.cpp
//
// ************************************************************

#include "CmxOutput.h"
cmxClass::cmxClass()
{
}//end default constructor
cmxClass::cmxClass(char *FileName)
{
        OpenFile(FileName);
}//end constructor
void cmxClass::OpenFile(char *FileName)
{
        Out.open(FileName);
}//end OpenFile
void cmxClass::ClosePile()
{
        Out.close();
}//end ClosePile
void cmxClass::Title(char Angl)
{
Out << "SATURATED GW CONCENTRATION (underneath columns) DUE TO MIXING" << endl;
        if(Angl=='T')Out<<"SOIL COLUMNS ARE ARRANGED 90 DEGREE TO GROUNDWATER DIRECTION"
<< endl;
        else
                Out<<"SOIL COLUMNS ARE ARRANGED PARALLEL TO GROUNDWATER DIRECTION"<<endl;

Out << "                Liquid Phase Concentrations (mg/L) " << endl;
}//end Title
void cmxClass::ColInfo(char Angl, int SoilCol)
{
Out << endl << endl;
        Out << "Soil Column  " << SoilCol << "      (mg/L)" << endl;
    if (Angl == 'T')
                Out<<"TIME  Cin(vadose) Cin(aquifer)   Cout(mixed)" << endl;
    else
        {
        if (SoilCol == 1)
            Out << "   TIME    Cw,vad(in)    Caq,aqui(in)   Cmx(out)" << endl;
        else
            Out << "   TIME    Cw,vad(in)    Caq,aqui(in)   Cmx(in)   Cmx(out)" << endl;
    }//end else
}//end ColINfo
void cmxClass::RAngleOutput(float Time, float Cliq, float Cinaqf, float Cmixed)
{
Out.setf(ios::dec);
Out << setiosflags(ios::dec) << setprecision(3) << setw(8) << Time;
Out << setiosflags(ios::scientific)                                    << setw(17) << Cliq
        << setiosflags(ios::scientific)                                << setw(20) << Cinaqf
        << setiosflags(ios::scientific)                                << setw(19) << Cmixed
<< endl;
}//end RAngleOutput
void cmxClass::PAngleOutput_1(float Time, float Cliqend, float Cinpaq, float Cmixe)
{
Out.unsetf(Flag);
Out << setiosflags(ios::dec) << setprecision(3) << setw(8) << Time
        << setiosflags(ios::scientific)       << setw(18) << Cliqend
        << setiosflags(ios::scientific)       << setw(20) << Cinpaq
        << setiosflags(ios::scientific)       << setw(16) << Cmixe << endl;
```

```
}//end PAngleOutput_1
void cmxClass::PAngleOutput_2(float Time, float Cliqend, float Cinpaq, float CmixeIn,
float CmixeOut)
{
Out.unsetf(Flag);
Out << setiosflags(ios::dec) << setprecision(3) << setw(8) << Time
       << setiosflags(ios::scientific)       << setw(18) << Cliqend
       << setiosflags(ios::scientific)       << setw(20) << Cinpaq
       << setiosflags(ios::scientific)       << setw(15) << CmixeIn
       << setiosflags(ios::scientific)       << setw(16) << CmixeOut << endl;
}//end PAngleOutput_2

// ************************************************************
//
// GimOutput.cpp
//
// ************************************************************

#include "GimOutput.h"
gimClass::gimClass()
{
}//end default constructor
gimClass::gimClass(char *FileName)
{
       OpenFile(FileName);
}//end constructor
void gimClass::OpenFile(char *FileName)
{
       Out.open(FileName);
}//end OpenFile
void gimClass::CloseFile()
{
       Out.close();
}//end CloseFile
void gimClass::Title(int SoilCol)
{
       Out << "Soil Column " << SoilCol+1 << endl;
}//end Title
void gimClass::NumOutput(float Time, float Num)
{
       Out << Time << ", " << Num << endl;
}//end NumOutput
void gimClass::Terminate()
{
       Out << "9999" << endl;
}//end Terminate
```

APPENDIX B

SIMULATION PARAMETER INFORMATION

1. Properties of Selected Volatile and Semi-Volatile Compounds
2. Diffusion Coefficients of Selected Organic Compounds
3. Environmental Fate Constant Values at 25°C for Additional 27 Constituents
4. Soil Hydraulic Properties
5. Physical Properties of Soil

Table B.1. Properties of Selected Volatile and Semi-Volatile Compounds (Adapted from USEPA, 1990c)

CHEMICAL	GROUP	TYPE	SOLUBILITY CONSTANT (*mg/l*)	HENRY'S CONSTANT (*atm m/mol*)	HENRY'S (*dimensionless*)	DENSITY (*g/cc*)	LOG K_{ow}	LOG K_{oc} (*cc/g*)
Carbon Tetrachloride	Halogenated Volatile Organic	Liquid Solvent	8.00E+02	2.00E-02	8.13E-01	1.595	2.83	2.64
Chlorobenzene	Halogenated Volatile Organic	Liquid Solvent	4.90E+02	3.46E-03	1.41E-01	1.106	2.84	2.20
Chloroform	Halogenated Volatile Organic	Liquid Solvent	8.22E+03	3.75E-03	1.52E-01	1.485	1.97	1.64
Cis-1, 2 –dichloroethylene (d)	Halogenated Volatile Organic	Liquid Solvent	3.50E+03	7.50E-03	3.05E-01	1.284	1.86	1.50
