The HYDRUS-2D Software Package for Simulating the Two-Dimensional Movement of Water, Heat, and Multiple Solutes in Variably-Saturated Media

Version 2.0

April 1999

U. S. SALINITY LABORATORY
AGRICULTURAL RESEARCH SERVICE
U. S. DEPARTMENT OF AGRICULTURE
RIVERSIDE, CALIFORNIA

The HYDRUS-2D Software Package for Simulating the Two-Dimensional Movement of Water, Heat, and Multiple Solutes in Variably-Saturated Media

Version 2.0

by

J. Šimůnek, M. Šejna and M. Th. van Genuchten

April 1999

U. S. SALINITY LABORATORY
AGRICULTURAL RESEARCH SERVICE
U. S. DEPARTMENT OF AGRICULTURE
RIVERSIDE, CALIFORNIA

DISCLAIMER

This report documents version 2.0 of HYDRUS-2D, a software package for simulating water, heat and solute movement in two-dimensional variably saturated media. The software has been verified against a large number of test cases. However, no warranty is given that the program is completely error-free. If you do encounter problems with the code, find errors, or have suggestions for improvement, please contact one of the authors at

U. S. Salinity Laboratory
USDA, ARS
450 West Big Springs Road
Riverside, CA 92507

Tel. 909-369-4865 (J. Šimůnek)
Tel. 909-369-4846 (M. Th. van Genuchten)
Fax. 909-342-4964
E-mail jsimunek@ussl.ars.usda.gov
rvang@ussl.ars.usda.gov

ABSTRACT

Šimůnek, J., Šejna, M., and M. Th. van Genuchten. 1999. The HYDRUS-2D Software Package for Simulating Two-Dimensional Movement of Water, Heat, and Multiple Solutes in Variably-Saturated Media, Version 2.0, U.S. Salinity Laboratory, USDA, ARS, Riverside, California.

This report documents version 2.0 of HYDRUS-2D, a software package for simulating water, heat, and solute movement in two-dimensional variably saturated media. The software package consists of the HYDRUS2 computer program, and the interactive graphics-based user interface HYDRUS2D. The HYDRUS-2D program numerically solves the Richards' equation for saturated-unsaturated water flow and the convection-dispersion equation for heat and solute transport. The flow equation incorporates a sink term to account for water uptake by plant roots. The heat transport equation considers transport due to conduction and convection with flowing water. The solute transport equations consider convective-dispersive transport in the liquid phase, as well as diffusion in the gaseous phase. The transport equations also include provisions for nonlinear nonequilibrium reactions between the solid and liquid phases, linear equilibrium reactions between the liquid and gaseous phases, zero-order production, and two first-order degradation reactions: one which is independent of other solutes, and one which provides the coupling between solutes involved in sequential first-order decay reactions. In addition, physical nonequilibrium solute transport can be accounted for by assuming a two-region, dual-porosity type formulation which partitions the liquid phase into mobile and immobile regions. The program may be used to analyze water and solute movement in unsaturated, partially saturated, or fully saturated porous media. HYDRUS-2D can handle flow regions delineated by irregular boundaries. The flow region itself may be composed of nonuniform soils having an arbitrary degree of local anisotropy. Flow and transport can occur in the vertical plane, the horizontal plane, or in a three-dimensional region exhibiting radial symmetry about the vertical axis. The water flow part of the model can deal with prescribed head and flux boundaries, boundaries controlled by atmospheric conditions, free drainage boundary conditions, as well as a simplified representation of nodal drains using results of electric analog experiments. This version 2.0 of HYDRUS-2D also includes a Marquardt-Levenberg type parameter optimization algorithm for inverse estimation of soil hydraulic and/or solute transport and reaction parameters from measured transient or steady-state flow and/or transport data.

The governing flow and transport equations are solved numerically using Galerkin-type linear finite element schemes. Depending upon the size of the problem, the matrix equations

resulting from discretization of the governing equations are solved using either Gaussian elimination for banded matrices, or a conjugate gradient method for symmetric matrices and the ORTHOMIN method for asymmetric matrices.

The program is distributed by means of two different options. Option A pertains to the executable HYDRUS2 code (a modification of the former CHAIN_2D code) for use with a graphics-based user interface, HYDRUS2D, for easy data preparation and output display in the MS Windows environment. This option also includes the HYDRUS2D interface, and a structured mesh generator for relatively simple flow domain geometries. Option B additionaly includes a CAD program for more general domain geometries, and the MESHGEN2D mesh generator for an unstructured finite element mesh specifically designed for variably-saturated subsurface flow transport problems.

This report serves as both a user manual and reference document. Detailed instructions are given for data input preparation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF VARIABLES

a	ion activity in soil solution [-]
a_i	conversion factor from concentration to osmotic head [L^{4}M^{-1}]
a_v	air content [L^{3}L^{-3}]
$\overline{a}$	ion activity on the exchange surfaces [-]
A	amplitude of temperature sine wave [K]
A_e	area of a triangular element [L^{2}]
A_{qh}	parameter in equation (8.1) [LT^{-1}]
$[A]$	coefficient matrix in the global matrix equation for water flow, [LT^{-1}] or [L^{2}T^{-1}]$^{+}$
b	normalized root water uptake distribution, [L^{-2}] or [L^{-3}]$^{+}$
b'	arbitrary root water uptake distribution, [L^{-2}] or [L^{-3}]$^{+}$
b_i, c_i	geometrical shape factors [L]
b_1, b_2, b_3	empirical parameters to calculate thermal conductivity λ_0 [MLT^{-3}K^{-1}](e.g.Wm^{-1}K^{-1})
B_{qh}	parameter in equation (8.1) [L^{-1}]
$\{B\}$	vector in the global matrix equation for water flow, [L^{2}T^{-1}] or [L^{3}T^{-1}]$^{+}$
c	solution concentration [ML^{-3}]
c'	finite element approximation of c [ML^{-3}]
c_i	initial solution concentration [ML^{-3}]
c_n	value of the concentration at node n [ML^{-3}]
c_r	concentration of the sink term [ML^{-3}]
c_0	prescribed concentration boundary condition [ML^{-3}]
C	volumetric heat capacity of porous medium [ML^{-1}T^{-2}K^{-1}] (e.g. Jm^{-3}K^{-1})
C_d	factor used to adjust the hydraulic conductivity of elements in the vicinity of drains [-]
C_g	volumetric heat capacity of gas phase [ML^{-1}T^{-2}K^{-1}] (e.g. Jm^{-3}K^{-1})
C_n	volumetric heat capacity of solid phase [ML^{-1}T^{-2}K^{-1}] (e.g. Jm^{-3}K^{-1})
C_o	volumetric heat capacity of organic matter [ML^{-1}T^{-2}K^{-1}] (e.g. Jm^{-3}K^{-1})
C_T	total solution concentration [ML^{-3}] (mmol$_c$l^{-1})
C_w	volumetric heat capacity of liquid phase [ML^{-1}T^{-2}K^{-1}] (e.g. Jm^{-3}K^{-1})
Cr_i^e	local Courant number [-]

d thickness of stagnant boundary layer [L]

d_e effective drain diameter [L]

D side length of the square in the finite element mesh surrounding a drain (elements have adjusted hydraulic conductivities) [L]

D_g ionic or molecular diffusion coefficient in the gas phase [L^2T^{-1}]

D_{ij} effective dispersion coefficient tensor in the soil matrix [L^2T^{-1}]

D_{ij}^g diffusion coefficient tensor for the gas phase [L^2T^{-1}]

D_{ij}^w dispersion coefficient tensor for the liquid phase [L^2T^{-1}]

D_L longitudinal dispersivity [L]

D_T transverse dispersivity [L]

D_w ionic or molecular diffusion coefficient in free water [L^2T^{-1}]

$\{D\}$ vector in the global matrix equation for water flow, [L^2T^{-1}] or [L^3T^{-1}]$^+$

e_n subelements which contain node n [-]

E maximum (potential) rate of infiltration or evaporation under the prevailing atmospheric conditions [LT^{-1}]

E_a activation energy of a reaction or process [$ML^2T^{-2}M^{-1}$] ($m^2s^{-2}mol^{-1}$)

f fraction of exchange sites assumed to be at equilibrium with the solution concentration [-]

$\{f\}$ vector in the global matrix equation for solute transport, [$MT^{-1}L^{-1}$] or [MT^{-1}]$^+$

$[F]$ coefficient matrix in the global matrix equation for water flow, [L^2] or [L^3]$^+$

g gas concentration [ML^{-3}]

g_{atm} gas concentration above the stagnant boundary layer [ML^{-3}]

$\{g\}$ vector in the global matrix equation for solute transport, [$MT^{-1}L^{-1}$] or [MT^{-1}]$^+$

$[G]$ coefficient matrix in the global matrix equation for solute transport, [L^2T^{-1}] or [L^3T^{-1}]$^+$

h pressure head [L]

h^* scaled pressure head [L]

h' finite element approximation of h [L]

h_A minimum pressure head allowed at the soil surface [L]

h_n nodal values of the pressure head [L]

h_{ref} pressure head at reference temperature T_{ref} [L]

h_s air-entry value in the Brooks and Corey soil water retention function [L]

h_S	maximum pressure head allowed at the soil surface [L]
h_T	pressure head at soil temperature T [L]
h_0	initial condition for the pressure head [L]
h_{50}	pressure head at which root water uptake is reduced by 50 % [L]
h_ϕ	osmotic head [L]
$h_{\phi 50}$	osmotic head at which root water uptake is reduced by 50 % [L]
h_Δ	pressure head at the reversal point in a hysteretic retention function [L]
k	kth chain number [-]
k_g	empirical constant relating the solution and gas concentrations [-]
k_s	empirical constant relating the solution and adsorbed concentrations [L^3M^{-1}]
K	unsaturated hydraulic conductivity [LT^{-1}]
K^*	scaled unsaturated hydraulic conductivity [LT^{-1}]
$\mathbf{K}^A$	dimensionless anisotropy tensor for the unsaturated hydraulic conductivity K [-]
K^d	unsaturated hydraulic conductivity of the main drying branch [LT^{-1}]
K^w	unsaturated hydraulic conductivity of the main wetting branch [LT^{-1}]
K_{drain}	adjusted hydraulic conductivity in the elements surrounding a drain [LT^{-1}]
K_{ex}	dimensionless thermodynamic equilibrium constant [-]
K_H	Henry's Law constant [$MT^2M^{-1}L^{-2}$]
K_{ij}^A	components of the dimensionless anisotropy tensor $\mathbf{K}^A$ [-]
K_k	measured value of the unsaturated hydraulic conductivity at θ_k [LT^{-1}]
K_r	relative hydraulic conductivity [-]
K_{ref}	hydraulic conductivity at reference temperature T_{ref} [LT^{-1}]
K_s	saturated hydraulic conductivity [LT^{-1}]
K_s^d	saturated hydraulic conductivity associated with the main drying branch [LT^{-1}]
K_s^w	saturated hydraulic conductivity associated with the main wetting branch [LT^{-1}]
K_T	hydraulic conductivity at soil temperature T [LT^{-1}]
K_v	Vanselow selectivity coefficient [-]
K_{12}	selectivity coefficient [-]
K_Δ	unsaturated hydraulic conductivity at the reversal point in a hysteretic conductivity function [LT^{-1}]

l	pore-connectivity parameter [-]
L	length of the side of an element [L]
L_i	local coordinate [-]
L_n	length of a boundary segment [L]
L_t	width of soil surface associated with transpiration, [L] or $[L^2]^+$
L_x	width of the root zone [L]
L_z	depth of the root zone [L]
m	parameter in the soil water retention function [-]
M^0	cumulative amount of solute removed from the flow region by zero-order reactions, $[ML^{-1}]$ or $[M]^+$
M^1	cumulative amount of solute removed from the flow region by first-order reactions, $[ML^{-1}]$ or $[M]^+$
M_r	cumulative amount of solute removed from the flow region by root water uptake, $[ML^{-1}]$ or $[M]^+$
M_t	amount of solute in the flow region at time *t*, $[ML^{-1}]$ or $[M]^+$
M_t^e	amount of solute in element *e* at time *t*, $[ML^{-1}]$ or $[M]^+$
M_0	amount of solute in the flow region at the beginning of the simulation, $[ML^{-1}]$ or $[M]^+$
M_0^e	amount of solute in element *e* at the beginning of the simulation, $[ML^{-1}]$ or $[M]^+$
n	exponent in the soil water retention function [-]
n^d	exponent in the soil water retention function; drying branch [-]
n^w	exponent in the soil water retention function; wetting branch [-]
n_i	components of the outward unit vector normal to boundary Γ_N or Γ_G [-]
n_s	number of solutes involved in the chain reaction [-]
N	total number of nodes [-]
N_e	number of subelements e_n, which contain node *n* [-]
O	actual rate of inflow/outflow to/from a subregion, $[L^2T^{-1}]$ or $[L^3T^{-1}]^+$
p	exponent in the water and osmotic stress response function [-]
p_t	period of time necessary to complete one temperature cycle (1 day) [T]
p_1	exponent in the water stress response function [-]
p_2	exponent in the osmotic stress response function [-]
Pe_i^e	local Peclet number [-]

q_i components of the Darcian fluid flux density $[LT^{-1}]$

Q_n^A convective solute flux at node *n*, $[MT^{-1}L^{-1}]$ or $[MT^{-1}]^+$

Q_n^D dispersive solute flux at node *n*, $[MT^{-1}L^{-1}]$ or $[MT^{-1}]^+$

Q_n^T total solute flux at node *n*, $[MT^{-1}L^{-1}]$ or $[MT^{-1}]^+$

$\{Q\}$ vector in the global matrix equation for water flow, $[L^2T^{-1}]$ or $[L^3T^{-1}]^+$

$[Q]$ coefficient matrix in the global matrix equation for solute transport, $[L^2]$ or $[L^3]^+$

R solute retardation factor [-]

R_u universal gas constant $[ML^2T^{-2}K^{-1}M^{-1}]$ (=8.314kg $m^2s^{-2}K^{-1}mol^{-1}$)

s adsorbed solute concentration [-]

s^e adsorbed solute concentration on type-1 sites [-]

s_i initial value of adsorbed solute concentration [-]

s^k adsorbed solute concentration on type-2 sites [-]

S sink term $[T^{-1}]$

S_e degree of saturation [-]

S_{ek} degree of saturation at θ_k [-]

S_p spatial distribution of the potential transpiration rate $[T^{-1}]$

S_T cation exchange capacity $[MM^{-1}]$ ($mmol_ckg^{-1}$)

$[S]$ coefficient matrix in the global matrix equation for solute transport, $[L^2T^{-1}]$ or $[L^3T^{-1}]^+$

t time [T]

t^* local time within the time period t_p [T]

t_p period of time covering one complete cycle of the temperature sine wave [T] T temperature [K]

T_a actual transpiration rate per unit surface length $[LT^{-1}]$

$\bar{T}$ average temperature at soil surface during period t_p [K]

T^A absolute temperature [K]

T_i initial temperature [K]

T_p potential transpiration rate $[LT^{-1}]$

T_r^A reference absolute temperature [K] (293.15K=20°C)

T_0 prescribed temperature boundary condition [K]

v average pore-water velocity $[LT^{-1}]$

V	volume of water in each subregion, [L^2] or [L^3]$^+$
V_{new}	volume of water in each subregion at the new time level, [L^2] or [L^3]$^+$
V_{old}	volume of water in each subregion at the previous time level, [L^2] or [L^3]$^+$
V_t	volume of water in the flow domain at time *t*, [L^2] or [L^3]$^+$
V_t^e	volume of water in element *e* at time *t*, [L^2] or [L^3]$^+$
V_0	volume of water in the flow domain at time zero, [L^2] or [L^3]$^+$
V_0^e	volume of water in element *e* at time zero, [L^2] or [L^3]$^+$
W	total amount of energy in the flow region, [MLT^{-2}] or [ML^{2}T^{-2}]$^+$
x_i	spatial coordinates (*i*=1,2) [L]
Z_0	characteristic impedance of a transmission line analog to drain
Z_0'	characteristic impedance of free space (≈376.7 ohms)
α	coefficient in the soil water retention function [L^{-1}]
α	dimensionless water stress response function [-]
α^d	value of α for a drying branch of the soil water retention function [L^{-1}]
α^w	value of α for a wetting branch of the soil water retention function [L^{-1}]
α^w	weighing factor [-]
α_h	scaling factor for the pressure head [-]
α_h^*	temperature scaling factor for the pressure head [-]
α_K	scaling factor for the hydraulic conductivity [-]
α_K^*	temperature scaling factor for the hydraulic conductivity [-]
α_θ	scaling factor for the water content [-]
β	empirical constant in adsorption isotherm [-]
γ_g	zero-order rate constant for solutes in the gas phase [ML^{-3}T^{-1}]
γ_i	activity coefficient in soil solution [L^{3}M^{-1}] (l mol^{-1})
γ_s	zero-order rate constant for solutes adsorbed onto the solid phase [T^{-1}]
γ_w	zero-order rate constants for solutes in the liquid phase [ML^{-3}T^{-1}]
γ_e	boundary segments connected to node *n*
Γ_D	part of flow domain boundary where Dirichlet type conditions are specified
Γ_G	part of flow domain boundary where gradient type conditions are specified
Γ_N	part of flow domain boundary where Neumann type conditions are specified

Γ_C	part of flow domain boundary where Cauchy type conditions are specified
δ_{ij}	Kronecker delta [-]
Δt	time increment [T]
Δt_{max}	maximum permitted time increment [T]
Δt_{min}	minimum permitted time increment [T]
ε	temporal weighing factor [-]
ε_a^c	absolute error in the solute mass balance, $[\mathrm{ML}^{-1}]$ or $[\mathrm{M}]^+$
ε_a^w	absolute error in the water mass balance, $[\mathrm{L}^2]$ or $[\mathrm{L}^3]^+$
ε_r^c	relative error in the solute mass balance [%]
ε_r^w	relative error in the water mass balance [%]
ε_0	permittivity of free space (used in electric analog representation of drains)
η	empirical constant in adsorption isotherm $[\mathrm{L}^3\mathrm{M}^{-3}]$
θ	volumetric water content $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ^*	scaled volumetric water content $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_a	parameter in the soil water retention function $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_k	volumetric water content corresponding to K_k $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_m	parameter in the soil water retention function $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_m^d	parameter in soil water retention function; drying branch $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_m^w	parameter in soil water retention function; wetting branch $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_n	volumetric solid phase fraction $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_o	volumetric organic matter fraction $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_r	residual soil water content $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_r^*	scaled residual soil water content $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_r^d	residual soil water content of the main drying branch $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_r^w	residual soil water content of the main wetting branch $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_s	saturated soil water content $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_s^d	saturated soil water content of the main drying branch $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_s^w	saturated soil water content of the main wetting branch $[\mathrm{L}^3\mathrm{L}^{-3}]$
θ_Δ	water content at the reversal point of a hysteretic retention function $[\mathrm{L}^3\mathrm{L}^{-3}]$
κ	parameter which depends on the type of flow being analyzed, [-] or $[\mathrm{L}]^+$

λ	first-order rate constant [T^{-1}]
λ_{ij}	apparent thermal conductivity tensor of the soil [MLT^{-3}K^{-1}] (e.g. Wm^{-1}K^{-1})
λ_L	longitudinal thermal dispersivity [L]
λ_T	transverse thermal dispersivity [L]
λ_0	thermal conductivity of porous medium in the absence of flow [MLT^{-3}K^{-1}] (e.g. W m^{-1}K^{-1})
μ_g	first-order rate constant for solutes in the gas phase [T^{-1}]
μ_{ref}	dynamic viscosity at reference temperature T_{ref} [MT^{-1}L^{-1}]
μ_s	first-order rate constant for solutes adsorbed onto the solid phase [T^{-1}]
μ_T	dynamic viscosity at temperature T [MT^{-1}L^{-1}]
μ_w	first-order rate constant for solutes in the liquid phase [T^{-1}]
μ_g'	first-order rate constant for chain solutes in the gas phase [T^{-1}]
μ_s'	first-order rate constant for chain solutes adsorbed onto the solid phase [T^{-1}]
μ_w'	first-order rate constant for chain solutes in the liquid phase [T^{-1}]
ξ_i	activity coefficient on the exchange surfaces [MM^{-1}] (kg mol^{-1})
ρ	bulk density of porous medium [ML^{-3}]
ρ_d	dimensionless ratio between the side of the square in the finite element mesh surrounding the drain, D, and the effective diameter of a drain, d_e [-]
ρ_{ref}	density of soil water at reference temperature T_{ref} [ML^{-3}]
ρ_T	density of soil water at temperature T [ML^{-3}]
σ_{ref}	surface tension at reference temperature T_{ref} [MT^{-2}]
σ_T	surface tension at temperature T [MT^{-2}]
σ_1	prescribed flux boundary condition at boundary Γ_N [LT^{-1}]
σ_2	prescribed gradient boundary condition at boundary Γ_G [-]
τ_a	tortuosity factor in the gas phase [-]
τ_w	tortuosity factor in the liquid phase [-]
ϕ_n	linear basis functions [-]
ϕ_n^u	upstream weighted basis functions [-]
ψ	prescribed pressure head boundary condition at boundary Γ_D [L]
ω	first-order adsorption rate constant [T^{-1}]
ω_a	angle between principal direction of K_1^A and the x-axis of the global coordinate system

	[-]
ω_s	performance index used as a criterion to minimize or eliminate numerical oscillations [-]
Ω	flow region
Ω_e	domain occupied by element *e*
Ω_R	region occupied by the root zone

[+] for plane and axisymmetric flow, respectively

GENERAL INTRODUCTION AND OVERVIEW OF HYDRUS-2D

The importance of the unsaturated zone as an integral part of the hydrological cycle has long been recognized. The zone plays an inextricable role in many aspects of hydrology, including infiltration, soil moisture storage, evaporation, plant water uptake, groundwater recharge, runoff and erosion. Initial studies of the unsaturated (vadose) zone focused primarily on water supply studies, inspired in part by attempts to optimally manage the root zone of agricultural soils for maximum crop production. Interest in the unsaturated zone has dramatically increased in recent years because of growing concern that the quality of the subsurface environment is being adversely affected by agricultural, industrial and municipal activities. Federal, state and local action and planning agencies, as well as the public at large, are now scrutinizing the intentional or accidental release of surface-applied and soil-incorporated chemicals into the environment. Fertilizers and pesticides applied to agricultural lands inevitably move below the soil root zone and may contaminate underlying groundwater reservoirs. Chemicals migrating from municipal and industrial disposal sites also represent environmental hazards. The same is true for radionuclides emanating from energy waste disposal facilities.

The past several decades has seen considerable progress in the conceptual understanding and mathematical description of water flow and solute transport processes in the unsaturated zone. A variety of analytical and numerical models are now available to predict water and/or solute transfer processes between the soil surface and the groundwater table. The most popular models remain the Richards' equation for variably saturated flow, and the Fickian-based convection-dispersion equation for solute transport. Deterministic solutions of these classical equations have been used, and likely will continue to be used in the near future, for (1) predicting water and solute movement in the vadose zone, (2) analyzing specific laboratory or field experiments involving unsaturated water flow and/or solute transport, and (3) extrapolating information from a limited number of field experiments to different soil, crop and climatic conditions, as well as to different soil and water management schemes.

Once released into the subsurface environment, industrial and agricultural chemicals are generally subjected to a large number of simultaneous physical, chemical, and biological processes, including sorption-desorption, volatilization, photolysis, and biodegradation, as well as their kinetics. The extent of degradation, sorption and volatilization largely determines the persistence of a pollutant in the subsurface [*Chiou*, 1989]. For example, the fate of organic chemicals in soils is known to be strongly affected by the kinetics of biological degradation. *Alexander and Scow* [1989] gave a review of some of the equations used to represent the kinetics of biodegradation. These equations include zero-order, half-order, first-order, three-half-order, mixed-order, logistic, logarithmic, Michaelis-Menton, and Monod type (with or without growth) expressions. While most of these expressions have a theoretical bases, they are commonly used only in an empirical fashion by fitting the equations to observed data. Zero- and first-order kinetic equations remain the most popular for describing biodegradation of organic compounds, mostly because of their simplicity and the ease at which they can be incorporated in solute transport models. Conditions for the application of these two equations are described by *Alexander and Scow* [1989].

One special group of degradation reactions involves decay chains in which solutes are subject to sequential (or consecutive) decay reactions. Problems of solute transport involving sequential first-order decay reactions frequently occur in soil and groundwater systems. Examples are the migration of various radionuclides [*Lester et al.*, 1975; *Rogers,* 1978; *Gureghian*, 1981; *Gureghian and Jansen*, 1983], the simultaneous movement of interacting nitrogen species [*Cho*, 1971; *Misra et al.*, 1974; *Wagenet et al.*, 1976; *Tillotson et al.*, 1980], organic phosphate transport [*Castro and Rolston*, 1977], and the transport of certain pesticides and their metabolites [*Bromilow and Leistra,* 1980; *Wagenet and Hutson*, 1987].

While in the past most pesticides were regarded as involatile, volatilization is now increasingly recognized as being an important process affecting the fate of pesticides in field soils [*Glotfelty and Schomburg*, 1989; *Spencer*, 1991]. Another process affecting pesticide fate and transport is the relative reactivity of solutes in the sorbed and solution phases. Several processes such as gaseous and liquid phase molecular diffusion, and convective-dispersive transport, act only on solutes that are not adsorbed. Degradation of organic compounds likely occurs mainly, or even exclusively, in the liquid phase [*Pignatello*, 1989]. On the other side, radioactive decay takes place equally in the solution and adsorbed phases, while other reactions or transformations may occur only or primarily in the sorbed phase.

Several analytical solutions have been published for simplified transport systems involving consecutive decay reactions [*Cho*, 1971; *Wagenet et al.*, 1976; *Harada et al.*, 1980; *Higashi and Pigford,* 1980; *van Genuchten*, 1985]. Unfortunately, analytical solutions for more complex situations, such as for transient water flow or the nonequilibrium solute transport with nonlinear reactions, are not available and/or cannot be derived, in which case numerical models must be employed. To be useful, such numerical models must allow for different reaction rates to take place in the solid, liquid, and gaseous phases, as well as for a correct distribution of the solutes among the different phases.

The purpose of this report is to document version 2.0 of the HYDRUS-2D software package simulating two-dimensional variably-saturated water flow, heat movement, and transport of solutes involved in sequential first-order decay reactions. The software package consists of the computer program HYDRUS2, version 2.0, and the interactive graphics-based user interface HYDRUS2D, version 2.0. The HYDRUS2 program numerically solves the Richards' equation for saturated-unsaturated water flow and convection-dispersion type equations for heat and solute transport. The flow equation incorporates a sink term to account for water uptake by plant roots. The heat transport equation considers movement by conduction as well as convection with flowing water. The governing convection-dispersion solute transport equations are written in a very general form by including provisions for nonlinear nonequilibrium reactions between the solid and liquid phases, and linear equilibrium reaction between the liquid and gaseous phases. Hence, both adsorbed and volatile solutes such as pesticides can be considered. The solute transport equations also incorporate the effects of zero-order production, first-order degradation independent of other solutes, and first-order decay/production reactions that provides the required coupling between the solutes involved in the sequential first-order chain. The transport models also account for convection and dispersion in the liquid phase, as well as for diffusion in the gas phase, thus permitting one to simulate solute transport simultaneously in both the liquid and gaseous phases. HYDRUS-2D at present considers up to five solutes which can be either coupled in a unidirectional chain or may move independently of each other. Physical nonequilibrium solute transport can be accounted for by assuming a two-region, dual porosity type formulation which partition the liquid phase into mobile and immobile regions.

The program may be used to analyze water and solute movement in unsaturated, partially saturated, or fully saturated porous media. HYDRUS-2D can handle flow domains delineated by irregular boundaries. The flow region itself may be composed of nonuniform soils having an arbitrary degree of local anisotropy. Flow and transport can occur in the vertical plane, the horizontal plane, or in a three-dimensional region exhibiting radial symmetry about a vertical axis. The water flow part of the model considers prescribed head and flux boundaries, boundaries controlled by atmospheric conditions, free drainage boundary conditions, as well as a simplified representation of nodal drains using results of electric analog experiments. First- or third-type boundary conditions can be implemented in both the solute and heat transport parts of the model. In addition, HYDRUS-2D implements a Marquardt-Levenberg type parameter estimation scheme for inverse estimation of soil hydraulic and/or solute transport and reaction parameters from measured transient or steady-state flow and/or transport data.

The governing flow and transport equations are solved numerically using Galerkin-type linear finite element schemes. Depending upon the size of the problem, the matrix equations resulting from discretization of the governing equations are solved using either Gaussian elimination for banded matrices, or the conjugate gradient method for symmetric matrices and the ORTHOMIN method for asymmetric matrices [*Mendoza et al.*, 1991]. The program is an extension of the variably saturated flow codes SWMS_2D of *Šimůnek et al.* [1992] and CHAIN_2D of *Šimůnek and van Genuchten* [1994], which in turn were based in part on the early numerical work of *Vogel* [1987] and Neuman and colleagues [*Neuman*, 1972, 1973, *Neuman et al*., 1974; *Neuman*, 1975; *Davis and Neuman*, 1983].

Even with well-documented multi-dimensional numerical computer models available, one major problem which often prevents the use of such codes is the extensive work required for data preparation, finite element grid design, and graphical presentation of output results. Hence, a more widespread use of multi-dimensional models requires techniques which make it easier to create, manipulate and display large data files, and which facilitate interactive data management. Introducing such techniques will free users from cumbersome manual data processing, and should enhance the efficiency in which programs are implemented for a particular example. To avoid or simplify the preparation and management of relatively complex input data files for two-dimensional (or three-dimensional axi-symmetrical) flow problems, and to graphically display the final simulation results, we developed an interactive graphics-based user-friendly interface for the MS Windows 95, 98, and/or NT environment. The interface is connected to HYDRUS2, a version of CHAIN_2D specifically written for use with the graphics interface.

The HYDRUS-2D software is distributed in the following forms:

1) **Option A** includes the executable code HYDRUS2 (an update of the former CHAIN_2D code) and a graphics-based user interface, HYDRUS2D, so as to facilitate data preparation and output display in the MS WINDOWS 95, 98, and/or NT environments. A mesh generator for a relatively simple rectangular domain geometry is made part of option A. The user interface is written in MS Visual C++. Because HYDRUS2 was written in Microsoft FORTRAN, this code uses several extensions that are not part of ANSI-standard FORTRAN, such as dynamically allocated arrays. Option A is distributed on a CD ROM containing all necessary files needed to run the interface, the input and output files of seven examples discussed in this report, plus many additional examples which illustrate the interface and the program in its full complexity.

2) **Option B** consists of version A, but further augmented with a CAD program MESHGEN2D for designing a more general domain geometry, and its discretization into an unstructured finite element mesh for a variety of problems involving variably-saturated subsurface flow and transport. Option B is also distributed on a CD ROM.

A general overview of the graphics-based interface is described in section B. In addition to the detailed description in Section B, extensive on-line help files are available with each module of the user interface.

PART A

The HYDRUS2 Code for Simulating the Two-Dimensional Movement of Water, Heat, and Multiple Solute in Variably-Saturated Porous Media

Version 2.0

by

J. Šimůnek and M. Th. van Genuchten

1. INTRODUCTION

This part of the manual gives a detailed description of the HYDRUS2 version 2.0 computer code. HYDRUS2 numerically solves the Richards' equation for variably-saturated water flow and convection-dispersion type equations for heat and solute transport. The flow equation incorporates a sink term to account for water uptake by plant roots. The heat transport equation considers transport due to conduction and convection with flowing water. The solute transport equations consider convective-dispersive transport in the liquid phase, as well as diffusion in the gaseous phase. The transport equations also include provisions for nonlinear nonequilibrium reactions between the solid and liquid phases, linear equilibrium reactions between the liquid and gaseous phases, zero-order production, and two first-order degradation reactions: one which is independent of other solutes, and one which provides the coupling between solutes involved in sequential first-order decay reactions. In addition, physical nonequilibrium solute transport can be accounted for by assuming a two-region, dual-porosity type formulation which partition the liquid phase into mobile and immobile regions. The program may be used to analyze water and solute movement in unsaturated, partially saturated, or fully saturated porous media. The flow region may be composed of nonuniform soils. Flow and transport can occur in the vertical plane, the horizontal plane, or in a three-dimensional region exhibiting radial symmetry about a vertical axis. The water flow part of the model can deal with prescribed head and flux boundaries, boundaries controlled by atmospheric conditions, as well as free drainage boundary conditions. The governing flow and transport equations are solved numerically using Galerkin-type linear finite element schemes. The current version 2.0 of HYDRUS2 also includes a Marquardt-Levenberg parameter optimization algorithm for inverse estimation of soil hydraulic and/or solute transport and reaction parameters from measured transient or steady-state flow and/or transport data.

Part B gives a general overview of the HYDRUS2D graphics-based interface.

2. VARIABLY SATURATED WATER FLOW

2.1. *Governing Flow Equation*

Consider two-dimensional isothermal Darcian flow of water in a variably saturated rigid porous medium and assume that the air phase plays an insignificant role in the liquid flow process. The governing flow equation for these conditions is given by the following modified form of the Richards' equation:

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial x_i}\left[K \left(K_{ij}^{A} \frac{\partial h}{\partial x_j} + K_{iz}^{A} \right) \right] - S \tag{2.1}$$

where θ is the volumetric water content [L^3L^{-3}], h is the pressure head [L], S is a sink term [T^{-1}], x_i (i=1,2) are the spatial coordinates [L], t is time [T], K_{ij}^A are components of a dimensionless anisotropy tensor $\mathbf{K}^A$, and K is the unsaturated hydraulic conductivity function [LT^{-1}] given by

$$K(h,x,z) = K_s(x,z)\, K_r(h,x,z) \tag{2.2}$$

where K_r is the relative hydraulic conductivity and K_s the saturated hydraulic conductivity [LT^{-1}]. The anisotropy tensor K_{ij}^A in (2.1) is used to account for an anisotropic medium. The diagonal entries of K_{ij}^A equal one and the off-diagonal entries zero for an isotropic medium. If (2.1) is applied to planar flow in a vertical cross-section, x_1=x is the horizontal coordinate and x_2=z is the vertical coordinate, the latter taken to be positive upward. Einstein's summation convention is used in (2.1) and throughout this report. Hence, when an index appears twice in an algebraic term, this particular term must be summed over all possible values of the index.