1, 1 – Dichoroethane (a)	Halogenated Volatile Organic	Liquid Solvent	5.50E+03	5.70E-03	2.32E-01	1.175	1.79	1.48
1, 2 - Dichoroethane	Halogenated Volatile Organic	Liquid Solvent	8.69E+03	1.10E-03	4.47E-02	1.253	1.48	1.15
1, 1 – Dichoroethylene	Halogenated Volatile Organic	Liquid Solvent	4.00E+02	1.54E-01	6.26E+00	1.214	2.13	1.81
1, 2 – Dichoropopane (a)	Halogenated Volatile Organic	Liquid Solvent	2.70E+03	3.60E-03	1.46E-01	1.158	2.02	1.71
Ethylene Dibromide (g)	Halogenated Volatile Organic	Liquid Solvent	3.40E+03	3.18E-04	1.29E-02	2.172	1.76	1.45
Methylene chloride	Halogenated Volatile Organic	Liquid Solvent	1.32E+04	2.57E-03	1.04E-01	1.325	1.25	0.94
1, 1, 2, 2 - Tetrachloroethane	Halogenated Volatile Organic	Liquid Solvent	2.90E+03	5.00E-04	2.03E-02	1.600	2.39	2.24
Tetrachloroethylene	Halogenated Volatile Organic	Liquid Solvent	1.50E+02	2.27E-02	9.23E-01	1.625	3.14	2.82
Trans1,2–dichloroethylene(d)	Halogenated Volatile Organic	Liquid Solvent	6.30E+03	6.60E-03	2.68E-01	1.257	2.09	1.77
1, 1, 1 - Trichloroethane	Halogenated Volatile Organic	Liquid Solvent	9.50E+02	2.76E-03	1.12E-01	1.325	2.49	2.18
1, 1, 2 – Trichloroethane	Halogenated Volatile Organic	Liquid Solvent	4.50E+03	1.17E-03	4.76E-02	1.444	2.17	1.75
Trichloroethylene	Halogenated Volatile Organic	Liquid Solvent	1.00E+03	8.92E-03	3.63E-01	1.462	2.42	2.10
Chloroethane	Halogenated Volatile Organic	Gas	5.70E+03	1.10E-02	4.47E-01	0.941	1.43	1.17
Vinyl Chloride	Halogenated Volatile Organic	Gas	1.10E+03	6.95E-01	2.83E+01	0.912	0.60	0.91
Methyl Ethyl Ketone	Non-Halogenated Volatile Organic	Kentone/furan	2.68E=05	2.74E-05	1.11E-03	0.805	0.29	0.65
4 – Methyl – 2 – Pentanone	Non-Halogenated Volatile Organic	Kentone/furan	1.90E+04	1.55E-04	6.30E-03	0.802	1.25	1.38
Tetrahydrofuran	Non-Halogenated Volatile Organic	Kentone/furan	3.00E+05	1.10E-04	4.47E-03	0.889	0.46	No data
Benzen	Non-Halogenated Volatile Organic	Aromatic	1.78E+03	5.43E-03	2.21E-01	0.877	2.13	1.81
Ethyl Benzene	Non-Halogenated Volatile Organic	Aromatic	1.52E+02	7.90E-03	3.21E-01	0.867	3.15	2.83
Styrene	Non-Halogenated Volatile Organic	Aromatic	3.00E+02	2.28E-03	9.27E-02	0.906	3.16	No data
Toluene	Non-Halogenated Volatile Organic	Aromatic	5.15E+02	6.61E-03	2.69E-01	0.867	2.73	2.41
m-Xylene	Non-Halogenated Volatile Organic	Aromatic	2.00E+02	6.91E-03	2.81E-01	0.864	3.20	2.84
o-Xylene	Non-Halogenated Volatile Organic	Aromatic	1.70E+02	4.94E-03	2.01E-01	0.880	3.12	2.84
p-Xylene	Non-Halogenated Volatile Organic	Aromatic	1.98E+02	7.01E-03	2.85E-01	0.861	3.15	2.84
Arochlor 1242	Halogeneated semi-volatile Organic	PCB	4.50E-01	3.40E-04	1.38E-02	1.385	5.58	5.00
Arochlor 1254	Halogeneated semi-volatile Organic	PCB	1.20E-02	2.80E-04	1.14E-02	1.538	6.03	No data
Arochlor 1260	Halogeneated semi-volatile Organic	PCB	2.70E-03	3.40E-04	1.38E-02	1.440	7.15	No data
Chlordane	Halogeneated semi-volatile Organic	Pesticide	5.60E-02	2.20E-04	8.94E-03	1.600	5.48	4.58

Table B.1. Properties of Selected Volatile and Semi-Volatile Compounds (Adapted from USEPA, 1990c) (Continued)

CHEMICAL	GROUP	TYPE	SOLUBILITY CONSTANT (*mg/l*)	HENRY'S CONSTANT (*atm m/mol*)	HENRY'S (*dimensionless*)	DENSITY (*g/cc*)	LOG K_{ow}	LOG K_{oc} (*cc/g*)
DDD	Halogeneated semi-volatile Organic	Pesticide	1.60E-01	7.96E-06	3.24E-04	1.385	5.56	5.38
DDE	Halogeneated semi-volatile Organic	Pesticide	4.00E-02	1.90E-04	7.72E-03	No data	5.69	5.41
DDT	Halogeneated semi-volatile Organic	Pesticide	3.10E-03	2.80E-05	1.14E-03	0.985	6.36	5.48
Dieldrin	Halogeneated semi-volatile Organic	Pesticide	1.86E-01	9.70E-06	3.94E-04	1.750	5.34	3.23
1, 2 – Dichorobenzene	Halogeneated semi-volatile Organic	Chlorinated Benzene	1.00E+02	1.88E-03	7.64E-02	1.306	3.38	3.06
1, 4 – Dichorobenzene	Halogeneated semi-volatile Organic	Chlorinated Benzene	8.00E+01	1.58E-03	6.42E-02	1.248	3.39	3.07
Pentachlorophenol (w)	Halogeneated semi-volatile Organic	Chlorinated Phenol	1.40E+01	2.80E-06	1.14E-04	1.978	5.12	4.80
2,3,4,6-Tetrachlorophenol	Halogeneated semi-volatile Organic	Chlorinated Phenol	1.00E+03	No data	No data	1.839	4.10	2.00
Acenaphthene	Non-Halogeneated semi-volatile Organic	PAH	3.88E+00	1.20E-03	4.88E-02	1.225	3.92	3.70
Anthracene	Non-Halogeneated semi-volatile Organic	PAH	7.50E-02	3.38E-05	1.37E-03	1.250	4.45	4.10
Benzo(a)anthracene	Non-Halogeneated semi-volatile Organic	PAH	1.40E-02	4.50E-06	1.83E-04	1.174	5.61	6.14
Benzo(a)pyrene	Non-Halogeneated semi-volatile Organic	PAH	3.80E-03	1.80E-05	7.32E-04	No data	6.06	6.74
Benzo(b)fluoranthene	Non-Halogeneated semi-volatile Organic	PAH	1.40E-02	1.19E-05	4.84E-04	No data	6.57	5.74