2.2. *Root Water Uptake*

The sink term, S, in (2.1) represents the volume of water removed per unit time from a unit volume of soil due to plant water uptake. *Feddes et al.* [1978] defined S as

$$S(h) = \alpha(h)\, S_p \tag{2.3}$$

where the water stress response function $\alpha(h)$ is a prescribed dimensionless function (Fig. 2.1) of the soil water pressure head ($0 \leq \alpha \leq 1$), and S_p is the potential water uptake rate [T^{-1}]. Figure 2.1. gives a schematic plot of the stress response function as used by *Feddes et al.* [1978]. Notice that water uptake is assumed to be zero close to saturation (i.e., wetter than some arbitrary "anaerobiosis point", h_1). For $h<h_4$ (the wilting point pressure head), water uptake is also assumed to be zero. Water uptake is considered optimal between pressure heads h_2 and h_3, whereas for pressure head between h_3 and h_4 (or h_1 and h_2), water uptake decreases (or increases) linearly with h. The variable S_p in (2.3) is equal to the water uptake rate during periods of no water stress when α (h)=1. *van Genuchten* [1987] expanded the formulation of Feddes by including osmotic stress as follows

$$S(h, h_\phi) = \alpha(h, h_\phi)\, S_p \tag{2.4}$$

where h_ϕ is the osmotic head [L], which is assumed here to be given by a linear combination of the concentrations, c_i, of all solutes present, i.e.,

$$h_\phi = a_i\, c_i \tag{2.5}$$

in which a_i are experimental coefficients [L^4M] converting concentrations into osmotic heads. *van Genuchten* [1987] proposed an alternative S-shaped function to describe the water uptake stress response function (Fig. 2.1), and suggested that the influence of the osmotic head reduction can be either additive or multiplicative as follows

$$\alpha(h, h_\phi) = \frac{1}{1+\left(\dfrac{h+h_\phi}{h_{50}}\right)^p} \tag{2.6}$$

or

$$\alpha(h, h_\phi) = \frac{1}{1+(h/h_{50})^{p_1}}\frac{1}{1+(h_\phi/h_{\phi_{50}})^{p_2}} \tag{2.7}$$

respectively, where p, p_1, and p_2 are experimental constants. The exponent p was found to be approximately 3 when applied to salinity stress data only [*van Genuchten*, 1987]. The parameter h_{50} in (2.6) and (2.7) represents the pressure head at which the water extraction rate is reduced by 50% during conditions of negligible osmotic stress. Similarly, $h_{\phi 50}$ represents the osmotic head at which the water extraction rate is reduced by 50% during conditions of negligible water stress. Note that, in contrast to the expression of *Feddes et al.* [1978], this formulation of the stress response function, $\alpha(h,h_\phi)$, does not consider a transpiration reduction near saturation. Such a simplification seems justified when saturation occurs for only relatively short periods of time.

When the potential water uptake rate is equally distributed over a two-dimensional rectangular root domain, S_p becomes

$$S_p = \frac{1}{L_x L_z} L_t T_p \tag{2.8}$$

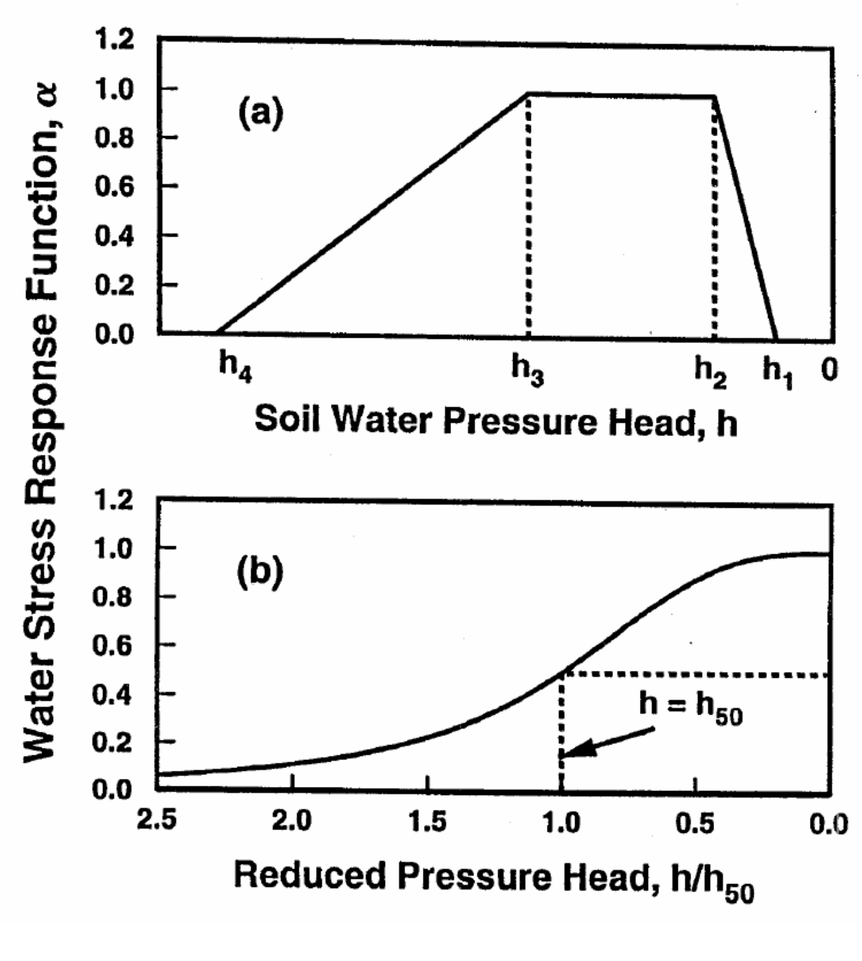


Fig. 2.1. Schematic of the plant water stress response function, $\alpha(h)$, as used by a) *Feddes et al.* [1978] and b) *van Genuchten* [1987].

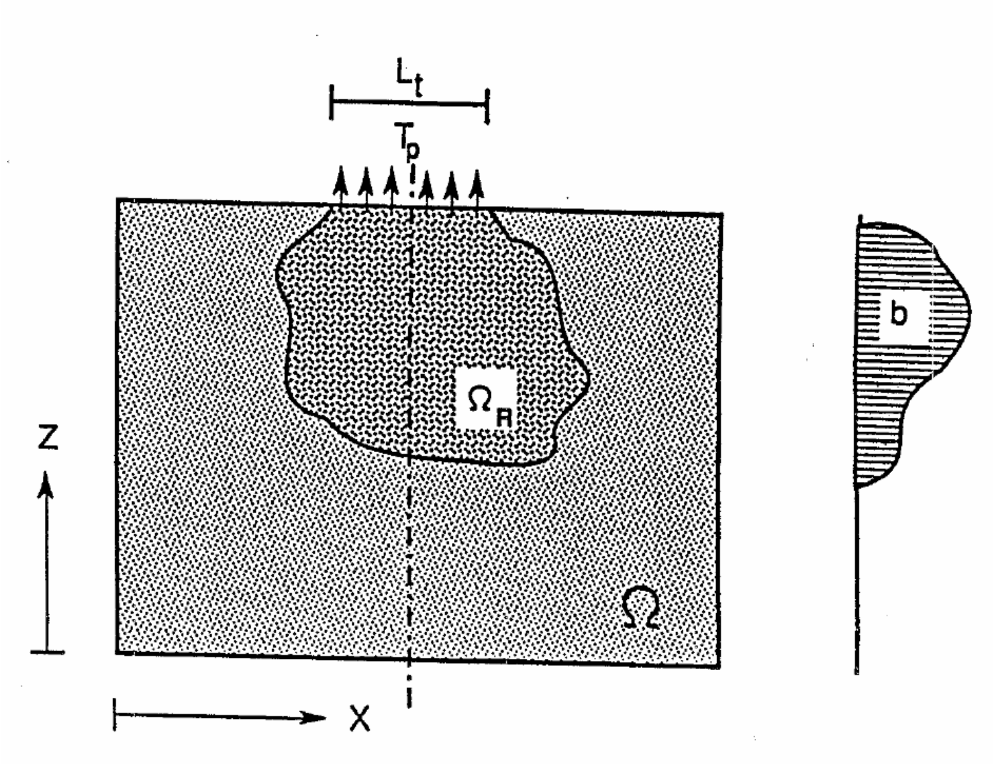


Fig. 2.2. Schematic of the potential water uptake distribution function, $b(x,z)$, in the soil root zone.

where T_p is the potential transpiration rate [LT^{-1}], L_z is the depth [L] of the root zone, L_x is the width [L] of the root zone, and L_t is the width [L] of the soil surface associated with the transpiration process. Notice that S_p reduces to T_p/L_z when L_t=L_x.

Equation (2.8) may be generalized by introducing a non-uniform distribution of the potential water uptake rate over a root zone of arbitrary shape [*Vogel*, 1987]:

$$S_p = b(x,z)L_tT_p \tag{2.9}$$

where $b(x,z)$ is the normalized water uptake distribution [L^{-2}]. This function describes the spatial variation of the potential extraction term, S_p, over the root zone (Fig. 2.2), and is obtained from $b'(x,z)$ as follows

$$b(x,z) = \frac{b'(x,z)}{\int\limits_{\Omega_R} b'(x,z)d\Omega} \tag{2.10}$$

where Ω_R is the region occupied by the root zone, and $b'(x,z)$ is an arbitrarily prescribed distribution function. Normalizing the uptake distribution ensures that $b(x,z)$ integrates to unity over the flow domain, i.e.,

$$\int_{\Omega_R} b(x,z)d\Omega = 1 \tag{2.11}$$

From (2.9) and (2.11) it follows that S_p is related to T_p by the expression

$$\frac{1}{L_t}\int_{\Omega R} S_p d\Omega = T_p \tag{2.12}$$

The actual water uptake distribution is obtained by substituting (2.9) into (2.3):

$$S(h,h_\phi,x,z) = \alpha(h,h_\phi,x,z)b(x,z)L_t T_p \tag{2.13}$$

whereas the actual transpiration rate, T_a, is obtained by integrating (2.13) as follows

$$T_a = \frac{1}{L_t}\int_{\Omega_R} Sd\Omega = T_p \int_{\Omega_R} \alpha(h,h_\phi,x,z)b(x,z)d\Omega \tag{2.14}$$

2.3. *The Unsaturated Soil Hydraulic Properties*

The unsaturated soil hydraulic properties, $\theta(h)$ and $K(h)$, in (2.1) are in general highly nonlinear functions of the pressure head. HYDRUS2 permits the use of three different analytical models for the hydraulic properties [*Brooks and Corey*, 1964; *van Genuchten*, 1980; and *Vogel and Císlerová*, 1988].

The soil water retention, $\theta(h)$, and hydraulic conductivity, $K(h)$, functions according to *Brooks and Corey* [1964] are given by

$$S_e = \begin{cases} |\alpha h|^{-n} & h < -1/\alpha \\ 1 & h \geq -1/\alpha \end{cases} \tag{2.15}$$

$$K = K_s S_e^{2/n+l+2} \tag{2.16}$$

respectively, where S_e is the effective water content,

$$S_e = \frac{\theta - \theta_r}{\theta_s - \theta_r} \tag{2.17}$$

in which θ_r and θ_s denote the residual and saturated water content, respectively; K_s is the saturated hydraulic conductivity, α is the inverse of the air-entry value (or bubbling pressure), n is a pore-size distribution index, and l is a pore-connectivity parameter assumed to be 2.0 in the original study of *Brooks and Corey* [1964]. The parameters α, n and l in HYDRUS2 are considered to be merely empirical coefficients affecting the shape of the hydraulic functions.

HYDRUS2 also implements the soil-hydraulic functions of *van Genuchten* [1980] who used the statistical pore-size distribution model of *Mualem* [1976] to obtain a predictive equation for the unsaturated hydraulic conductivity function in terms of soil water retention parameters. The expressions of *van Genuchten* [1980] are given by

$$\theta(h) = \begin{cases} \theta_r + \dfrac{\theta_s - \theta_r}{[1+|\alpha h|^n]^m} & h < 0 \\ \theta_s & h \geq 0 \end{cases} \tag{2.18}$$

$$K(h) = K_s S_e^l [1 - (1 - S_e^{1/m})^m]^2 \tag{2.19}$$

where

$$m = 1 - 1/n \ , \quad n > 1 \tag{2.20}$$

The above equations contain five independent parameters: θ_r, θ_s, α, n, and K_s. The pore-connectivity parameter l in the hydraulic conductivity function was estimated [*Mualem*, 1976] to be about 0.5 as an average for many soils.

A third set of hydraulic equations implemented in HYDRUS2 are those by *Vogel and Císlerová* [1988] who modified the equations of *van Genuchten* [1980] to add flexibility in the description of the hydraulic properties near saturation. The soil water retention, $\theta(h)$, and hydraulic conductivity, $K(h)$, functions of *Vogel and Císlerová* [1988] are given by (Fig. 2.3)

$$\theta(h) = \begin{cases} \theta_a + \dfrac{\theta_m - \theta_a}{(1+|\alpha h|^n)^m} & h < h_s \\ \theta_s & h \geq h_s \end{cases} \tag{2.21}$$

and

$$K(h) = \begin{cases} K_s K_r(h) & h \leq h_s \\ K_k + \dfrac{(h-h_k)(K_s-K_k)}{h_s-h_k} & h_k < h < h_s \\ K_s & h \geq h_s \end{cases} \tag{2.22}$$

respectively, where

$$K_r = \frac{K_k}{K_S}\left(\frac{S_e}{S_{ek}}\right)^{1/2}\left[\frac{F(\theta_r)-F(\theta)}{F(\theta_r)-F(\theta_k)}\right]^2 \tag{2.23}$$

$$F(\theta) = \left[1-\left(\frac{\theta-\theta_a}{\theta_m-\theta_a}\right)^{1/m}\right]^m \tag{2.24}$$

$$S_{ek} = \frac{\theta_k-\theta_r}{\theta_s-\theta_r} \tag{2.25}$$

The above equations allow for a non-zero minimum capillary height, h_s, by replacing the parameter θ_s in van Genuchten's retention function by a fictitious (extrapolated) parameter θ_m slightly larger than θ_s as shown in Fig. 2.3. While this change from θ_s to θ_m has little or no effect on the retention curve, the effect on the shape and value of the hydraulic conductivity function can be considerable, especially for fine-textured soils when n is relatively small (e.g., $1.0 < n < 1.3$). To increase the flexibility of the analytical expressions, the parameter θ_r in the retention function was replaced by the fictitious (extrapolated) parameter $\theta_a \leq \theta_r$. The approach maintains the physical meaning of θ_r and θ_s as measurable quantities. Equation (2.23) assumes that the predicted hydraulic conductivity function is matched to a measured value of the hydraulic conductivity, K_k=K(θ_k), at some water content, θ_k, less that or equal to the saturated water content, i.e., $\theta_k \leq \theta_s$ and $K_k \leq K_s$ [*Vogel and Císlerová*, 1988; *Luckner et al.*, 1989]. Inspection of (2.21) through (2.23) shows that the hydraulic characteristics contain 9 unknown parameters: θ_r, θ_s, θ_a, θ_m, α, n, K_s , K_k , and θ_k . When θ_a=θ_r, θ_m=θ_k=θ_s and K_k=K_s, the soil hydraulic functions of *Vogel and Císlerová* [1988] reduce to the original expressions of *van Genuchten* [1980].

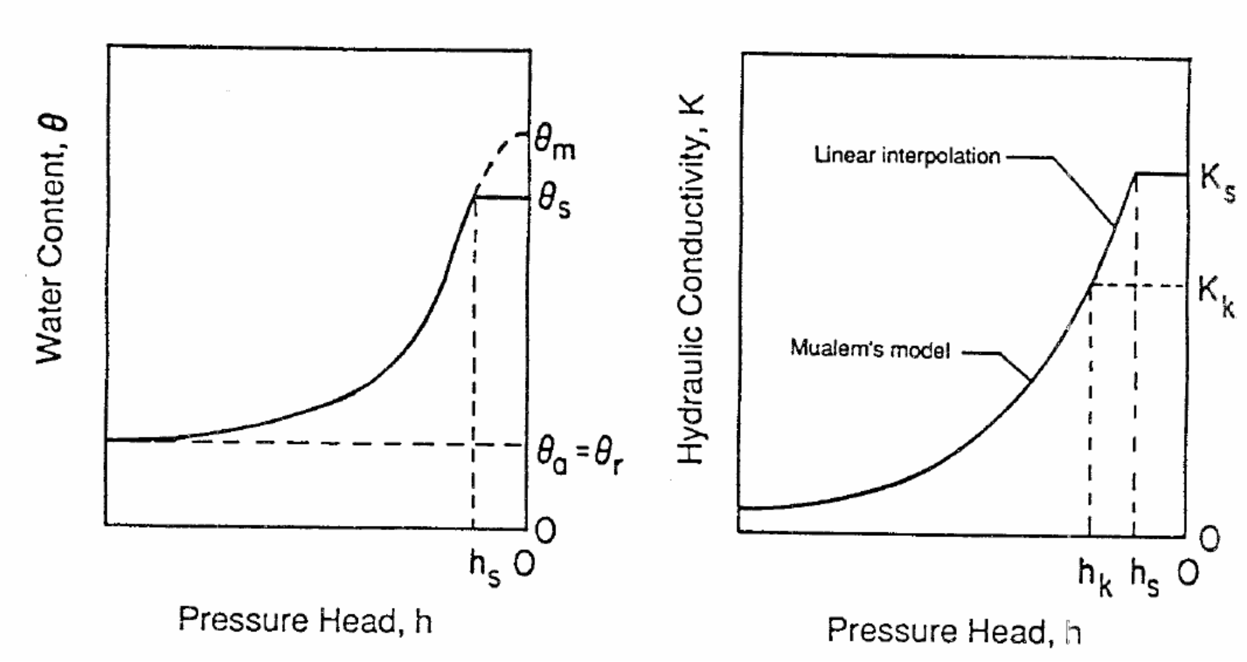


Fig. 2.3. Schematics of the soil water retention (a) and hydraulic conductivity (b) functions as given by equations (2.21) and (2.22), respectively.

2.4. *Scaling in the Soil Hydraulic Functions*

HYDRUS2 implements a scaling procedure designed to simplify the description of the spatial variability of the unsaturated soil hydraulic properties in the flow domain. The code assumes that the hydraulic variability in a given area can be approximated by means of a set of linear scaling transformations which relate the individual soil hydraulic characteristics $\theta(h)$ and $K(h)$ to reference characteristics $\theta^*(h^*)$ and $K^*(h^*)$. The technique is based on the similar media concept introduced by

Miller and Miller [1956] for porous media which differ only in the scale of their internal geometry. The concept was extended by *Simmons et al.* [1979] to materials which differ in morphological properties, but which exhibit 'scale-similar' soil hydraulic functions. Three independent scaling factors are embodied in HYDRUS2. These three scaling parameters may be used to define a linear model of the actual spatial variability in the soil hydraulic properties as follows [*Vogel et al.*, 1991]:

$$\begin{aligned} K(h) &= \alpha_K K^*(h^*) \\ \theta(h) &= \theta_r + \alpha_\theta\left[\theta^*(h^*) - \theta_r^*\right] \\ h &= \alpha_h h^* \end{aligned} \tag{2.26}$$

in which, for the most general case, α_θ, α_h and α_K are mutually independent scaling factors for the water content, the pressure head and the hydraulic conductivity, respectively. Less general scaling methods arise by invoking certain relationships between α_θ, α_h and/or α_K. For example, the original Miller-Miller scaling procedure is obtained by assuming α_θ=1 (with θ_r* = θ_r), and α_K=α_h^{-2}. A detailed discussion of the scaling relationships given by (2.26), and their application to the hydraulic description of heterogeneous soil profiles, is given by *Vogel et al.* [1991].

2.5. *Temperature Dependence of the Soil Hydraulic Functions*

A similar scaling technique as described above is used in HYDRUS2 to express the temperature dependence of the soil hydraulic functions. Based on capillary theory that assumes that the influence of temperature on the soil water pressure head can be quantitatively predicted from the influence of temperature on surface tension, *Philip and de Vries* [1957] derived the following equation

$$\frac{dh}{dT} = \frac{h}{\sigma}\frac{d\sigma}{dT} \tag{2.27}$$

where T is temperature [K] and σ is the surface tension at the air-water interface [MT^{-2}]. From (2.27) it follows that

$$h_T = \frac{\sigma_T}{\sigma_{ref}} h_{ref} = \alpha_h^* h_{ref} \tag{2.28}$$

where h_T and h_{ref} (σ_T and σ_{ref}) are pressure heads (surface tensions) at temperature T and reference temperature T_{ref}, respectively; and α_h^* is the temperature scaling factor for the pressure head.

Following *Constantz* [1982], the temperature dependence of the hydraulic conductivity can be expressed as

$$K_T(\theta) = \frac{\mu_{ref}}{\mu_T} \frac{\rho_T}{\rho_{ref}} K_{ref}(\theta) = \alpha_K^* K_{ref}(\theta) \tag{2.29}$$

where K_{ref} and K_T denote hydraulic conductivities at the reference temperature T_{ref} and soil temperature T, respectively; μ_{ref} and μ_T (ρ_{ref} and ρ_T) represent the dynamic viscosity [$ML^{-1}T^{-1}$] (density of soil water [ML^{-3}]) at temperatures T_{ref} and T, respectively; and α_K^* is the temperature scaling factor for the hydraulic conductivity.

2.6. *Hysteresis in the Soil Hydraulic Properties*

Applications of unsaturated flow models often assume unique, single-valued (non-hysteretic) functions for $\theta(h)$ and $K(h)$ to characterize the hydraulic properties at a certain point in the soil profile. While such a simplification may be acceptable for some flow simulations, many cases require a more realistic description involving hysteresis in the soil hydraulic properties. The HYDRUS2 code incorporates hysteresis by using the empirical model introduced by *Scott et al.* [1983]. This model was also employed by *Kool and Parker* [1987], who modified the formulation to account for air entrapment. The present version of HYDRUS2 further extends the model of *Kool and Parker* according to *Vogel et al.* [1996] by considering also hysteresis in the hydraulic conductivity function.

The adopted procedure for modeling hysteresis in the retention function requires that both the main drying and main wetting curves be known (Fig. 2.4). These two curves are described with (2.19) using the parameter vectors (θ_r^d, θ_s^d, θ_m^d, α^d, n^d) and (θ_r^w, θ_s^w, θ_m^w, α^w, n^w), respectively, where the subscripts d and w indicate wetting and drying, respectively. The following restrictions are expected to hold in most practical applications:

$$\theta_r^{\,d} = \theta_r^{\,w} \quad ; \quad \alpha^{\,d} \leq \alpha^{\,w} \tag{2.30}$$

We also invoke the often-assumed restriction

$$n^d = n^w \tag{2.31}$$

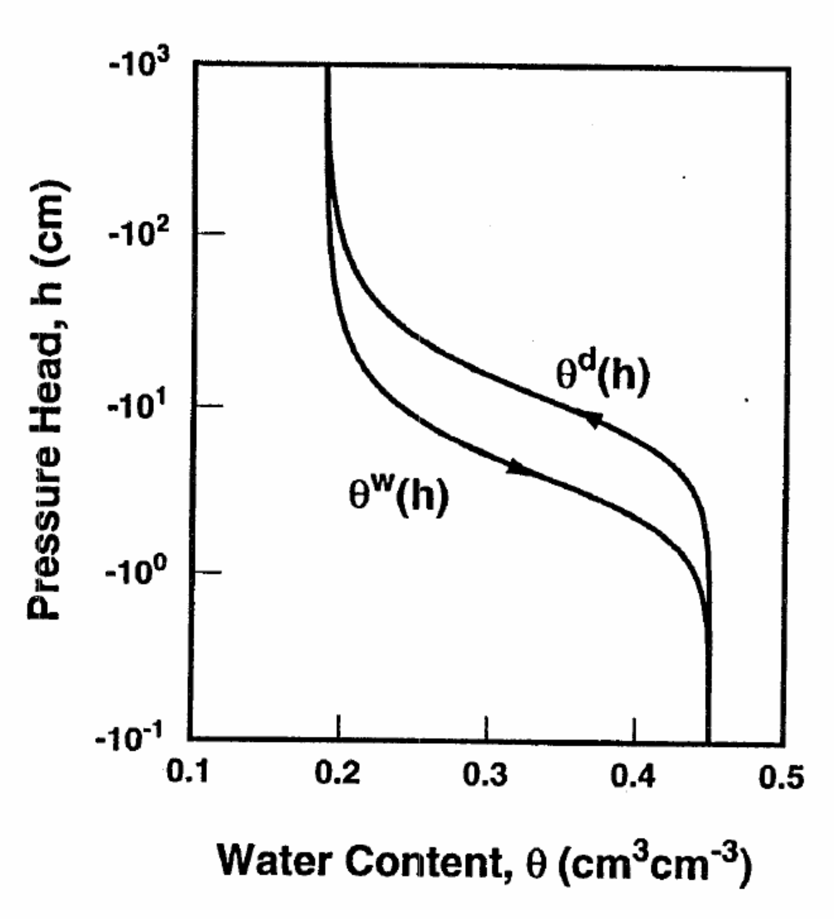


Fig. 2.4. Example of a water retention curve showing hysteresis. Shown are the boundary wetting curve, $\theta^w(h)$, and the boundary drying curve, $\theta^d(h)$.

If data are lacking, one may use $\alpha^w = 2\alpha^d$ as a reasonable first approximation [*Kool and Parker*, 1987; *Nielsen and Luckner*, 1992]. We further assume

$$\theta_m^w = \theta_r + \frac{\theta_s^w - \theta_r}{\theta_s^d - \theta_r}(\theta_m^d - \theta_r) \qquad (2.32)$$

so that the parameters θ_s and α are the only independent parameters describing hysteresis in the retention function. According to the hysteresis model, drying scanning curves are scaled from the main drying curve, and wetting scanning curves from the main wetting curve. The scaling factors for the drying scanning curves can be obtained by considering the main drying curve as a reference curve in scaling equation (2.26) (keeping $\alpha_h = 1$ to scale only in the water content direction), i.e.:

$$\theta(h) = \theta_r' + \alpha_\theta[\theta^d(h) - \theta_r^d] \qquad (2.33)$$

and forcing each scanning curve, $\theta(h)$, to pass through the point (θ_Δ, h_Δ) characterizing the latest reversal from wetting to drying. Substituting this reversal point into (2.33), and assuming that $\theta_r = \theta_r^d$, leads to

$$\alpha_\theta = \frac{\theta_\Delta - \theta_r}{\theta^d(h_\Delta) - \theta_r} \qquad (2.34)$$

Note that the scaling procedure results in a fictitious value of the parameter θ_s' for the drying scanning curve (this parameter may be located outside of the main hysteresis loop). The scaling relationship is similarly for the wetting scanning curves

$$\theta(h) = \theta_r' + \alpha_\theta[\theta^w(h) - \theta_r] \qquad (2.35)$$

in which the fictitious parameter θ_r' is now used (again possibly scaled outside of the main loop). The scaling factor α_θ for a particular scanning curve can be obtained by substituting the reversal point (θ_Δ, h_Δ) and the full saturation point (θ_s, 0) into (2.35), and subtracting the two resulting equations to eliminate θ_r' to give

$$\alpha_\theta = \frac{\theta_\Delta - \theta_s}{\theta^w(h_\Delta) - \theta_s^w} \tag{2.36}$$

The parameter θ_r' is subsequently determined from (2.35) as $\theta_r' = \theta_s - \alpha_\theta(\theta_s^w - \theta_r)$. If the main hysteresis loop is not closed at saturation, the water content at saturation for a particular wetting scanning curve is evaluated using the empirical relationship of *Aziz and Settari* [1979]

$$\theta_s = \theta_s^d - \frac{\theta_s^d - \theta_\Delta}{1 + R\,(\theta_s^d - \theta_\Delta)} \quad ; \quad R = \frac{1}{\theta_s^d - \theta_s^w} - \frac{1}{\theta_s^d - \theta_r^d} \tag{2.37}$$

An analogous hysteretic procedure can be applied to the unsaturated hydraulic conductivity function *K*(*h*). The main branches $K^d(h)$ and $K^w(h)$ of the hysteresis loop are characterized by the same set of parameters as the corresponding retention curves $\theta^d(h)$ and $\theta^w(h)$, and by the saturated conductivities K_s^d and K_s^w according to Eq. (2.19). For drying scanning curves we obtain from (2.26)

$$K(h) = \alpha_K\, K^d(h) \tag{2.38}$$

From knowledge of the reversal point (h_Δ, K_Δ) we obtain

$$\alpha_K = \frac{K_\Delta}{K^d(h_\Delta)} \tag{2.39}$$

For a wetting scanning curve we have now

$$K(h) = K_r`+\alpha_K\, K^w(h) \tag{2.40}$$

where K_r' is a fictitious parameter. Substituting the reversal point (h_Δ, K_Δ) and the saturation point (0, K_s) into (2.40) and solving for α_K yields

$$\alpha_K = \frac{K_\Delta - K_s}{K^w(h_\Delta) - K_s^w} \tag{2.41}$$

The fictitious conductivity parameter K_r' may be obtained from (2.40) as $K_r' = K_s - \alpha_K K_s^w$. If the main hysteresis loop is not closed at saturation, the hydraulic conductivity at saturation for a wetting scanning curve is evaluated using equations similar to (2.37), i.e.,

$$K_s = K_s^d - \frac{K_s^d - K_\Delta}{1 + R\,(K_s^d - K_\Delta)} \quad ; \quad R = \frac{1}{K_s^d - K_s^w} - \frac{1}{K_s^d} \tag{2.42}$$

2.7. *Initial and Boundary Conditions*

The solution of Eq. (2.1) requires knowledge of the initial distribution of the pressure head within the flow domain, *Ω*:

$$h(x,z,t) = h_0(x,z) \quad \text{for} \quad t = 0 \tag{2.43}$$

where h_0 is a prescribed function of *x* and *z*.

HYDRUS2 implements three types of conditions to describe system-independent interactions along the boundaries of the flow region. These conditions are specified pressure head (Dirichlet type) boundary conditions of the form

$$h(x,z,t) = \psi(x,z,t) \quad \text{for} \quad (x,z) \;\varepsilon\; \Gamma_D \tag{2.44}$$

specified flux (Neumann type) boundary conditions given by

$$-[K(K_{ij}^A \frac{\partial h}{\partial x_j} + K_{iz}^A)]n_i = \sigma_1(x,z,t) \quad \text{for} \quad (x,z) \;\varepsilon\; \Gamma_N \tag{2.45}$$

and specified gradient boundary conditions

$$(K_{ij}^{A}\frac{\partial h}{\partial x_j}+K_{iz}^{A})n_i=\sigma_2(x,z,t) \quad \text{for} \quad (x,z)\ \varepsilon\ \Gamma_G \tag{2.46}$$

where Γ_D, Γ_N, and Γ_G indicate Dirichlet, Neumann, and gradient type boundary segments, respectively; ψ [L], σ_1 [LT^{-1}], and σ_2 [-] are prescribed functions of x, z and t; and n_i are the components of the outward unit vector normal to boundary Γ_N or Γ_G. As pointed out by *McCord* [1991], the use of the term "Neumann type boundary condition" for the flux boundary is not very appropriate since this term should hold for a gradient type condition (see also Section 3.2 for solute transport). However, since the use of the Neumann condition is standard in the hydrologic literature [*Neuman*, 1972; *Neuman et al.*, 1974], we shall also use this term to indicate flux boundaries throughout this report. HYDRUS2 implements the gradient boundary condition only in terms of a unit vertical hydraulic gradient simulating free drainage from a relatively deep soil profile. This situation is often observed in field studies of water flow and drainage in the vadose zone [*Sisson*, 1987; *McCord*, 1991]. *McCord* [1991] states that the most pertinent application of (2.46) is its use as a bottom outflow boundary condition for situations where the water table is situated far below the domain of interest.

In addition to the system-independent boundary conditions given by (2.44), (2.45), and (2. 46), HYDRUS2 considers three different types of system-dependent boundary conditions which cannot be defined a priori. One of these involves soil-air interfaces, which are exposed to atmospheric conditions. The potential fluid flux across these interfaces is controlled exclusively by external conditions. However, the actual flux depends also on the prevailing (transient) soil moisture conditions. Soil surface boundary conditions may change from prescribed flux to prescribed head type conditions (and vice-versa). In the absence of surface ponding, the numerical solution of (2.1) is obtained by limiting the absolute value of the flux such that the following two conditions are satisfied [*Neuman et al.*, 1974]:

$$|K(K_{ij}^{A}\frac{\partial h}{\partial x_j}+K_{iz}^{A})n_i|\leq E \tag{2.47}$$

and

$$h_A \leq h \leq h_S \tag{2.48}$$

where E is the maximum potential rate of infiltration or evaporation under the current atmospheric conditions, h is the pressure head at the soil surface, and h_A and h_S are, respectively, minimum and maximum pressure heads allowed under the prevailing soil conditions. The value for h_A is determined from the equilibrium conditions between soil water and atmospheric water vapor, whereas h_S is usually set equal to zero. HYDRUS2 assumes that any excess water on the soil surface is immediately removed. When one of the end points of (2.48) is reached, a prescribed head boundary condition will be used to calculate the actual surface flux. Methods of calculating E and h_A on the basis of atmospheric data have been discussed by *Feddes et al.* [1974].

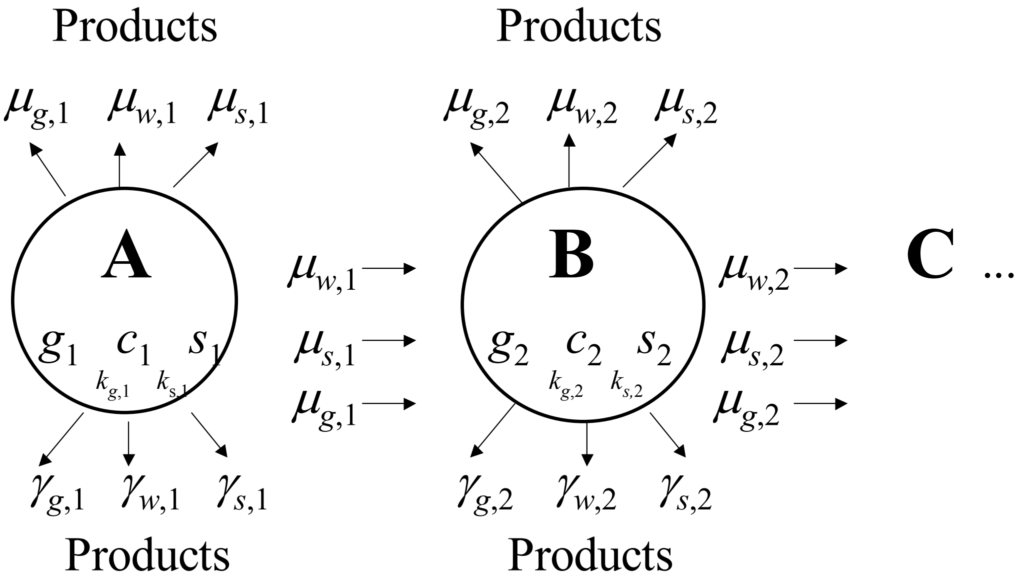
A second type of system-dependent boundary condition considered in HYDRUS2 is a seepage face through which water leaves the saturated part of the flow domain. In this case, the length of the seepage face is not known a priori. HYDRUS2 assumes that the pressure head is always uniformly equal to zero along a seepage face. Additionally, the code assumes that water leaving the saturated zone across a seepage face is immediately removed by overland flow or some other removal process.

Finally, a third class of system-dependent boundary conditions in HYDRUS2 concerns tile drains. Similarly as for seepage phase, HYDRUS2 assumes that as long as a drain is located in the saturated zone, the pressure head along the drain will be equal to zero; the drain then acts as a pressure head sink. However, the drain will behave as a nodal sink/source with zero recharge when located in the unsaturated zone. More information can be found in Section 5.3.7.