Benzo(ghi)fluoranthene	Non-Halogeneated semi-volatile Organic	PAH	2.60E-04	5.34E-08	2.17E-06	No data	6.51	6.20
Benzo(k)fluoranthene	Non-Halogeneated semi-volatile Organic	PAH	4.30E-03	3.94E-05	1.60E-03	No data	6.06	5.74
Chrysene	Non-Halogeneated semi-volatile Organic	PAH	6.00E-03	1.05E-06	4.27E-05	1.274	5.61	5.30
Dibenz9a,h)anthracene	Non-Halogeneated semi-volatile Organic	PAH	2.50E-03	7.33E-08	2.98E-06	1.252	6.80	6.52
Flucanthene	Non-Halogeneated semi-volatile Organic	PAH	2.65E-01	6.50E-06	2.64E-04	1.252	4.90	4.58
Fluorene	Non-Halogeneated semi-volatile Organic	PAH	1.90E+00	7.65E-05	3.11E-03	1.203	4.18	3.90
Indeno(1,2,3-cd)pyrene	Non-Halogeneated semi-volatile Organic	PAH	5.30E-04	6.95E-08	2.83E-06	No data	6.50	6.20
2 – Methyl naphthalene	Non-Halogeneated semi-volatile Organic	PAH	2.54E+01	5.06E-02	2.06E+00	1.006	3.86	3.93
Napthalene	Non-Halogeneated semi-volatile Organic	PAH	3.10E+01	1.27E-03	5.16E-02	1.162	3.30	3.11
Phenanthrene	Non-Halogeneated semi-volatile Organic	PAH	1.18E+00	3.98E-05	1.62E-03	0.980	4.46	4.10
Pyrene	Non-Halogeneated semi-volatile Organic	PAH	1.48E-01	1.20E-05	4.88E-04	1.271	4.88	4.58
Phenol	Non-Halogeneated semi-volatile Organic	Non-chlorinated phenol	8.40E+04	7.80E-07	3.17E-05	1.058	1.46	1.15
2, 4 - Dinitrophenol	Non-Halogeneated semi-volatile Organic	Non-chlorinated phenol	6.00E+03	6.50E-10	2.64E-08	1.680	1.54	1.22
m-Cresol (e)	Non-Halogeneated semi-volatile Organic	Non-chlorinated phenol	2.35E+04	3.80E-05	1.54E-03	1.038	1.96	1.43
o-Cresol	Non-Halogeneated semi-volatile Organic	Non-chlorinated phenol	3.10E+04	4.70E-05	1.91E-03	1.027	1.95	1.23
p-Cresol	Non-Halogeneated semi-volatile Organic	Non-chlorinated phenol	2.40E+04	3.50E-04	1.42E-02	1.035	1.94	1.28

USEPA, 1990c, Subsurface Remediation Guidance Table 3, EPA/540/2-90/011b.

Table B.2. Diffusion Coefficients of Selected Organic Compounds
(Adapted from USEPA, 1988)

COMPOUND	FORMULA	MOLECULAR WEIGHT	ATOMIC DIFFUSION VOLUME	DIFFUSION COEFFICIENTS (cm^2/sec)		
				at 10° C	at 30° C	at 50° C
Acetaldehyde	C_2H_4O	44	46.40	0.11758	0.13249	0.14816
Acetic Acid	$C_2H_4O_2$	60	51.88	0.10655	0.12007	0.13427
Acetone	C_3H_6O	58	66.86	0.09699	0.10930	0.12223
Aniline	C_6H_7N	93	118.55	0.07157	0.08065	0.09019
Benzene	C_6H_6	78	90.68	0.08195	0.09234	0.10327
Bromoethane	CH_3Br	95	57.44	0.09611	0.10830	0.12111
Bromoform	$CHBr_3$	118	53.48	0.09655	0.10880	0.12167
Carbon Tetrachloride	CCl_4	154	94.50	0.07500	0.08451	0.09451
Chlorobenzene	C_6H_5Cl	113	128.40	0.06769	0.07627	0.08530
Chloroethane	C_2H_5Cl	65	62.40	0.09789	0.11031	0.12336
Chloroform	$CHCl_3$	120	76.89	0.08345	0.09404	0.10517
Chloromethane	CH_3Cl	51	57.94	0.10496	0.11827	0.13226
Cyclohexane	C_6H_{12}	84	122.76	0.07139	0.08045	0.08996
Dichloroethane	$C_2H_4Cl_2$	99	75.96	0.08557	0.09643	0.10784
Dichloroethylene	$C_2H_2Cl_2$	97	106.96	0.07442	0.08386	0.09377
Dichloropropalene	$C_3H_6Cl_2$	113	100.38	0.07519	0.08473	0.09475
Dimethylamine	C_2H_7N	45	52.55	0.11161	0.12577	0.14065
Ethanol	C_2H_6O	46	50.36	0.11297	0.12730	0.14236
Ethyl Acetate	$C_4H_8O_2$	88	92.80	0.07991	0.09005	0.10070
Ethylamine	C_2H_7N	45	5255	0.11161	0.12577	0.14065
Ethylbenzene	C_8H_{10}	116	151.80	0.06274	0.07070	0.07906
Fluorotoluene	C_7H_7F	110	154.36	0.06262	0.07056	0.07891
Heptane	C_7H_{16}	100	146.86	0.06467	0.07287	0.08149
Hexane	C_6H_{14}	86	126.72	0.07021	0.07912	0.08848
Isopropanol	$C_3H_8O_2$	60	37.82	0.12004	0.13526	0.15126
Methanol	CH_4O	32	29.90	0.14808	0.16686	0.18660
Methyl Acetate	$C_3H_6O_2$	74	72.34	0.09054	0.10203	0.11410
Methyl Chloride	CH_2Cl_2	85	59.46	0.09610	0.10830	0.12111
Methylethyl Ketone	C_4H_8O	72	87.32	0.08417	0.09485	0.10607
PCB (1 Cl)	$C_{12}H_9Cl$	189	235.32	0.04944	0.05571	0.06230
Pentane	C_5H_{12}	72	106.26	0.07753	0.08737	0.09770
Phenol	C_6H_6O	84	96.16	0.07919	0.08924	0.09980
Styrene	C_8H_8	104	137.84	0.06620	0.07460	0.08343
Tetrachloroethane	$C_2H_2Cl_4$	168	1143.96	0.06858	0.07729	0.08643
Tetrachloroethylene	C_2Cl_4	166	111.00	0.06968	0.07852	0.08781
Toluene	C_7H_8	92	111.14	0.07367	0.08301	0.09283
Tricyhloroethane	$C_2H_3Cl_3$	133	97.44	0.07496	0.08447	0.09446
Trichlororthylene	C_2HCl_3	131	93.48	0.07638	0.08606	0.09625
Trichlorofluoromethane	CCl_3F	138	100.00	0.07391	0.08329	0.09314
Vinyl Chloride	C_2H_3Cl	63	58.44	0.10094	0.11375	0.12720
Xylene	C_8H_{10}	106	131.60	0.06742	0.07597	0.08495

USEPA, 1988, Superfund Exposure Assessment Manual, office of Remedial Response, Wash. D.C., EPA/540/1-88/001.

Table B.3. Environmental Fate Constant Values at 25°C for Additional 27 Consituents (from USEPA, 1995)

No.	Common Name	Chemical Abstract: Service No.	Water Solubility (*mg/L*)	Sorption Log K_{oc}	Sorption Log K_{ow}	Henry's Law Constant (*atm-m³/mol*)	Vapor Pressure (*torr*)	Diffusivity in Air (cm^2/s)
1.	Anthracene	120-12-7	7.6E-2	4.21	4.53	1.9E-5	6.1E-6	0.055
2.	Benzenethiol[a]	108-98-5	7.6E2	1.32	2.35	4.4E4	2.4	0.076
3.	Benzo[g,h,I]perylene	191-24-2	1.2E4	6.28	6.60	1.2E-7	4.0E-11	0.039
4.	Benzo[k]fluoranthene	207-08-9	9.4E4	6.0	6.3	5.0E-7	1.4E-9	0.041
5.	Bromobenzene	108-86-1	4.1E2	2.43	2.75	2.1E-3	4.2	0.073
6.	n-Butylbenzene	104-51-8	21	3.8	4.1	9.7E-3	1.1	0.060
7.	Sec-Butylbenzene	135-98-8	38	3.6	3.9	9.8E-3	2.1	0.061
8.	Carbazole	86-74-8	4.0E-1	3.3	3.6	8.6E-7	1.6E-6	0.062
9.	Crotonaldehyde	4170-30-3	1.3E5	-0.06	0.26	3.1E-5	4.5E1	0.093
10.	Dibenzofuran	132-64-9	4.3	3.8	4.1	1.4E-4	2.7E-3	0.059
11.	1,2,3,4,6,7,8-Heptachloro-dibenzofuran	67562-39-4	9.5E-7	8.20	8.52	3.7E-5	6.5E-11	0.043
12.	1,2,3,4,7,8,9- Heptachloro-dibenzofuran	55673-89-7	1.3E-6	8.2	8.5	3.8E-5	9.5E-11	0.043
13.	1,2,3,4,6,7,8- Heptachloro-dibenzo-p-dioxin	35822-46-9	1.9E-7	8.53	8.86	4.1E-5	1.4E-11	0.045
14.	1,2,3,4,7,8- Hexachloro-dibenzofuran	70648-26-9	7.3E-6	7.54	7.86	4.2E-5	6.3E-10	0.045
15.	1,2,3,6,7,8- Hexachloro-dibenzofuran	57117-44-9	6.9E-6	7.55	7.87	4.2E-5	5.9E-10	0.045
16.	1,2,3,7,8,9- Hexachloro-dibenzofuran	72918-21-9	7.3E-6	7.55	7.87	4.3E-5	6.4E-10	0.045
17.	2,3,4,6,7,8- Hexachloro-dibenzofuran	60851-34-5	7.6E-6	7.54	7.86	4.1E-5	6.3E-10	0.045
18.	2-Hexanone	591-78-6	1.8E4	1.0	1.3	8.7E-58.7E-5	1.2E1	0.072
19.	Indene	95-13-6	3.9E2	2.5	2.8	5.0E4	1.3	0.071
20.	p-Isopropyltoluene	99-87-6	28	3.7	4.0	9.3E-3	1.5	0.060
21.	2-Methylchrysene	3351-32-4	8.5E4	5.82	6.14	1.2E-6	3.1E-9	0.044
22.	1-Methylnaphthalene	90-12-0	40	3.52	3.84	2.8E4	6.6E-2	0.060
23.	3-Methylnaphthalene	91-57-6	33	3.54	3.86	3.0E4	5.8E-2	0.061
24.	Phenanthrene	85-01-8	1.1	4.25	4.57	2.3E-5	1.0E4	0.055
25.	n-Propylbenzene	103-65-1	57	3.35	3.67	9.9E-3	3.6	0.065
26.	1,2,4-Trimethylbenzene	95-63-6	70	3.28	3.60	4.9E-3	2.2	0.065
27.	1.3.5-Trimethylbenzene	108-67-8	67	3.37	3.69	6.3E-3	2.7	0.065

a. Values reported are for neutral species.
b. Bromobenzene was tested in the laboratory for hydrolysis. No disappearance was noted after 29 days at 85°C in 0.1N sodium hydroxide and 0.1N hydrochloric acid.

USEPA, 1995, Environmental Fate Constants for Additional 27 Organic Chemicals under Consideration for EPA's Hazardous Waste Indentification Projects, Project Summary, EPA/600/SR-95/039.