3. NONEQUILIBRIUM TRANSPORT OF SOLUTES INVOLVED IN SEQUENTIAL FIRST-ORDER DECAY REACTIONS

3.1. *Governing Transport Equation*

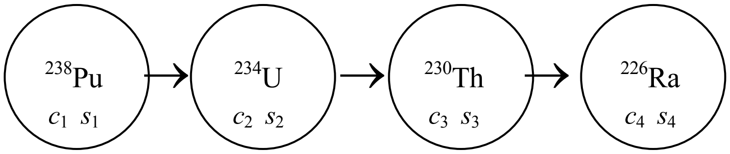
We assume that solutes can exist in all three phases (liquid, solid, and gaseous) and that the decay and production processes can be different in each phase. Interactions between the solid and liquid phases may be described by nonlinear nonequilibrium equations, while interactions between the liquid and gaseous phases are assumed to be linear and instantaneous. We further assume that the solutes are transported by convection and dispersion in the liquid phase, as well as by diffusion in the gas phase. A general structure of the system of first-order decay reactions for three solutes (A, B and C) is as follows [*Šimůnek and van Genuchten*, 1995]:



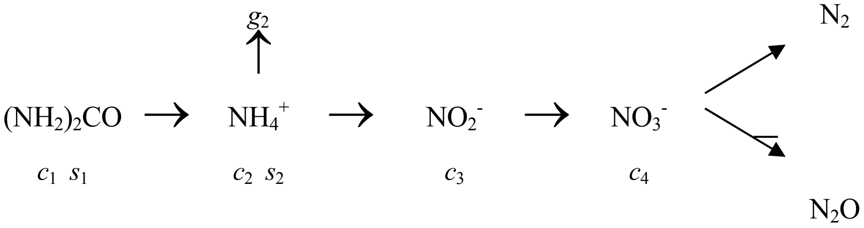
where *c*, *s*, and *g* represent concentrations in the liquid, solid, and gaseous phases, respectively; the subscripts *s*, *w*, and *g* refer to solid, liquid and gaseous phases, respectively; straight arrows represent the different zero-order (γ) and first-order (μ,μ') rate reactions, and circular arrows (k_g, k_s) indicate equilibrium distribution coefficients between phases.

Typical examples of sequential first-order decay chains are:

1. Radionuclides [*van Genuchten*, 1985]

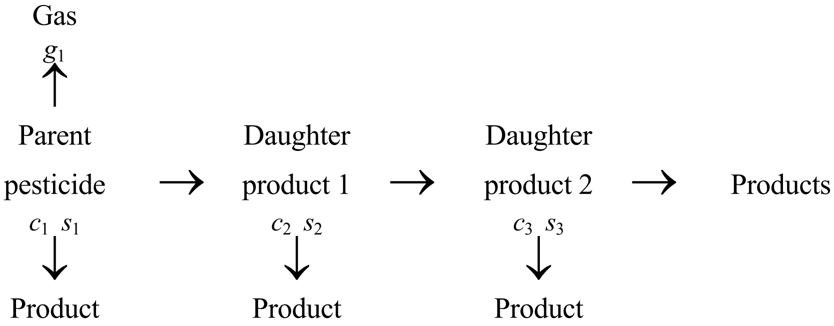


2. Nitrogen [*Tillotson et al.*, 1980]

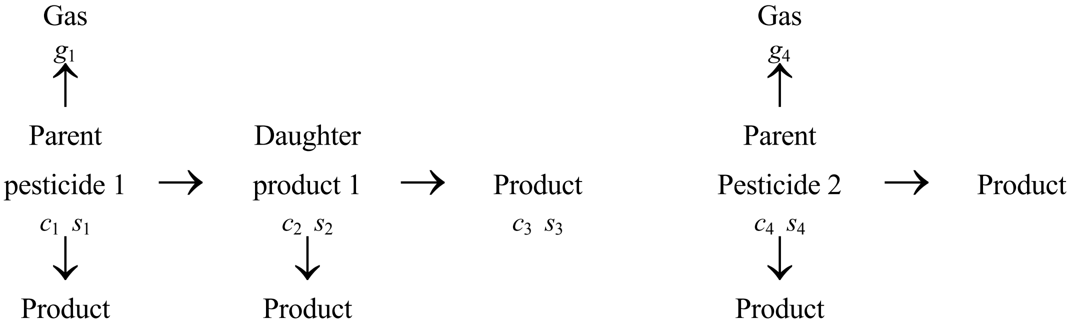


3. Pesticides [*Wagenet and Hutson*, 1987]:

a) Uninterrupted chain - one reaction path:



b) Interrupted chain - two independent reaction paths:



HYDRUS2 at present considers up to six solutes, which can be either coupled in a unidirectional chain or may move independently of each other.

The partial differential equations governing two-dimensional nonequilibrium chemical transport of solutes involved in a sequential first-order decay chain during transient water flow in a variably saturated rigid porous medium are taken as

$$\begin{aligned}\frac{\partial\theta c_1}{\partial t}+\frac{\partial\rho s_1}{\partial t}+\frac{\partial a_v g_1}{\partial t}&=\frac{\partial}{\partial x_i}\left(\theta D^w_{ij,1}\frac{\partial c_1}{\partial x_j}\right)+\frac{\partial}{\partial x_i}\left(a_v D^g_{ij,1}\frac{\partial g_1}{\partial x_j}\right)-\frac{\partial q_i c_1}{\partial x_i}-Sc_{r,1}-\\&-(\mu_{w,1}+\mu'_{w,1})\theta c_1-(\mu_{s,1}+\mu'_{s,1})\rho s_1-(\mu_{g,1}+\mu'_{g,1})a_v g_1+\gamma_{w,1}\theta+\gamma_{s,1}\rho+\gamma_{g,1}a_v\end{aligned}\tag{3.1}$$

$$\begin{aligned}\frac{\partial\theta c_k}{\partial t}+\frac{\partial\rho s_k}{\partial t}+\frac{\partial a_v g_k}{\partial t}&=\frac{\partial}{\partial x_i}\left(\theta D^w_{ij,k}\frac{\partial c_k}{\partial x_j}\right)+\frac{\partial}{\partial x_i}\left(a_v D^g_{ij,k}\frac{\partial g_k}{\partial x_j}\right)-\frac{\partial q_i c_k}{\partial x_i}-\\&-(\mu_{w,k}+\mu'_{w,k})\theta c_k-(\mu_{s,k}+\mu'_{s,k})\rho s_k-(\mu_{g,k}+\mu'_{g,k})a_v g_k+\mu'_{w,k-1}\theta c_{k-1}+\\&+\mu'_{s,k-1}\rho s_{k-1}+\mu'_{g,k-1}a_v g_{k-1}+\gamma_{w,k}\theta+\gamma_{s,k}\rho+\gamma_{g,k}a_v-Sc_{r,k}\qquad k\ \varepsilon\ (2,n_s)\end{aligned}\tag{3.2}$$

where c, s, and g are solute concentrations in the liquid [ML^{-3}], solid [MM^{-1}], and gaseous [ML^{-3}], phases, respectively; q_i is the i-th component of the volumetric flux density [LT^{-1}], μ_w, μ_s , and μ_g are first-order rate constants for solutes in the liquid, solid, and gas phases [T^{-1}], respectively; μ_w', μ_s', and μ_g' are similar first-order rate constants providing connections between individual chain species, γ_w, γ_s, and γ_g are zero-order rate constants for the liquid [ML^{-3}T^{-1}], solid [T^{-1}], and gas [ML^{-3}T^{-1}] phases, respectively; ρ is the soil bulk density [ML^{-3}], a_v is the air content [L^{3}L^{-3}], S is the sink term in the water flow equation (2.1), c_r is the concentration of the sink term [ML^{-3}], $D_{ij}{}^w$ is the dispersion coefficient tensor [L^{2}T^{-1}] for the liquid phase, and $D_{ij}{}^g$ is the diffusion coefficient tensor [L^{2}T^{-1}] for the gas phase. As before, the subscripts *w, s*, and *g* correspond with the liquid, solid and gas phases, respectively; while the subscript *k* represents the *k*th chain number, and n_s is the number of solutes involved in the chain reaction. The indicial notation used in this report assumes summations over indices i and j (i,j = 1,2), but not over index k. The nine zero- and first-order rate constants in (3.1) and (3.2) may be used to represent a variety of reactions or transformations including biodegradation, volatilization, and precipitation.

HYDRUS2 assumes nonequilibrium interaction between the solution (c) and adsorbed (s) concentrations, and equilibrium interaction between the solution (c) and gas (g) concentrations of the solute in the soil system. The adsorption isotherm relating s_k and c_k is described by a generalized nonlinear equation of the form

$$s_k = \frac{k_{s,k}\, c_k^{\beta_k}}{1+\eta_k c_k^{\beta_k}} \qquad k\, \varepsilon\, (1, n_s)$$

$$\frac{\partial s_k}{\partial t} = \frac{k_{s,k}\beta_k c_k^{\beta_k - 1}}{(1+\eta_k c_k^{\beta_k})^2}\frac{\partial c_k}{\partial t} + \frac{c_k^{\beta_k}}{1+\eta_k c_k^{\beta_k}}\frac{\partial k_{s,k}}{\partial t} - \frac{k_{s,k} c_k^{2\beta_k}}{(1+\eta_k c_k^{\beta_k})^2}\frac{\partial \eta_k}{\partial t} + \frac{k_{s,k} c_k^{\beta_k} \ln c_k}{(1+\eta_k c_k^{\beta_k})^2}\frac{\partial \beta_k}{\partial t} \tag{3.3}$$

where $k_{s,k}$ [L^3M^{-1}], β_k [-] and η_k [L^3M^{-1}] are empirical coefficients. The Freundlich, Langmuir, and linear adsorption equations are special cases of equation (3.3). When β_k=1, equation (3.3) becomes the Langmuir equation, when η_k=0, equation (3.3) becomes the Freundlich equation, and when both β_k=1 and η_k=0, equation (3.3) leads to a linear adsorption isotherm. Solute transport without adsorption is described with $k_{s,k}$=0. While the coefficients $k_{s,k}$, β_k, and η_k in equation (3.3) are assumed to be independent of concentration, they are permitted to change as a function of time through their dependency on temperature. This feature will be discussed later.

The concentrations g_k and c_k are related by a linear expression of the form

$$g_k = k_{g,k} c_k \qquad k\, \varepsilon\, (1, n_s) \tag{3.4}$$

where $k_{g,k}$ is an empirical constant [-] equal to $(K_H R_u T^A)^{-1}$ [*Stumm and Morgan*, 1981] in which K_H is Henry's Law constant [$MT^2M^{-1}L^{-2}$], R_u is the universal gas constant [$ML^2T^{-2}K^{-1}\ M^{-1}$] and T^A is absolute temperature [K].

3.1.1. *Chemical Nonequilibrium*

The concept of two-site sorption [*Selim et al.*, 1977; *van Genuchten and Wagenet*, 1989] is implemented in HYDRUS2 to permit consideration of nonequilibrium adsorption-desorption reactions. The two-site sorption concept assumes that the sorption sites can be divided into two fractions:

$$s_k = s_k^e + s_k^k \qquad k\, \varepsilon\, (1, n_s) \tag{3.5}$$

Sorption, s_k^e [MM^{-1}], on one fraction of the sites (the type-1 sites) is assumed to be instantaneous, while sorption, s_k^k [MM^{-1}], on the remaining (type-2) sites is considered to be time-dependent. At equilibrium we have for the type-1 (equilibrium) and type-2 (kinetic) sites, respectively

$$s_k^e = f s_k \qquad\qquad k\ \varepsilon\ (1,n_s) \qquad\qquad (3.6)$$

where f is the fraction of exchange sites assumed to be in equilibrium with the solution phase [-]. Because type-1 sorption sites are always at equilibrium, differentiation of (3.6) gives immediately the sorption rate for the type-1 equilibrium sites:

$$s_k^k = (1-f) s_k \qquad\qquad k\ \varepsilon\ (1,n_s) \qquad\qquad (3.7)$$

$$\frac{\partial s_k^e}{\partial t} = f \frac{\partial s_k}{\partial t} \qquad\qquad k\ \varepsilon\ (1,n_s) \qquad\qquad (3.8)$$

Sorption on the type-2 nonequilibrium sites is assumed to be a first-order kinetic rate process. Following *Toride et al.* [1993], the mass balance equation for the type-2 sites in the presence of production and degradation is given by

$$\frac{\partial s_k^k}{\partial t} = \omega_k \left[(1-f) \frac{k_{s,k}\, c_k^{\beta_k}}{1+\eta_k\, c_k^{\beta_k}} - s_k^k \right] - (\mu_{s,k} + \mu'_{s,k}) s_k^k + (1-f)\gamma_{s,k} \qquad k\ \varepsilon\ (1,n_s) \qquad (3.9)$$

where ω_k is the first-order rate constant for the kth solute [T^{-1}].

Substituting (3.3) through (3.9) into (3.1) and (3.2), and using the continuity equation describing the isothermal Darcian flow of water in a variably saturated porous medium, i.e.,

$$\frac{\partial \theta}{\partial t} = -\frac{\partial q_i}{\partial x_i} - S \qquad\qquad (3.10)$$

leads to the following equation

$$-\theta R_k \frac{\partial c_k}{\partial t} - q_i \frac{\partial c_k}{\partial x_i} + \frac{\partial}{\partial x_i}\left(\theta D_{ij,k} \frac{\partial c_k}{\partial x_j} \right) + F_k c_k + G_k = 0 \qquad k\,\varepsilon\,(1,n_s) \tag{3.11}$$

in which D_{ij} [L^{2}T^{-1}] is an effective dispersion coefficient tensor given by

$$\theta D_{ij,k} = \theta D^w_{ij,k} + a_v D^g_{ij,k}\, k_{g,k} \qquad k\,\varepsilon\,(1,n_s) \tag{3.12}$$

The coefficients F_k and G_k in (3.11) are defined as

$$\begin{aligned} F_k(c_k) = {} & -(\mu_{w,k} + \mu'_{w,k})\theta - (\mu_{s,k} + \mu'_{s,k})\rho f \frac{k_{s,k}\, c_k^{\beta_k \text{-} 1}}{1+\eta_k c_k^{\beta_k}} - (\mu_{g,k} + \mu'_{g,k}) a_v k_{g,k} + \\ & + S + k_{g,k} \frac{\partial \theta}{\partial t} - a_v \frac{\partial k_{g,k}}{\partial t} - \omega_k \rho \frac{(1 - f) k_{s,k}\, c_k^{\beta_k \text{-} 1}}{1+\eta_k c_k^{\beta_k}} - g_k(c_k) \qquad k\,\varepsilon\,(1,n_s) \end{aligned} \tag{3.13}$$

$$\begin{aligned} G_1(c_1) = {} & \gamma_{w,1}\theta + \gamma_{s,1} f \rho + \gamma_{g,1} a_v - S c_{r,1} + \omega_1 \rho s_1^k \\ G_k(c_k) = {} & (\mu'_{w,k-1}\theta + \mu'_{s,k-1} f \rho \frac{k_{s,k\text{-}1}\, c_{k\text{-}1}^{\beta_{k\text{-}1}\text{-}1}}{1+\eta_{k\text{-}1} c_{k\text{-}1}^{\beta_{k\text{-}1}}} + \mu'_{g,k-1} a_v k_{g,k\text{-}1}) c_{k\text{-}1} + \\ & + \mu'_{s,k-1} \rho s_{k\text{-}1}^k + \gamma_{w,k}\theta + \gamma_{s,k} f \rho + \gamma_{g,k} a_v - S c_{r,k} + \omega_k \rho s_k^k \qquad k\,\varepsilon\,(2,n_s) \end{aligned} \tag{3.14}$$

where the variable g_k accounts for possible changes in the adsorption parameters caused by temperature changes in the system as follows (see also section 3.4):

$$g_k(c_k) = \rho f \left(\frac{c_k^{\beta_k\text{-}1}}{1+\eta_k c_k^{\beta_k}} \frac{\partial k_{s,k}}{\partial t} - \frac{k_{s,k}\, c_k^{2\beta_k\text{-}1}}{(1+\eta_k c_k^{\beta_k})^2} \frac{\partial \eta_k}{\partial t} + \frac{k_{s,k} \ln c_k \;\, c_k^{\beta_k\text{-}1}}{(1+\eta_k c_k^{\beta_k})^2} \frac{\partial \beta_k}{\partial t} \right) \qquad k\,\varepsilon\,(1,n_s) \tag{3.15}$$

The retardation factor R_k [-] in (3.11) is given by

$$R_k(c_k) = 1 + \frac{\rho}{\theta} \frac{f\, k_{s,k}\, \beta_k\, c_k^{\beta_k\text{-}1}}{(1+\eta_k c_k^{\beta_k})^2} + \frac{a_v k_{g,k}}{\theta} \qquad k\,\varepsilon\,(1,n_s) \tag{3.16}$$

3.1.2. *Physical Nonequilibrium*

The concept of two-region, dual-porosity type solute transport [*van Genuchten and Wierenga*, 1976] is implemented in HYDRUS2 to permit consideration of physical nonequilibrium transport. The two-region concept assumes that the liquid phase can be partitioned into mobile (flowing), θ_m [L^3L^{-3}], and immobile (stagnant), θ_{im} [L^3L^{-3}], regions:

$$\theta = \theta_m + \theta_{im} \tag{3.17}$$

and that solute exchange between the two liquid regions can be modeled as a first-order process, i.e.,

$$\begin{aligned}\left[\theta_{im} + \rho(1-f)\frac{k_{s,k}\,\beta_k\,c_{k,im}^{\beta_k-1}}{(1+\eta_k\,c_{k,im}^{\beta_k})^2}\right]\frac{\partial c_{k,im}}{\partial t} &= \omega_{w,k}(c_k - c_{k,im}) + \gamma_{w,k}\theta_{im} + (1-f)\rho\,\gamma_{s,k} - \\ &- \left[\theta_{im}(\mu_{w,k} + \mu'_{w,k}) + \rho(\mu_{s,k} + \mu'_{s,k})(1-f)\frac{k_{s,k}\,c_{k,im}^{\beta_k-1}}{1+\eta_k\,c_{k,im}^{\beta_k}}\right]c_{k,im} \qquad k\,\varepsilon\,(1,n_s)\end{aligned} \tag{3.18}$$

where c_{im} [ML^{-3}] is the concentration in the immobile region and ω_k is the mass transfer coefficient for the *k*th solute [T^{-1}].

Substituting (3.17) and (3.18) into (3.1) and (3.2), the latter two equations modified for mobile and immobile regions as shown by *van Genuchten and Wagenet* [1989] for simplified two-dimensional conditions, leads to equation (3.11) in which θ is replaced with θ_m and with the coefficients F_k and G_k redefined as follows

$$F_k(c_k) = -(\mu_{w,k} + \mu'_{w,k})\theta_m - (\mu_{s,k} + \mu'_{s,k})\rho f\frac{k_{s,k}\,c_k^{\beta_k-1}}{1+\eta_k\,c_k^{\beta_k}} - (\mu_{g,k} + \mu'_{g,k})a_v\,k_{g,k} - \omega_k \tag{3.19}$$

$$
\begin{aligned}
G_1(c_1) &= \gamma_{w,1}\theta_m + \gamma_{s,1} f\rho + \gamma_{g,1} a_v - S c_{r,1} + \omega_1 c_{1,im} - \rho f g_1(c_1) \\
G_k(c_k) &= [\mu'_{w,k-1}\theta_m + \mu'_{s,k-1} f\rho \frac{k_{s,k-1} c_{k-1}^{\beta_{k-1}-1}}{1+\eta_{k-1} c_{k-1}^{\beta_{k-1}}} + \mu'_{g,k-1} a_v k_{g,k-1}] c_{k-1} + \\
&+[\mu'_{w,k-1}\theta_{im} + \mu'_{s,k-1}(1-f)\rho \frac{k_{s,k-1} c_{k-1,im}^{\beta_{k-1}-1}}{1+\eta_{k-1} c_{k-1,im}^{\beta_{k-1}}}] c_{k-1,im} + \\
&+\gamma_{w,k}\theta_m + \gamma_{s,k} f\rho + \gamma_{g,k} a_v - S c_{r,k} + \omega_k c_{k,im} - \rho f g_k(c_k) \qquad k\,\varepsilon\,(2,n_s)
\end{aligned}
\tag{3.20}
$$

In order to solve equation (3.11), it is necessary to know the water content θ and the volumetric flux q. Both variables are obtained from solutions of the Richards' equation. The above equations may appear to be relatively complicated. However, by selecting proper values of particular coefficients (i.e., γ_w , γ_s , γ_g, μ_w, μ_s, μ_g, μ_w', μ_s', μ_g', η, k_s, k_g, f, β, ω) the entire system can be simplified significantly. Assuming for example that μ_w', μ_s', μ_g', η, and k_g are zero, and f and β are equal to one, the entire system of equations (3.1) through (3.20) simplifies into a set of equations describing the transport of mutually independent solutes, i.e., single-ion transport as applicable to:

$$
\theta R \frac{\partial c}{\partial t} = \frac{\partial}{\partial x_i}\left(\theta D_{ij}^{w} \frac{\partial c}{\partial x_j}\right) - \frac{\partial q_i c}{\partial x_i} + Fc + G = 0 \tag{3.21}
$$

3.2. *Initial and Boundary Conditions*

The solution of (3.11) requires knowledge of the initial concentration within the flow region, Ω, i.e.,

$$
\begin{aligned}
c(x,z,0) &= c_i(x,z) \\
s^k(x,z,0) &= s_i^k(x,z) \\
c_{im}(x,z,0) &= c_{im,i}(x,z)
\end{aligned}
\tag{3.22}
$$

where c_i [ML^{-3}], $c_{im,i}$ [ML^{-3}] and s_i^k [-] are prescribed functions of x and z. The initial condition for s_i^k must be specified only when nonequilibrium adsorption is considered. The subscript k is dropped in (3.22) and throughout the remainder of this report, thus assuming that the transport-related equations in the theoretical development and numerical solution apply to each of the solutes in the decay chain.

Two types of boundary conditions (Dirichlet and Cauchy type conditions) can be specified along the boundary of Ω. First-type (or Dirichlet type) boundary conditions prescribe the concentration along a boundary segment Γ_D:

$$c(x, z, t) = c_0(x, z, t) \qquad \text{for } (x, z) \ \varepsilon \ \Gamma_D \tag{3.23}$$

whereas third-type (Cauchy type) boundary conditions may be used to prescribe the concentration flux along a boundary segment Γ_C as follows:

$$-\theta D_{ij}\frac{\partial c}{\partial x_j}n_i + q_i n_i c = q_i n_i c_0 \qquad \text{for } (x, z) \ \varepsilon \ \Gamma_C \tag{3.24}$$

in which $q_i\, n_i$ represents the outward fluid flux, n_i is the outward unit normal vector and c_0 is the concentration of the incoming fluid [ML^{-3}]. In some cases, for example when Γ_C is an impermeable boundary ($q_i n_i$=0) or when water flow is directed out of the region, (3.24) reduces to a second-type (Neumann type) boundary condition of the form:

$$\theta D_{ij}\frac{\partial c}{\partial x_j}n_i = 0 \qquad \text{for } (x, z) \ \varepsilon \ \Gamma_N \tag{3.25}$$

A different type of boundary condition is needed for volatile solutes when they are present in both the liquid and gas phases. This situation requires a third-type boundary condition which has on the right-hand side an additional term to account for gas diffusion through a stagnant boundary layer of thickness d [L] on the soil surface. The additional solute flux is proportional to the difference in gas concentrations above and below this boundary layer [*Jury et al.*, 1983]. This modified boundary condition has the form

$$-\theta D_{ij}\frac{\partial c}{\partial x_j}n_i + q_i n_i c = q_i n_i c_0 + \frac{D_g}{d}(k_g c - g_{atm}) \qquad \text{for } (x, z) \ \varepsilon \ \Gamma_C \tag{3.26}$$

where D_g is the molecular diffusion coefficient in the gas phase [L^{2}T^{-1}] and g_{atm} is the gas concentration above the stagnant boundary layer [ML^{-3}]. We note that *Jury et al.* [1983] assumed

g_{atm} to be zero. Similarly as for (3.24), (3.26) reduces to a second-type (Neumann type) boundary condition when water flow is zero or directed out of the region:

$$-\theta D_{ij}\frac{\partial c}{\partial x_j}n_i = \frac{D_g}{d}(k_g c - g_{atm}) \qquad \text{for } (x, z)\ \varepsilon\ \Gamma_N \qquad (3.27)$$

Equations (3.26) and (3.27) can only be used when the additional gas diffusion flux is positive. *Jury et al.* [1983] discussed how to estimate the thickness of the boundary layer, *d*, and recommended *d*=0.5 cm as a good average value for a bare surface.

3.3. *Effective Dispersion Coefficient*

The components of the dispersion tensor in the liquid phase, D_{ij}^{w}, are given by [*Bear*, 1972]

$$\theta D_{ij}^{w} = D_T|q|\delta_{ij} + (D_L - D_T)\frac{q_j q_i}{|q|} + \theta D_w \tau_w \delta_{ij} \qquad (3.28)$$

where D_w is the molecular diffusion coefficient in free water [L^2T^{-1}], τ_w is a tortuosity factor in the liquid phase [-],.q is the absolute value of the Darcian fluid flux density [LT^{-1}], δ_{ij} is the Kronecker delta function (δ_{ij}=1 if i=j, and δ_{ij}=0 if $i\neq j$), and D_L and D_T are the longitudinal and transverse dispersivities, respectively [L]. After adding the diffusion contribution from the gas phase, the individual components of the effective dispersion tensor in the soil matrix for two-dimensional transport are as follows:

$$\begin{aligned}\theta D_{xx} &= D_L\frac{q_x^2}{|q|} + D_T\frac{q_z^2}{|q|} + \theta D_w\tau_w + a_v D_g k_g \tau_g\\ \theta D_{zz} &= D_L\frac{q_z^2}{|q|} + D_T\frac{q_x^2}{|q|} + \theta D_w\tau_w + a_v D_g k_g \tau_g \qquad (3.29)\\ \theta D_{xz} &= (D_L - D_T)\frac{q_x q_z}{|q|}\end{aligned}$$

where D_g is the molecular diffusion coefficient in the gas phase [L^2T^{-1}] and τ_g is a tortuosity factor in the gas phase [-].

The tortuosity factors for both phases are evaluated in HYDRUS2 as a function of the water and gas contents using the relationship of *Millington and Quirk* [1961]:

$$\tau_w = \frac{\theta^{7/3}}{\theta_s^2} \qquad \tau_g = \frac{a_v^{7/3}}{\theta_s^2} \tag{3.30}$$

3.4. *Temperature Dependence of Transport and Reaction Coefficients*

Several of the diffusion (D_w, D_g), zero-order production (γ_w, γ_s, γ_g), first-order degradation (μ_w', μ_s', μ_g', μ_w, μ_s, and μ_g), and adsorption (k_s, k_g, β, η, ω) coefficients may be strongly dependent upon temperature. HYDRUS2 assumes that this dependency can be expressed by the Arrhenius equation [*Stumm and Morgan*, 1981]. After some modification, this equation can be expressed in the general form [*Šimůnek and Suarez,* 1993a]

$$a_T = a_r \exp\left[\frac{E_a(T^A - T_r^A)}{R_u T^A T_r^A}\right] \tag{3.31}$$

where a_r and a_T are the values of the coefficient being considered at a reference absolute temperature T_r^A and absolute temperature T^A, respectively; R_u is the universal gas constant, and E_a [$ML^2T^{-2}M^{-1}$] is the activation energy of the particular reaction or process being modeled.

4. HEAT TRANSPORT

4.1. *Governing Heat Transport Equations*

Neglecting the effects of water vapor diffusion, two-dimensional heat transport can be described as [*Sophocleous*, 1979]:

$$C(\theta)\frac{\partial T}{\partial t} = \frac{\partial}{\partial x_i}\left(\lambda_{ij}(\theta)\frac{\partial T}{\partial x_j}\right) - C_w q_i \frac{\partial T}{\partial x_i} \tag{4.1}$$

where $\lambda_{ij}(\theta)$ is the apparent thermal conductivity of the soil [MLT^{-3}K^{-1}] (e.g. Wm^{-1}K^{-1}) and $C(\theta)$ and C_w are the volumetric heat capacities [ML^{-1}T^{-2}K^{-1}] (e.g. Jm^{-3}K^{-1}) of the porous medium and the liquid phase, respectively. Volumetric heat capacity is defined as the product of the bulk density and gravimetric heat capacity. The first term on the right-hand side of (4.1) represents heat flow due to conduction and the second term accounts for heat being transported by flowing water. We do not consider the transfer of latent heat by vapor movement.According to *de Vries* [1963] the volumetric heat capacity can be expressed as

$$C(\theta) = C_n\theta_n + C_o\theta_o + C_w\theta + C_g a_v \approx (1.92\,\theta_n + 2.51\,\theta_o + 4.18\theta)10^6 \;\; [\mathrm{J\,m^{-3}K^{-1}}] \tag{4.2}$$

where θ refers to a volumetric fraction [L^{3}L^{-3}], and subscripts *n, o, g, w* represent solid phase, organic matter, gas phase and liquid phase, respectively.

4.2. *Apparent Thermal Conductivity Coefficient*

The apparent thermal conductivity, $\lambda_{ij}(\theta)$, combines the thermal conductivity $\lambda_0(\theta)$ of the porous medium (solid plus water) in the absence of flow, and the macrodispersivity which is assumed to be a linear function of the velocity [*de Marsily*, 1986]. In analogy with the dispersion coefficient for solute transport, the apparent thermal conductivity $\lambda_{ij}(\theta)$ is given by [*Šimůnek and Suarez*, 1993b]

$$\lambda_{ij}(\theta) = \lambda_T C_w |q| \delta_{ij} + (\lambda_L - \lambda_T) C_w \frac{q_j q_i}{|q|} + \lambda_0(\theta) \delta_{ij} \qquad (4.3)$$

where $|q|$ is the absolute value of the Darcian fluid flux density [LT^{-1}], δ_{ij} is the Kronecker delta function as before, and λ_L and λ_T are the longitudinal and transverse thermal dispersivities [L], respectively. The individual components of the thermal conductivity tensor for two-dimensional transport are as follows:

$$\begin{aligned} \lambda_{xx} &= \lambda_L C_w \frac{q_x^2}{|q|} + \lambda_T C_w \frac{q_z^2}{|q|} + \lambda_0 \\ \lambda_{zz} &= \lambda_L C_w \frac{q_z^2}{|q|} + \lambda_T C_w \frac{q_x^2}{|q|} + \lambda_0 \\ \lambda_{xz} &= (\lambda_L - \lambda_T) C_w \frac{q_x q_z}{|q|} \end{aligned} \qquad (4.4)$$

The volumetric heat capacity of the liquid phase is included here in the definition of the thermal conductivity in order to have the dimensions of the thermal dispersivities in the length units [*de Marsily*, 1986]. The thermal conductivity, $\lambda_0(\theta)$, accounts for the tortuosity of the porous medium, and is described with the simple equation [*Chung and Horton*, 1987]

$$\lambda_0(\theta) = b_1 + b_2 \theta_w + b_3 \theta_w^{0.5} \qquad (4.5)$$

where b_1, b_2 and b_3 are empirical parameters [MLT^{-3}K^{-1}] (e.g. Wm^{-1}K^{-1}).

4.3. *Initial and Boundary Conditions*

Equation (4.1) will be solved subject to the general initial condition

$$T(x, z, 0) = T_i(x, z) \qquad (4.6)$$

where T_i is a prescribed function of x and z.

Two types of boundary conditions (Dirichlet and Cauchy type conditions) can again be specified along the boundary of Ω. First-type (or Dirichlet type) boundary conditions prescribe the temperature along a boundary segment Γ_D:

$$T(x,z,t) = T_0(x,z,t) \qquad \text{for } (x,z)\ \varepsilon\ \Gamma_D \tag{4.7}$$

whereas third-type (Cauchy type) boundary conditions prescribe the heat flux along a boundary segment Γ_C as follows

$$-\lambda_{ij}\frac{\partial T}{\partial x_j}n_i + TC_w q_i n_i = T_0 C_w q_i n_i \qquad \text{for } (x,z)\ \varepsilon\ \Gamma_C \tag{4.8}$$

in which $q_i\, n_i$ represents the outward fluid flux, n_i is the outward unit normal vector and T_0 is the temperature of the incoming fluid. When Γ_C is an impermeable boundary ($q_i n_i$=0) or when water flow is directed out of the region, (4.8) reduces to a second-type (Neumann type) boundary condition of the form:

$$\lambda_{ij}\frac{\partial T}{\partial x_j}n_i = 0 \qquad \text{for } (x,z)\ \varepsilon\ \Gamma_N \tag{4.9}$$

The atmospheric boundary condition for soil temperature is assumed to be given by a sine function as follows [*Kirkham and Powers*, 1972]:

$$T_0 = \overline{T} + A\sin\left(\frac{2\pi t^*}{t_p} - \frac{7\pi}{12}\right) \tag{4.10}$$

where t_p is the period of time [T] necessary to complete one cycle of the sine wave (taken to be 1 day), $\overline{T}$ is the average temperature at the soil surface [K] during period t_p, A is the amplitude of the sine wave [K], and t^* is the local time [T] within the period t_p. The second term within the argument of the sine function is included to allow the highest temperature to occur at 1 p.m.

5. NUMERICAL SOLUTION OF THE WATER FLOW EQUATION

The Galerkin finite element method with linear basis functions is used to obtain a solution of the flow equation (2.1) subject to the imposed initial and boundary conditions. Since the Galerkin method is relatively standard and has been covered in detail elsewhere [*Neuman*, 1975; *Zienkiewicz,* 1977; *Pinder and Gray*, 1977], only the most pertinent steps in the solution process are given here.

5.1. *Space Discretization*

The flow region is divided into a network of triangular elements. The corners of these elements are taken to be the nodal points. The dependent variable, the pressure head function $h(x,z,t)$, is approximated by a function $h'(x,z,t)$ as follows

$$h'(x,z,t) = \sum_{n=1}^{N} \phi_n(x,z)\ h_n(t) \tag{5.1}$$

where ϕ_n are piecewise linear basis functions satisfying the condition $\phi_n(x_m, z_m)=\delta_{nm}$, h_n are unknown coefficients representing the solution of (2.1) at the nodal points, and N is the total number of nodal points.

The Galerkin method postulates that the differential operator associated with the Richards' equation (2.1) is orthogonal to each of the N basis functions, i.e.,

$$\int_{\Omega} \left\{ \frac{\partial \theta}{\partial t} - \frac{\partial}{\partial x_i} \left[K \left(K_{ij}^{A} \frac{\partial h}{\partial x_j} + K_{iz}^{A} \right) \right] + S \right\} \phi_n \, d\Omega = 0 \tag{5.2}$$

Applying Green's first identity to (5.2), and replacing h by hN, leads to

$$\sum_e \int_{\Omega_e} \left(\frac{\partial \theta}{\partial t} \phi_n + K K_{ij}^A \frac{\partial h'}{\partial x_j} \frac{\partial \phi_n}{\partial x_i} \right) d\Omega =$$

$$\sum_e \int_{\Gamma_e} K \left(K_{ij}^A \frac{\partial h'}{\partial x_j} + K_{iz}^A \right) n_i \phi_n d\Gamma + \sum_e \int_{\Omega_e} \left(-K K_{iz}^A \frac{\partial \phi_n}{\partial x_i} - S \phi_n \right) d\Omega \tag{5.3}$$

where *Ωe* represents the domain occupied by element *e*, and *Γe* is a boundary segment of element *e*. Natural flux-type (Neumann) and gradient type boundary conditions can be immediately incorporated into the numerical scheme by specifying the line integral in equation (5.3).