Table B.4. Soil Hydraulic Properties

Soil Texture	Saturated Hydraulic conductivity (cm/hr)		Irreducible water Content
	(Li et al., 1976)	(McCuen et al., 1981)	(Carsel and Parish, 1988)
Sand	63.36	24.60	0.045
Loamy sand	56.28	78.84	0.057
Sandy Loam	12.48	17.93	0.065
Silt Loam	2.59	1.62	0.078
Loam	2.50	5.98	0.067
Sandy Clay Loam	2.27	4.72	0.100
Silty Clay Loam	0.61	1.07	0.089
Clay Loam	0.88	3.64	0.095
Sandy Clay	0.78	1.25	-
Silty Clay	0.37	1.80	0.070
Clay	0.46	1.07	0.068

Carsel, R.F. and Parrish, R.S., 1988, Developing joint probability distributions of soil water retention characteristics, Water Resources Research, 24(5): pp. 755-769.

Li, E.A., Shanholtz, and Carson, E.W., 1976, Estimating Saturated Hydraulic Conductivity and Capillary Potential at the Wetting Front, Dept. of Agricultural Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA.

McCuen, R.H., Rawls, W.J., and Brakensiek, D.L., 1981, Statistical Analysis of the Brooks-Corey and the Green-Ampt Parameters, Water Resources Research, 17(4): pp. 1005-1013.

Table B.5. Physical Properties of Soil

Soil Texture	Bulk Density (g/cc)	Porosity		Organic Matter Content (gr/gr)
	(Jury, 1986)	(Li et al., 1976)	(Brakensiek et al., 1981)	(Rawls, 1983)
		Mean (std. Deviation)	Mean (std. Deviation)	Mean (std. Deviation)
Sand	1.59 - 1.65	0.359 (0.056)	0.349 (0.107)	0.0071 (0.0106)
Loamy sand	-	0.410 (0.068)	0.410 (0.065)	0.0061 (0.0116)
Sandy Loam	1.20 - 1.47	0.435 (0.086)	0.423 (0.076)	0.0071 (0.0129)
Silt Loam	1.47	0.485 (0.059)	0.484 (0.057)	0.0058 (0.0129)
Loam	-	0.451 (0.078)	0.452 (0.069)	0.0052 (0.0099)
Sandy Clay Loam	-	0.420 (0.059)	0.406 (0.049)	0.0019 (0.0042)
Silty Clay Loam	-	0.477 (0.057)	0.473 (0.046)	0.0013 (0.0042)
Clay Loam	1.20 - 1.36	0.476 (0.053)	0.476 (0.054)	0.0010 (0.0042)
Sandy Clay	-	0.426 (0.057)	-	0.0038 (0.0120)
Silty Clay	1.26	0.492 (0.064)	-	-
Clay	-	0.482 (0.050)	0.475 (0.048)	0.0038 (0.0083)

- oc = om/1.724; where oc = organic carbon content, om = organic matter content (after Dragun, 1988)

Dragun, J., 1988, The Soil Chemistry of Hazardous Materials, Hazardous Materials Control Research Institute, Silver Spring, MD.

Jury, W.A., 1986, in Vadose Zone Modeling of Organic Pollutants, Chapter 11, eds., Hern, S.C. and Melancon, S.M., Lewis Publishers, Chelsea, MI.

Li, E.A., Shanholtz, and Carson, E.W., 1976, Estimating Saturated Hydraulic Conductivity and Capillary Potential at the Wetting Front, Dept. of Agricultural Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA.

Brakensiek, D.L., Engleman, R.L., and Rawls, 1981, Variation within Texture Classes of Soil Water Parameters, Trans. ASAE, Vol. 24, pp. 335-339.

Rawls, W.J., 1983, Estimating Bulk Density from Particle Size Analysis and Organic Matter Content, Soil Science, vol. 135, no. 2, pp. 123-125.

APPENDIX C

FINITE DIFFERENCE FORM OF THE THIRD TYPE BOUNDARY CONDITION

By Samuel Sangwon Lee

1998

Finite Difference Form of the Third Type Boundary Condition

As shown in equations (3.15a) and (3.15b) in the section 3.1.1, the third type boundary condition at the top of the soil column is expressed as:

$$-D\frac{\partial C_w}{\partial z}+q_w C_w\Big|_{z=0}=q_w C_0\exp(-\gamma t)\quad (t\le t_0) \tag{C.1a}$$

$$-D\frac{\partial C_w}{\partial z}+q_w C_w\Big|_{z=0}=0\qquad (t>t_0) \tag{C.1b}$$

If we assume D is constant over the top pixels involved here, the finite difference form of the first term of the left hand of the above equations can be written (omitting the subscript *w*) as:

$$D\frac{\partial C}{\partial z}=\frac{D}{2}\left[\frac{1}{2}\left\{\frac{C_{i+1}^{k+1}-C_i^{k+1}}{\Delta z}+\frac{C_{i+1}^{k}-C_i^{k}}{\Delta z}\right\}+\frac{1}{2}\left\{\frac{C_i^{k+1}-C_{i-1}^{k+1}}{\Delta z}+\frac{C_i^{k}-C_{i-1}^{k}}{\Delta z}\right\}\right]$$

$$=\frac{D}{4(\Delta z)}\left(C_{i+1}^{k+1}-C_{i-1}^{k+1}+C_{i+1}^{k}-C_{i-1}^{k}\right) \tag{C.2}$$

If we assume q_w is constant over top two pixels involved here, the second term of the equations (C.1a) and (C.1b) becomes

$$q_w C=\frac{q_w}{2}\left(C_i^{k+1}+C_i^{k}\right) \tag{C.3}$$

By substituting the equations (C.2) and (C.3) into (C.1a) and (C.1b),

$$\frac{-D}{4(\Delta z)}\left(C_{i+1}^{k+1}-C_{i-1}^{k+1}+C_{i+1}^{k}-C_{i-1}^{k}\right)+\frac{q_w}{2}\left(C_i^{k+1}+C_i^{k}\right)=q_w C_0\exp(-\gamma t) \qquad \text{(C.4)}$$

where C_0 becomes zero when $t > t_0$.