After imposing additional simplifying assumptions to be discussed later, and performing integration over the elements, the procedure leads to a system of time- dependent ordinary differential equations with nonlinear coefficients. In matrix form, these equations are given by

$$[F] \frac{d\{\theta\}}{dt} + [A]\{h\} = \{Q\} - \{B\} - \{D\} \tag{5.4}$$

where

$$A_{nm} = \sum_e K_l K_{ij}^A \int_{\Omega_e} \phi_l \frac{\partial \phi_n}{\partial x_i} \frac{\partial \phi_m}{\partial x_j} d\Omega =$$

$$= \sum_e \frac{\kappa}{4 A_e} \overline{K} \left[K_{xx}^A b_m b_n + K_{xz}^A (c_m b_n + b_m c_n) + K_{zz}^A c_n c_m \right] \tag{5.5}$$

$$B_n = \sum_e K_l K_{iz}^A \int_{\Omega_e} \phi_l \frac{\partial \phi_n}{\partial x_i} d\Omega = \sum_e \frac{\kappa}{2} \overline{K} \left(K_{xz}^A b_n + K_{zz}^A c_n \right) \tag{5.6}$$

$$F_{nm} = \delta_{nm} \sum_e \int_{\Omega_e} \phi_n d\Omega = \delta_{nm} \sum_e \frac{\kappa}{3} A_e \tag{5.7}$$

$$Q_n = -\sum_e \sigma_{1l} \int_{\Gamma_e} \phi_l \phi_n d\Gamma = -\sum_e \sigma_{1_n} \lambda_n \tag{5.8}$$

$$D_n = \sum_e S_l \int_{\Omega_e} \phi_l \phi_n d\Omega = \sum_e \frac{\kappa}{12} A_e (3\overline{S} + S_n) \tag{5.9}$$

where the overlined variables represent average values over an element *e*, the subscripts *i* and *j* are space direction indices (*i,j* = 1,2), and

$$l = 1,2,...,N \qquad m = 1,2,...,N \qquad n = 1,2,...,N$$

$$\begin{aligned} b_i &= z_j - z_k \qquad & c_i &= x_k - x_j \\ b_j &= z_k - z_i \qquad & c_j &= x_i - x_k \\ b_k &= z_i - z_j \qquad & c_k &= x_j - x_i \end{aligned} \tag{5.10}$$

$$A_e = \frac{c_k b_j - c_j b_k}{2} \qquad \overline{K} = \frac{K_i + K_j + K_k}{3} \qquad \overline{S} = \frac{S_i + S_j + S_k}{3}$$

Equation (5.8) is valid for a flux-type boundary condition. For a gradient-type boundary condition the variable σ_1 in (5.8) must be replaced by the product of the hydraulic conductivity *K* and the prescribed gradient σ_2 (=1). Equations (5.5) through (5.9) hold for flow in a two-dimensional Cartesian (*x,z*) domain, as well as for flow in an axisymmetric (*x,z*) system in which *x* is used as the radial coordinate. For plane flow we have

$$\kappa = 1 \qquad \lambda_n = \frac{L_n}{2} \tag{5.11}$$

while for axisymmetric flow

$$\kappa = 2\pi \frac{x_i + x_j + x_k}{3} \qquad \lambda_n = L_n \pi \frac{x'_n + 2x_n}{3} \tag{5.12}$$

The subscripts *i*, *j* and *k* in equations (5.10) and (5.12) represent the three corners of a triangular element *e*. A_e is the area of element *e*, $\overline{K}$ and $\overline{S}$ are the average hydraulic conductivity and root water extraction values over element *e*, L_n is the length of the boundary segment connected to node *n*, and x'_n is the *x*-coordinate of a boundary node adjacent to node *n*. The symbol σ_n in equation (5.8) stands for the flux [LT^{-1}] across the boundary in the vicinity of boundary node *n* (positive when directed outward of the system). The boundary flux is assumed to be uniform over each boundary segment. The entries of the vector Q_n are zero at all internal nodes, which do not act as sources or sinks for water.

The numerical procedure leading to (5.4) incorporates two important assumptions in addition to those related to the Galerkin finite element approach. One assumption concerns the time derivatives of the nodal values of the water content in (5.4). These time derivatives were weighted according to

$$\frac{d\theta_n}{dt} = \frac{\sum_e \int_{\Omega_e} \frac{\partial \theta}{\partial t} \phi_n d\Omega}{\sum_e \int_{\Omega_e} \phi_n d\Omega} \tag{5.13}$$

This assumption implements mass-lumping, which has been shown to improve the rate of convergence of the iterative solution process [e.g., *Neuman*, 1973].

A second assumption in the numerical scheme is related to the anisotropy tensor $\mathbf{K}^A$ which is taken to be constant over each element. By contrast, the water content θ, the hydraulic conductivity *K*, the soil water capacity *C*, and the root water extraction rate *S,* at a given point in time are assumed to vary linearly over each element, *e*. For example, the water content is expanded over each element as follows:

$$\theta(x,z) = \sum_{n=1}^{3} \theta(x_n, z_n) \phi_n(x,z) \qquad \text{for } (x,z) \ \varepsilon \ \Omega_e \tag{5.14}$$

where *n* stands for the corners of element *e*. The advantage of linear interpolation is that no numerical integration is needed to evaluate the coefficients in (5.4).

5.2. *Time Discretization*

Integration of (5.4) in time is achieved by discretizing the time domain into a sequence of finite intervals and replacing the time derivatives by finite differences. An implicit (backward) finite difference scheme is used for both saturated and unsaturated conditions:

$$[F]\frac{\{\theta\}_{j+1}-\{\theta\}_{j}}{\Delta t_{j}}+[A]_{j+1}\{h\}_{j+1}=\{Q\}_{j}-\{B\}_{j+1}-\{D\}_{j} \tag{5.15}$$

where j+1 denotes the current time level at which the solution is being considered, j refers to the previous time level, and $\Delta t_j=t_{j+1}-t_j$. Equation (5.15) represents the final set of algebraic equations to be solved. Since the coefficients θ, A, B, D, and Q (Q for only gradient-type boundary conditions) are functions of h, the set of equations is generally highly nonlinear. Note that the vectors D and Q are evaluated at the old time level.

5.3. *Numerical Solution Strategy*

5.3.1. *Iterative Process*

Because of the nonlinear nature of (5.15), an iterative process must be used to obtain solutions of the global matrix equation at each new time step. For each iteration a system of linearized algebraic equations is first derived from (5.15), which, after incorporation of the boundary conditions, is solved using either Gaussian elimination or the conjugate gradient method (see Section 8.5). The Gaussian elimination process takes advantage of the banded and symmetric features of the coefficient matrices in (5.15). After inversion, the coefficients in (5.15) are re-evaluated using the first solution, and the new equations are again solved. The iterative process continues until a satisfactory degree of convergence is obtained, i.e., until at all nodes in the saturated (or unsaturated) region the absolute change in pressure head (or water content) between two successive iterations becomes less than some small value determined by the imposed absolute pressure head (or water content) tolerance [*Šimůnek and van Genuchten*, 1994]. The first estimate (at zero iteration) of the unknown pressure heads at each time step is obtained by extrapolation from the pressure head values at the previous two time levels.

5.3.2. *Treatment of the Water Capacity Term*

The iteration process is extremely sensitive to the method used for evaluating the water content term ($\Delta\theta/\Delta t$) in equation (5.15). The present version of HYDRUS2 code uses the "mass-conservative" method proposed by *Celia et al.* [1990]. Their method has been shown to provide excellent results in terms of minimizing the mass balance error. The mass-conservative method proceeds by separating the water content term into two parts:

$$[F]\frac{\{\theta\}_{j+1}-\{\theta\}_j}{\Delta t_j}=[F]\frac{\{\theta\}_{j+1}^{k+1}-\{\theta\}_{j+1}^{k}}{\Delta t_j}+[F]\frac{\{\theta\}_{j+1}^{k}-\{\theta\}_j}{\Delta t_j} \tag{5.16}$$

where *k*+1 and *k* denote the current and previous iteration levels, respectively; and *j*+1 and *j* the current and previous time levels, respectively. Notice that the second term on the right hand side of (5.16) is known prior to the current iteration. The first term on the right hand side can be expressed in terms of the pressure head, so that (5.16) becomes

$$[F]\frac{\{\theta\}_{j+1}-\{\theta\}_j}{\Delta t_j}=[F][C]_{j+1}\frac{\{h\}_{j+1}^{k+1}-\{h\}_{j+1}^{k}}{\Delta t_j}+[F]\frac{\{\theta\}_{j+1}^{k}-\{\theta\}_j}{\Delta t_j} \tag{5.17}$$

where $C_{nm}=\delta_{nm}C_n$, in which C_n represents the nodal value of the soil water capacity. The first term on the right hand side of (5.17) should vanish at the end of the iteration process if the numerical solution converges. This particular feature guarantees relatively small mass balance errors in the solution.

5.3.3. *Time Control*

Three different time discretizations are introduced in HYDRUS2: (1) time discretizations associated with the numerical solution, (2) time discretizations associated with the implementation of boundary conditions, and (3) time discretizations which provide printed output of the simulation results (e.g., nodal values of dependent variables, water and solute mass balance components, and other information about the flow regime).

Discretizations 2 and 3 are mutually independent; they generally involve variable time steps as described in the input data file. Discretization 1 starts with a prescribed initial time increment, *Δt*. This time increment is automatically adjusted at each time level according to the following rules [*Mls*, 1982; *Vogel*, 1987]:

a. Discretization 1 must coincide with time values resulting from discretizations 2 and 3.
b. Time increments cannot become less than a preselected minimum time step, Δt_{min}, nor exceed a maximum time step, Δt_{max} (i.e., $\Delta t_{min} \leq \Delta t \leq \Delta t_{max}$).
c. If, during a particular time step, the number of iterations necessary to reach convergence is ≤ 3, the time increment for the next time step is increased by multiplying Δt by a predetermined constant >1 (usually between 1.1 and 1.5). If the number of iterations is ≥ 7, Δt for the next time level is multiplied by a constant <1 (usually between 0.3 and 0.9).
d. If, during a particular time step, the number of iterations at any time level becomes greater than a prescribed maximum (usually between 10 and 50), the iterative process for that time level is terminated. The time step is subsequently reset to $\Delta t/3$, and the iterative process restarted.

The selection of optimal time steps, Δt, is also influenced by the solution scheme for solute transport (see Section 6.4.6).

5.3.4. *Treatment of Pressure Head Boundary Conditions*

Finite element equations corresponding to Dirichlet nodes where the pressure head is prescribed can, at least in principle, be eliminated from the global matrix equation. An alternative and numerically simpler approach is to replace the Dirichlet finite element equations by dummy expressions of the form [*Neuman*, 1974]

$$\delta_{nm} h_m = \psi_n \tag{5.18}$$

where δ_{nm} is the Kronecker delta and ψ_n is the prescribed value of the pressure head at node *n*. The values of h_n in all other equations are set equal to ψ_n and the appropriate entries containing ψ_n in the left hand side matrix are incorporated into the known vector on the right-hand side of the global matrix equation. When done properly, this rearrangement will preserve symmetry in the matrix equation. This procedure is applied only when Gaussian elimination is used to solve the matrix equations. When the conjugate gradient solver is used, then the finite element equation representing the Dirichlet node is modified as follows. The right hand side of this equation is set equal to the prescribed pressure head multiplied by a large number (10^{30}), and entry on the left hand side representing the Dirichlet node is set equal to this large number. After solving for all pressure heads, the value of the flux Q_n can be calculated explicitly and accurately from the original finite element equation associated with node *n* [e.g., *Lynch*, 1984].

5.3.5. *Flux and Gradient Boundary Conditions*

The values of the fluxes Q_n at nodal points along prescribed flux and gradient boundaries are computed according to equation (5.8). Internal nodes which act as Neumann type sources or sinks have values of Q_n equal to the imposed fluid injection or extraction rate.

5.3.6. *Atmospheric Boundary Conditions and Seepage Faces*

Atmospheric boundaries are simulated by applying either prescribed head or prescribed flux boundary conditions depending upon whether equation (2.47) or (2.48) is satisfied [*Neuman*, 1974]. If (2.48) is not satisfied, node *n* becomes a prescribed head boundary. If, at any point in time during the computations, the calculated flux exceeds the specified potential flux in (2.47), the node will be assigned a flux equal to the potential value and treated again as a prescribed flux boundary.

All nodes expected to be part of a seepage face during code execution must be identified a priori. During each iteration, the saturated part of a potential seepage face is treated as a prescribed pressure head boundary with h=0, while the unsaturated part is treated as a prescribed flux boundary with Q=0. The lengths of the two surface segments are continually adjusted [*Neuman*, 1974] during the iterative process until the calculated values of Q (equation (5.8)) along the saturated part, and the calculated values of h along the unsaturated part, are all negative, thus indicating that water is leaving the flow region through the saturated part of the surface boundary only.

5.3.7. *Tile Drains as Boundary Conditions*

The representation of tile drains as boundary conditions is based on studies by *Vimoke et al.* [1963] and *Fipps et al.* [1986]. The approach uses results of electric analog experiments conducted by *Vimoke and Taylor* [1962] who reasoned that drains can be represented by nodal points in a regular finite element mesh, provided adjustments are made in the hydraulic conductivity, K, of neighboring elements. The adjustments should correspond to changes in the electric resistance of conducting paper as follows

$$K_{drain} = K\,C_d \tag{5.19}$$

where K_{drain} is the adjusted conductivity [LT^{-1}], and C_d is the correction factor [-]. C_d is determined from the ratio of the effective radius, d_e [L], of the drain to the side length, D [L], of the square formed by finite elements surrounding the drain node [*Vimoke at al.*, 1962]:

$$C_d = \frac{Z_0^{'}}{Z_0} \approx \frac{\sqrt{\mu_0/\varepsilon_0}}{138\ \log_{10}\rho_d + 6.48 - 2.34A - 0.48B - 0.12C} \tag{5.20}$$

where Z_0*'* is the characteristic impedance of free space (≈376.7 ohms), μ_0 is the permeability of free space, ε_0 is the permittivity of free space, and Z_0 is the characteristic impedance of a transmission line analog of the drain. The coefficients in (5.20) are given by

$$\rho_d = \frac{D}{d_e} \qquad A = \frac{1+0.405\,\rho_d^{-4}}{1-0.405\,\rho_d^{-4}}$$
$$B = \frac{1+0.163\,\rho_d^{-8}}{1-0.163\,\rho_d^{-8}} \qquad C = \frac{1+0.067\,\rho_d^{-12}}{1-0.067\,\rho_d^{-12}} \tag{5.21}$$

where d_e is the effective drain diameter to be calculated from the number and size of small openings in the drain tube [*Mohammad and Skaggs*, 1984], and D is the size of the square in the finite element mesh surrounding the drain having adjusted hydraulic conductivities. The approach above assumes that the node representing a drain must be surrounded by finite elements (either triangular or quadrilateral), which form a square whose hydraulic conductivities are adjusted according to (5.19). This method of implementing drains by means of a boundary condition gives an efficient, yet relatively accurate, prediction of the hydraulic head in the immediate vicinity of the drain, as well as of the drain flow rate [*Fipps et al.*, 1986]. More recent studies have shown that the correction factor, C_d, could be further reduced by a factor of 2 [*Rogers and Fouss*, 1989] or 4 [*Tseng*, 1994, personal communication]. These two studies compared numerical simulations of the flow of ponded water into a tile drain system with an analytical solution given by *Kirkham* [1949]. Pressure head contours calculated numerically with the original correction factor C_d (5.20), as well as with the additionally reduced correction factor $C_d/4$, were compared with the analytical results in *Šimůnek et al.* [1994].

5.3.8. *Water Balance Computations*

The HYDRUS2 code performs water balance computations at prescribed times for several preselected subregions of the flow domain. The water balance information for each subregion consists of the actual volume of water, V, in that subregion, and the rate, O, of inflow or outflow to or from the subregion. V and O are given by

$$V = \sum_e \kappa A_e \frac{\theta_i + \theta_j + \theta_k}{3} \tag{5.22}$$

and

$$O = \frac{V_{new} - V_{old}}{\Delta t} \tag{5.23}$$

respectively, where θ_i, θ_j, and θ_k are water contents evaluated at the corner nodes of element e, and where V_{new} and V_{old} are volumes of water in the subregion computed at the current and previous time levels, respectively. The summation in (5.22) is taken over all elements within the subregion.

The absolute error in the mass balance is calculated as

$$\varepsilon_a^w = V_t - V_0 + L_t \int_0^t T_a dt - \int_0^t \sum_{n_\Gamma} Q_n dt \tag{5.24}$$

where V_t and V_0 are the volumes of water in the flow domain at time t and zero, respectively, as calculated with (5.22). The third term on the right-hand side represents the cumulative root water uptake amount, while the fourth term gives the cumulative flux through nodes, n_Γ, located along the boundary of the flow domain or at internal source and sink nodes.