When $i = 1$ (*i.e.*, at the first cell), the equation (C.4) can be written as:

$$\frac{-D}{4(\Delta z)}\left(C_2^{k+1}-C_0^{k+1}+C_2^{k}-C_0^{k}\right)+\frac{q_w}{2}\left(C_1^{k+1}+C_1^{k}\right)=q_w C_0\exp(-\gamma t) \qquad \text{(C.5)}$$

Note that the C_0^{k+1} and C_0^k terms represent outside the column boundary. The same terms also can be found in the governing equation (4.9) of section 4.1 by applying $i = 1$.

When $i = 1$, the equation (4.9) becomes as follows.

$$\begin{aligned}&\left(-M_1+M_1^{'}-N_1\right)C_0^{k+1}+\left(1+2M_1+N_1^{'}+L_1\right)C_1^{k+1}+\left(-M_1-M_1^{'}+N_1\right)C_2^{k+1}\\&=\left(M_1-M_1^{'}+N_1\right)C_0^{k}+\left(1-2M_1-N_1^{'}-L_1\right)C_1^{k}+\left(M_1+M_1^{'}-N_1\right)C_2^{k}\end{aligned} \qquad \text{(C.6)}$$

Rewriting equation (C.6) and rearranging the C^{k+1} terms into the right hand side and C^k terms into the left side of equation, the following can be derived.

$$C_1^{k+1}-\frac{\Psi^{'}}{\Phi^{'}}C_2^{k+1}=-\frac{\Omega^{'}}{\Phi^{'}}C_1^{k}+\frac{\Psi^{'}}{\Phi^{'}}C_2^{k}+\frac{q_w C_0}{\Phi^{'}}\exp(-\gamma t) \qquad \text{(C.7)}$$

$$\Phi' = \frac{D\left(2M_1 + N_1' + L_1 + 1\right)}{4(\Delta z)(M_1 - M_1' + N_1)} + \frac{q_w}{2},$$

$$\Psi' = \frac{DM_1}{2(\Delta z)(M_1 - M_1' + N_1)}, \text{ and}$$

$$\Omega' = \frac{D\left(2M_1 + N_1' + L_1 - 1\right)}{4(\Delta z)(M_1 - M_1' + N_1)} + \frac{q_w}{2},$$

Because we assumed D and q_w are constant, M_1' and N_1' become 0. Then

$$\Phi' = \frac{D(2M + L + 1)}{4(\Delta z)(M + N)} + \frac{q_w}{2},$$

$$\Psi' = \frac{DM}{2(\Delta z)(M + N)},$$

$$\Omega' = \frac{D(2M + L - 1)}{4(\Delta z)(M + N)} + \frac{q_w}{2},$$

$$M = \frac{D(\Delta t)}{2\theta_w(\Delta z)^2},$$

$$N = \frac{q_w(\Delta t)}{4\theta_w(\Delta z)}, \text{ and}$$

$$L = \mu\left(\frac{\Delta t}{2}\right).$$

APPENDIX D

SIMPLIFIED ESTIMATION FOR DEPTH OF PENETRATION

Excerpted from

USEPA, 1990 USEPA, 1990, Background Document for EPA's Composite Model for Landfills (EPACML), Prepared by Woodward-Clyde Consultants for USEPA, Office of Solid Waste, Washington D.C.

Simplified Estimation for Depth of Penetration

The depth of penetration of a solute plume that is developing under a surface impoundment can be estimated by separating the contribution of advection and dispersion during solute transport

$$H = h_{adv} + h_{disp} \tag{D.1}$$

where H [L] is the depth of penetration, h_{adv} [L] is the vertically advected component of the penetration depth and h_{disp} [L] is the vertically dispersed component of the penetration depth.

The advected depth h_{adv} is the depth that a particle would be transported under the influence of vertical advection

$$h_{adv} = \int_{t=0}^{\tau} V_z dt \tag{D.2}$$

where V_z [L/T] is the vertical seepage velocity and τ [T] is time of travel. If the vertical seepage velocity is a constant with depth, then

$$h_{adv} = V_z \tau \tag{D.3}$$

However, under impoundments, the vertical seepage velocity varies linearly with depth, with a maximum value at the top of the water table and zero at the bottom of the aquifer. A numerical solution for a surface impoundment was performed using SEFTRAN, with the vertical velocity variation under the impoundment plotted in Figure D.1. This variation can be modeled mathematically as:

$$V_z = V_{zo}\left(1-\frac{z}{B}\right) \tag{D.4}$$

where B [L] is the saturated aquifer thickness, z [L] is the depth from the top of the water table and V_{zo} [L/T] is the maximum vertical seepage velocity. V_{zo} can be estimated from the net vertical recharge rate.

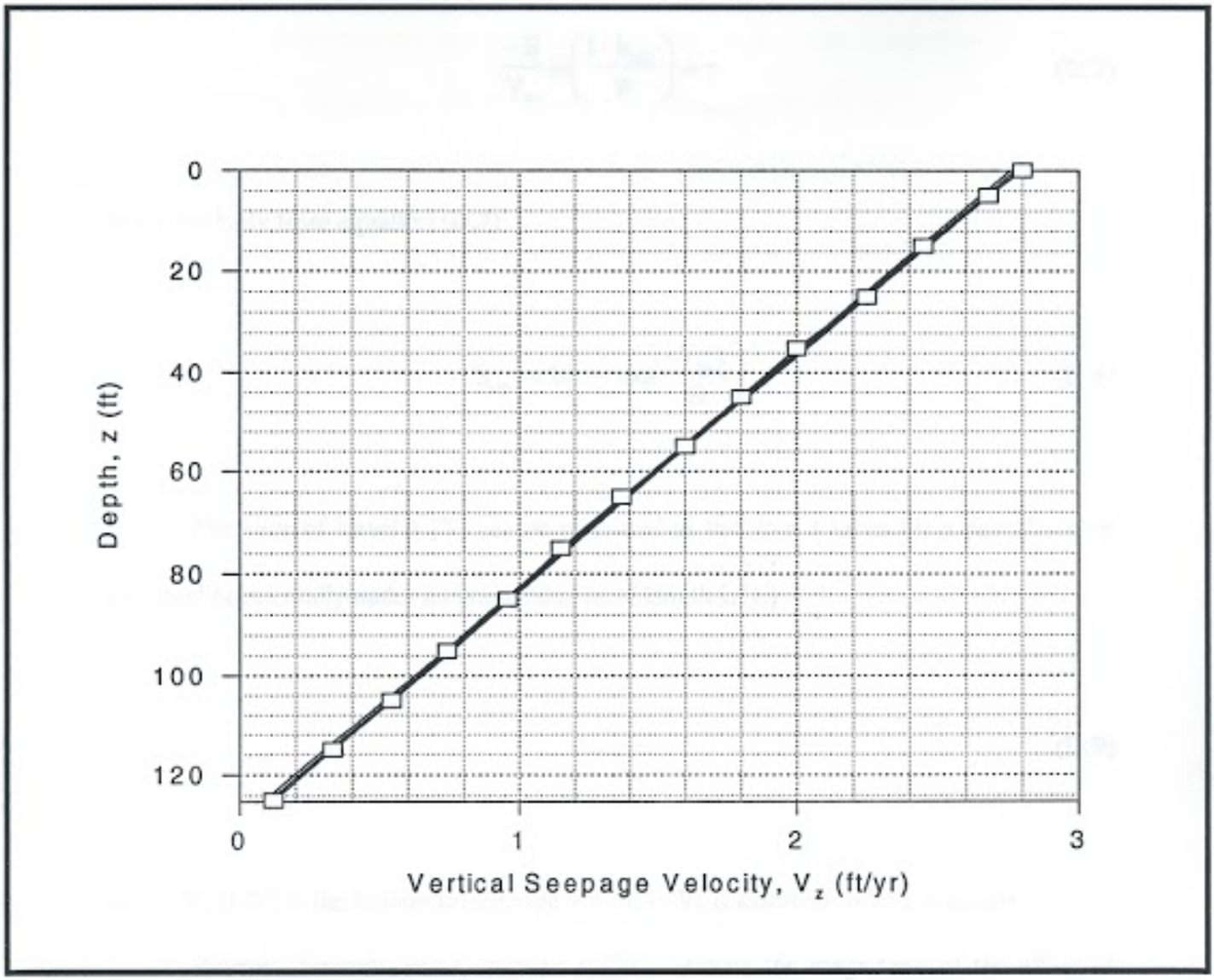
As written, equation (D.2) cannot be integrated since V_z is not an explicit function of time. Consider the following differential equation for the vertical seepage velocity

$$\frac{dz}{dt} = V_z(z) \tag{D.5}$$

Rearrange terms in equation (D.5) and integrate to depth hadv

$$\int_{z=0}^{h_{adv}} \frac{dz}{V_z(z)} = \int_{t=0}^{\tau} dt \tag{D.6}$$

Substitute equation (D.4) into equation (D.6) and integrate to get



SEFTRAN DATA

I = 10 in/yr	Net recharge rate
p = 0.3	porosity
H_o = 132.5 ft	upstream water table elevation
H_L = 118.0 ft	downstream water table elevation
L = 8000 ft	distance between boundaries
K_x = 36500 ft/yr	horizontal hydraulic conductivity
K_z = 3650 ft/yr	vertical hydraulic conductivity
Δx = 200 ft	horizontal element size
Δz = 10 ft	vertical element size
t	steady state

Figure D.1 Variation in the vertical seepage velocity with depth

$$\frac{-B}{V_{zo}} \ln\left(\frac{1-h_{adv}}{B}\right) = \tau \qquad (D.7)$$

Solve for hadv from equation (D.7)

$$h_{adv} = B\left(1 - \exp\frac{-V_{zo}\tau}{B}\right) \qquad (D.8)$$

The time of travel τ [T] can be estimated as the time it takes for a particle to be advected horizontally under an impoundment of length L [L]

$$\tau = \frac{L}{V_x} \qquad (D.9)$$

where V_x [L/T] is the horizontal seepage velocity. V_x is assumed to be a constant.

Prickett, Naymik, and Lonnquist (1981) estimate the magnitude of the effect of the effect of dispersion on particle transport as:

$$\Delta_{long} = \sqrt{2\alpha_L V\tau} \qquad (D.10)$$

$$\Delta_{vert} = \sqrt{2\alpha_V V\tau} \qquad (D.11)$$

where α_L and α_V [L] are the longitudinal and vertical dispersinities; V [L/T] is the magnitude of the seepage velocity; and Δ_{long} and Δ_{vert} [L] are the longitudinal and vertical dispersed distances that correspond to one standard deviation of random transport. If the

effect of the horizontal seepage velocity is assumed to be much larger than that of the vertical, then the dispersed depth is estimated from equation (D.11) as:

$$h_{disp} = \sqrt{2\alpha_v V_x \tau} \qquad (D.12)$$

Hence, the total depth of penetration is the sum of the vertically advected and dispersed components. Substitute equations (D.8) and (D.12) into equation (D.1) to obtain the total estimated depth of penetration

$$H = B\left\{1 - \exp\left(\frac{-V_{zo}\tau}{B}\right)\right\} + \sqrt{2\alpha_v V_x \tau} \qquad (D.13)$$

The solution to equation (D.13) needs to be checked when evaluating any particular case so that a value of H greater than the aquifer thickness B is not used. If the computed H is greater than B, set H equal to B.

References

Prickett, T., T. Naymik and C. Lonnquist. 1981. A random-walk solute transport model for selected groundwater quality evaluations. Bulletin 65 Illinois State Water Survey, Department of Energy and Natural Resources, Champaign, Illinois. 103 pages.

APPENDIX E

Thomas Algorithm for VLEACHSM 2.0 (Multi-Layer Version)

By Samuel Sangwon Lee

1998

Thomas Algorithm for Multi-Layers Version of VLEACHSM 2.0

By Samuel Sangwon Lee

Thomas algorithm solves a tridiagonal matrix equation:

$$\begin{bmatrix} b_1 & c_1 & 0 & 0 & \cdot & 0 \\ a_2 & b_2 & c_2 & 0 & \cdot & 0 \\ 0 & a_3 & b_3 & c_3 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & \cdot & 0 & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \cdot \\ f_{n-1} \\ f_n \end{bmatrix} \tag{E.1}$$

into a form:

$$\begin{bmatrix} 1 & \beta_1 & \cdot & \cdot & \cdot & 0 \\ \cdot & 1 & \beta_2 & \cdot & \cdot & 0 \\ 0 & \cdot & 1 & \beta_3 & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & 1 & \beta_{n-1} \\ 0 & 0 & \cdot & \cdot & \cdot & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \cdot \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \cdot \\ y_{n-1} \\ y_n \end{bmatrix} \tag{E.2}$$

where a_i's, b_i's, c_i's, and f_i's are give constants. β_i's and y_i's can be calculated by comparing corresponding elements of the above two equations, and x_i's can be expressed:

$$\begin{aligned} \alpha_i &= b_i - a_i \beta_{i-1} \\ \beta_i &= \frac{c_i}{\alpha_i} \\ y_i &= \frac{(f_i - a_i y_{i-1})}{\alpha_i} \\ x_i &= y_i - \beta_i x_{i+1} \end{aligned} \tag{E.3}$$

This equation can be solved by assuming:

$$\beta_0 = y_0 = 0 \tag{E.4}$$

and obtaining α_i's, β_i's, and y_i's incrementally:

$$\begin{cases} \alpha_1 = b_1 \\ \beta_1 = \dfrac{c_1}{\alpha_1} \\ y_1 = \dfrac{f_1}{\alpha_1} \end{cases} \tag{E.5}$$

$$\downarrow$$

$$\begin{cases} \alpha_2 = b_2 - \alpha_2\beta_1 \\ \beta_2 = \dfrac{c_2}{\alpha_2} \\ y_2 = \dfrac{(f_2 - a_2 y_1)}{\alpha_2} \end{cases}$$

$$\downarrow$$

$$\vdots$$

$$\downarrow$$

$$\begin{cases} \alpha_{n-1} = b_{n-1} - \alpha_{n-1}\beta_{n-2} \\ \beta_{n-1} = \dfrac{c_{n-1}}{\alpha_{n-1}} \\ y_{n-1} = \dfrac{(f_{n-1} - a_{n-1} y_{n-2})}{\alpha_{n-1}} \end{cases}$$

$$\downarrow$$

$$\begin{cases} \alpha_n = b_n - \alpha_n \beta_{n-1} \\ y_n = \dfrac{(f_n - a_n y_{n-1})}{\alpha_n} \end{cases}$$

$$\downarrow$$

Then x_i's can be obtained by a backsubstitution:

$$\begin{cases} x_n = y_n \\ x_{n-1} = y_{n-1} - \beta_{n-1} x_n \\ \vdots \\ x_2 = y_2 - \beta_2 x_3 \\ x_1 = y_1 - \beta_1 x_2 \end{cases} \qquad \text{(E.6)}$$

APPENDIX F

LIST OF SYMBOLS

APPENDIX F

LIST OF SYMBOLS

LIST OF SYMBOLS

A_{aq}	cross-sectional aquifer area perpendicular to the groundwater flow direction [L^2]
A_{soil}	cross-sectional area perpendicular to the vertical infiltration in the soil column [L^2]
B	aquifer thickness [ft]
C_0	liquid phase solute concentration in the infiltration water [mg/L]
C_{aq}	concentration of horizontal groundwater influx [mg/L]
C_w	concentration of a contaminant in liquid (water) phase [mg/L]
C_a	concentration of a contaminant in vapor (air) phase [mg/L]
$C_{mx(i)}$	mixed concentration in groundwater [W/V]
C_s	concentration of a contaminant in solid phase [mg/Kg]
$C_s(z,0)$	initial solid-phase concentration [mg/Kg]
D	combined dispersion coefficient
D_a	gaseous phase diffusion coefficient in the pore air [ft^2/yr]
D_{air}	diffusion coefficient of the contaminant in the free air [ft^2/yr]
D_w	dispersion coefficient for the liquid phase contaminant in the pore water [ft^2/yr]
f_{oc}	fraction organic carbon of the soil [g/g]
K_d	distribution coefficient between the solid phase and liquid phase [ml/g]
K_H	dimensional form of Henry's Law constant [atm-m^3/mol]
K_{oc}	organic carbon-water partition coefficient [ml/g]
H	Henry's partition coefficient between the air phase and water phase [dimensionless]

H_d	penetration depth [ft]
$\tilde{H}_{di}$	adjusted penetration depth for each column [L]
i	discretized soil column cell
k	time level
L	horizontal length dimension of the waste [ft]
L_u	vertical distance from the source to the observation point (*i.e.*, the water table) [L]
M	total contaminant mass per unit volume of the soil
n	total porosity of the soil: equals the sum of water filled porosity and air filled porosity
q_a	air flow velocity [ft/yr]
q_{aq}	is the Darcy velocity in the aquifer [ft/yr]
q_w	water flow velocity (recharge rate) [ft/yr]
R	universal gas constant ($R = 8.2 \times 10^{-5}$ atm-m^3/mol·K)
T	absolute temperature ($^\circ K = 273.16 + ^\circ C$)
t	time [yr]
t_0	duration of solute release [yr]
z	vertical coordinate with positive being downward
α_L	longitudinal dispersivity of the vadose zone [feet]
α_v	transverse (vertical) dispersivity of the aquifer [ft]
Δt	time increment
γ	decay rate of the solute source due to either degradation or flushing by the infiltration [1/yr]
μ	first order decay rate of a contaminant : in this model, for simplicity, it is assumed that $\mu_w = \mu_a = \mu_s = \mu$

μ_w first order decay rate of a contaminant in water phase [l/yr]

μ_a first order decay rate of a contaminant in gaseous phase [l/yr]

μ_s first order decay rate of a contaminant in solid phase [l/yr]

v seepage velocity

θ combined retardation factor

θ_w volumetric water content (volume of water / total volume) [ft^3/ft^3]

θ_a air-filled porosity (volume of air / total volume) [ft^3/ft^3]

ρ_b bulk density of the soil [gr/cm^3]