The accuracy of the numerical solution is evaluated in terms of the relative error, ε_r^w [%], in the water mass balance as follows:

$$\varepsilon_r^w = \frac{|\varepsilon_a^w|}{\max\left(\sum_e |V_t^e - V_0^e|,\quad L_t\int_0^t T_a dt + \int_0^t \sum_{n\Gamma} |Q_n|\, dt\right)} 100 \tag{5.25}$$

where V_t^e and V_0^e are the volumes of water in element e at times t and zero, respectively. Note that HYDRUS2 does not relate the absolute error to the volume of water in the flow domain, but instead to the maximum value of two quantities. The first quantity represents the sum of the absolute changes in water content over all elements, whereas the second quantity is the sum of the absolute values of all fluxes in and out of the flow domain. This criterion is much more strict than the usual criterion involving the total volume of water in the flow domain. This is because cumulative boundary fluxes are often much smaller than the volume in the domain, especially at the beginning of the simulation.

5.3.9. *Computation of Nodal Fluxes*

Components of the Darcian flux are computed at each time level during the simulation only when the water flow and solute transport equations are solved simultaneously. When the flow equation is being solved alone, the flux components are calculated only at selected print times. The x- and z-components of the nodal fluxes are computed for each node n according to:

$$\begin{gathered}
q_x = -\frac{K_n}{N_e}\sum_{e_n}\left[\frac{\gamma_i^x h_i + \gamma_j^x h_j + \gamma_k^x h_k}{2A_e} + K_{xz}^A\right] \\
q_z = -\frac{K_n}{N_e}\sum_{e_n}\left[\frac{\gamma_i^z h_i + \gamma_j^z h_j + \gamma_k^z h_k}{2A_e} + K_{zz}^A\right] \\
\gamma_n^x = K_{xx}^A b_n + K_{xz}^A c_n \\
\gamma_n^z = K_{xz}^A b_n + K_{zz}^A c_n
\end{gathered} \tag{5.26}$$

where N_e is the number of sub-elements e_n adjacent to node n. Einstein's summation convention is not used in (5.26).

5.3.10. *Water Uptake by Plant Roots*

HYDRUS2 considers the root zone to consist of all nodes, n, for which the potential root water uptake distribution, b (see Section 2.2), is greater than zero. The root water extraction rate is assumed to vary linearly over each element; this leads to approximation (5.9) for the root water extraction term D_n in the global matrix equation. The values of the actual root extraction rate S_n in (5.9) are evaluated with (2.13). In order to speed up the calculations, the extraction rates S_n are calculated at the old time level and are not updated during the iterative solution process at a given time step. HYDRUS2 calculates the total rate of transpiration per unit soil surface length using the equation

$$T_a = \frac{1}{L_t} \sum_e \kappa A_e \overline{S} \tag{5.27}$$

in which the summation takes place over all elements within the root zone.

5.3.11. *Evaluation of the Soil Hydraulic Properties*

At the beginning of a numerical simulation, HYDRUS2 generates for each soil type in the flow domain a table of water contents, hydraulic conductivities, and specific water capacities from the specified set of hydraulic parameters. The values of θ_i, K_i and C_i in the table are evaluated at prescribed pressure heads h_i within a specified interval (h_a, h_b). The entries in the table are generated such that

$$\frac{h_{i+1}}{h_i} = \text{constant} \tag{5.28}$$

which means that the spacing between two consecutive pressure head values increases in a logarithmic fashion. Values for the hydraulic properties, $\theta(h)$, $K(h)$ and $C(h)$, are computed during the iterative solution process using linear interpolation between the entries in the table. If an argument h falls outside the prescribed interval (h_a, h_b), the hydraulic characteristics are evaluated directly from the hydraulic functions, i.e., without interpolation. The above interpolation technique was found to be much faster computationally than direct evaluation of the hydraulic functions over the entire range of pressure heads, except when very simple hydraulic models were used.

5.3.12. *Implementation of Hydraulic Conductivity Anisotropy*

Since the hydraulic conductivity anisotropy tensor, $\mathbf{K}^A$, is assumed to be symmetric, it is possible to define at any point in the flow domain a local coordinate system for which the tensor $\mathbf{K}^A$ is diagonal (i.e., having zeroes everywhere except on the diagonal). The diagonal entries K_1^A and K_2^A of $\mathbf{K}^A$ are referred to as the principal components of $\mathbf{K}^A$.

The HYDRUS2 code permits one to vary the orientation of the local principal directions from element to element. For this purpose, the local coordinate axes are subjected to a rotation such that they coincide with the principal directions of the tensor $\mathbf{K}^A$. The principal components K_1^A and K_2^A, together with the angle ω_a between the principal direction of K_1^A and the x-axis of the global coordinate system, are specified for each element. Each locally determined tensor $\mathbf{K}^A$ is transformed to the global (x,z) coordinate system at the beginning of the simulation using the following rules:

$$K_{xx}^A = K_1^A \cos^2\omega_a + K_2^A \sin^2\omega_a$$

$$K_{zz}^A = K_1^A \sin^2\omega_a + K_2^A \cos^2\omega_a \qquad (5.29)$$

$$K_{xz}^A = (K_2^A - K_1^A) \sin\omega_a \cos\omega_a$$

5.3.13. *Steady-State Analysis*

All transient flow problems are solved by time marching until a prescribed time is reached. The steady-state problem can be solved in the same way, i.e., by time marching until two successive solutions differ less than some prescribed pressure head tolerance. HYDRUS2 implements a faster way of obtaining the steady-state solution without having to go through a large number of time steps. The steady-state solution for a set of imposed boundary conditions is obtained directly during one set of iterations at the first time step by equating the time derivative term in the Richards' equation (2.1) to zero.

6. NUMERICAL SOLUTION OF THE SOLUTE TRANSPORT EQUATION

The Galerkin finite element method is also used to solve the solute and heat transport equations (equations (3.11) and (4.1), respectively) subject to appropriate initial and boundary conditions. Since the heat transport equation (4.1) has the same mathematical form as the (linearized) solute transport equation (3.11), the numerical solution will be given here only for solute transport. The solution procedure largely parallels the approach used in Section 5 for the flow equation.

6.1. *Space Discretization*

The dependent variable, the concentration function *c*(*x*,*z*,*t*), is approximated by a finite series *c'* (*x*,*z*,*t*) of the form

$$c'(x,z,t)=\sum_{n=1}^{N}\phi_n(x,z)\;c_n(t) \tag{6.1}$$

where ϕ_n are the selected linear basis functions, c_n are the unknown time dependent coefficients which represent solutions of (5.11) at the finite element nodal points and, as before, *N* is the total number of nodal points. Application of the standard Galerkin method leads to the following set of *N* equations

$$\int_{\Omega}\left[-\theta R\frac{\partial c}{\partial t}-q_i\frac{\partial c}{\partial x_i}+\frac{\partial}{\partial x_i}\left(\theta D_{ij}\frac{\partial c}{\partial x_j}\right)+Fc+G\right]\phi_n d\Omega=0 \tag{6.2}$$

Application of Green's theorem to the second derivatives in (6.2) and substitution of *c* by *c'* results in the following system of time-dependent differential equations

$$\begin{aligned}\sum_e\int_{\Omega_e}\left[\left(-\theta R\frac{\partial c'}{\partial t}-q_i\frac{\partial c'}{\partial x_i}+Fc'+G\right)\phi_n-\theta D_{ij}\frac{\partial c'}{\partial x_j}\frac{\partial \phi_n}{\partial x_i}\right]d\Omega+\\+\sum_e\int_{\Gamma_e}\theta D_{ij}\frac{\partial c'}{\partial x_j}n_i\phi_n d\Gamma=0\end{aligned} \tag{6.3}$$

or in matrix form:

$$[Q]\frac{d\{c\}}{dt}+[S]\{c\}+\{f\}=-\{Q^D\} \tag{6.4}$$

where

$$Q_{nm}=\delta_{nm}\sum_e(-\theta R)_l\int_{\Omega_e}\phi_l\phi_n d\Omega= \tag{6.5}$$

$$\begin{aligned} S_{nm}&=\sum_e\left[(-q_i)_l\int_{\Omega_e}\phi_l\phi_n\frac{\partial\phi_m}{\partial x_i}d\Omega-(\theta D_{ij})_l\int_{\Omega_e}\phi_l\frac{\partial\phi_n}{\partial x_i}\frac{\partial\phi_m}{\partial x_j}d\Omega+F_l\int_{\Omega_e}\phi_l\phi_n\phi_m d\Omega\right]= \\ &=\sum_e\{-\frac{\kappa b_m}{24}(3\overline{q}_x+q_{xn})-\frac{\kappa c_m}{24}(3\overline{q}_z+q_{zn})+\frac{\kappa A_e}{60}(3\overline{F}+F_n+F_m)(1+\delta_{nm})- \\ &\quad-\frac{\kappa}{4A_e}\left[b_m b_n\theta\overline{D_{xx}}+(b_m c_n+c_m b_n)\theta\overline{D_{xz}}+c_m c_n\theta\overline{D_{zz}}\right]\} \end{aligned} \tag{6.6}$$

$$f_n=\sum_e G_l\int_{\Omega_e}\phi_l\phi_n d\Omega=\sum_e\frac{\kappa A_e}{12}(3\overline{G}+G_n) \tag{6.7}$$

in which the overlined variables represent average values over a given element *e*. The notation in the above equations is similar as in (5.10). The boundary integral in (6.3) represents the dispersive flux, Q_n^D, across the boundary and will be discussed later in Section 6.3.4.

The derivation of equations (6.5) through (6.7) invoked several important assumptions in addition to those involved in the Galerkin finite element approach [*Huyakorn and Pinder*, 1983; *van Genuchten*, 1978]. First, the different coefficients under the integral signs (θR, q_i, θD_{ij}, F, G) were expanded linearly over each element, similarly as for the dependent variable, i.e., in terms of their nodal values and associated basis functions. Second, mass lumping was invoked by redefining the nodal values of the time derivative in (6.4) as weighted averages over the entire flow region:

$$\frac{dc_n}{dt}=\frac{\sum_e\int_{\Omega_e}\theta R\frac{\partial c'}{\partial t}\phi_n d\Omega}{\sum_e\int_{\Omega_e}\theta R\phi_n d\Omega} \tag{6.8}$$

6.2. *Time Discretization*

The Galerkin method is used only for approximating the spatial derivatives while the time derivatives are discretized by means of finite differences. A first-order approximation of the time derivatives leads to the following set of algebraic equations:

$$[Q]_{j+\varepsilon}\frac{\{c\}_{j+1}-\{c\}_j}{\Delta t}+\varepsilon[S]_{j+1}\{c\}_{j+1}+(1-\varepsilon)[S]_j\{c\}_j+\varepsilon\{f\}_{j+1}+(1-\varepsilon)\{f\}_j=0 \tag{6.9}$$

where j and j+1 denote the previous and current time levels, respectively; Δt is the time increment, and ε is a time weighing factor. The incorporation of the dispersion flux, Q_n^D, into matrix [Q] and vector {f} is discussed in Section 6.3.4. The coefficient matrix $[Q]_{j+\varepsilon}$ is evaluated using weighted averages of the current and previous nodal values of θ and R. Equation (6.9) can be rewritten in the form:

$$[G]\{c\}_{j+1}=\{g\} \tag{6.10}$$

where

$$\begin{gathered}[G]=\frac{1}{\Delta t}[Q]_{j+\varepsilon}+\varepsilon[S]_{j+1}\\ \{g\}=\frac{1}{\Delta t}[Q]_{j+\varepsilon}\{c\}_j-(1-\varepsilon)[S]_j\{c\}_j-\varepsilon\{f\}_{j+1}-(1-\varepsilon)\{f\}_j\end{gathered} \tag{6.11}$$

Higher-order approximations for the time derivative in the transport equation were derived by *van Genuchten* [1976, 1978]. The higher-order effects may be incorporated into the transport equation by introducing time-dependent dispersion corrections as follows

$$
\begin{aligned}
D_{ij}^{-} &= D_{ij} - \frac{q_i q_j \Delta t}{6\theta^2 R} \\
D_{ij}^{+} &= D_{ij} + \frac{q_i q_j \Delta t}{6\theta^2 R}
\end{aligned}
\qquad (6.12)
$$

where the superscripts + and - indicate evaluation at the old and new time levels, respectively.

6.3. *Numerical Solution for Linear Nonequilibrium Solute Transport*

The same solution procedure as described in Sections 6.1 and 6.2 is used here for either linear equilibrium or nonlinear (both equilibrium and nonequilibrium) solute transport. However, linear nonequilibrium transport is implemented somewhat differently. First, equation (3.9), simplified for linear adsorption, is discretized using finite differences as follows

$$
\begin{aligned}
\frac{s^{t+\Delta t} - s^t}{\Delta t} &= \varepsilon[\omega(1-f)k_s c - \omega s^k - \mu_s s^k + (1-f)\gamma]^{t+\Delta t} + \\
&+ (1-\varepsilon)[\omega(1-f)k_s c - \omega s^k - \mu_s s^k + (1-f)\gamma]^t
\end{aligned}
\qquad (6.13)
$$

The new adsorbed concentration for type-2 sorption sites follows directly from (6.13):

$$
s^{t+\Delta t} = s^t \frac{2 - \Delta t(\omega + \mu_s)^t}{2 + \Delta t(\omega + \mu_s)^{t+\Delta t}} + \frac{\Delta t(1-f)[(\omega k_s c)^{t+\Delta t} + (\omega k_s c)^t + \gamma^{t+\Delta t} + \gamma^t]}{2 + \Delta t(\omega + \mu_s)^{t+\Delta t}} \qquad (6.14)
$$

This term is incorporated directly into *F* and *G* so that they have the following values:

$$
F_*^{t+\Delta t} = F^{t+\Delta t} - \left\{ \frac{\rho\omega\Delta t(1-f)\omega k_s}{2 + \Delta t(\omega + \mu_s)} \right\}^{t+\Delta t} \qquad (6.15)
$$

$$
G_*^{t+\Delta t} = G^{\{'t+\Delta t\}} + \rho\omega^{t+\Delta t} \left\{ s^t \frac{2 - \Delta t(\omega + \mu_s)^t}{2 + \Delta t(\omega + \mu_s)^{t+\Delta t}} + \frac{\Delta t(1-f)[(\omega k_s c)^t + \gamma^{t+\Delta t} + \gamma^t]}{2 + \Delta t(\omega + \mu_s)^{t+\Delta t}} \right\} \qquad (6.16)
$$

where $F_*^{t+\Delta t}$ and $G_*^{t+\Delta t}$ are the values of parameters F and G for linear nonequilibrium solute transport, and $F^{t+\Delta t}$' and $G^{t+\Delta t}{}'$ are the original values of F and G. The above procedure avoids having to solve two simultaneous equations for linear nonequilibrium transport. Once the transport equation with the modified F and G parameters is solved using the methods discussed earlier to yield the concentration $c^{t+\Delta t}$, equation (6.14) is used to update the adsorbed concentration $s^{t+\Delta t}$.

For physical nonequilibrium (dual-porosity) transport, equation (3.18), simplified for linear adsorption, is discretized using finite differences as follows

$$A\frac{c_{im}^{t+\Delta t} - c_{im}^{t}}{\Delta t} = \varepsilon[\omega(c - c_{im}) - Bc_{im} + E]^{t+\Delta t} + (1 - \varepsilon)[\omega(c - c_{im}) - Bc_{im} + E]^{t}$$

where

$$\begin{aligned} A &= \theta_{im} + (1 - f)\rho k_s \\ B &= \theta_{im}\mu_w + (1 - f)\rho k_s \mu_s \\ E &= \theta_{im}\gamma_w + (1 - f)\rho\gamma_s \end{aligned} \tag{6.17}$$

The new concentration in the immobile region follows directly from (6.13):

$$c_{im}^{t+\Delta t} = c_{im}^{t}\frac{2A - \Delta t(\omega + B)^{t}}{2A + \Delta t(\omega + B)^{t+\Delta t}} + \frac{\Delta t[(\omega c_{im})^{t+\Delta t} + (\omega c_{im})^{t} + E^{t+\Delta t} + E^{t}]}{2A + \Delta t(\omega + B)^{t+\Delta t}} \tag{6.18}$$

Similarly as for the chemical nonequilibrium case, equation (6.18) is incorporated directly into F and G to obtain following values:

$$F_*^{t+\Delta t} = F^{t+\Delta t} - \left\{\frac{\omega\Delta t\omega}{2A + \Delta t(\omega + B)}\right\}^{t+\Delta t} \tag{6.19}$$

$$G_*^{t+\Delta t} = G^{\{'t+\Delta t\}} + \omega^{t+\Delta t}\left\{c_{im}^{t}\frac{2A - \Delta t(\omega + B)^{t}}{2A + \Delta t(\omega + B)^{t+\Delta t}} + \frac{\Delta t[(\omega c)^{t} + E^{t+\Delta t} + E^{t}\,]}{2A + \Delta t(\omega + B)^{t+\Delta t}}\right\} \tag{6.20}$$

6.4. *Numerical Solution Strategy*

6.4.1. *Solution Process*

The solution process at each time step proceeds as follows. First, an iterative procedure is used to obtain the solution of the Richards' equation (2.1) (see Section 5.3.1). After achieving convergence, the solution of the transport equation (6.10) is implemented. This is done by first determining the nodal values of the fluid flux from nodal values of the pressure head by applying Darcy's law. Nodal values of the water content and the fluid flux at the previous time level are already known from the solution at the previous time step. Values for the water content and the fluid flux are subsequently used as input to the transport equations (first for heat transport and then for solute transport), leading to the system of linear algebraic equations given by (6.10). The structure of the final set of equations depends upon the value of the temporal weighing factor, ε. The explicit (ε=0) and fully implicit (ε=1) schemes require that the global matrix [G] and the vector {g} be evaluated at only one time level (the previous or current time level). All other schemes require evaluation at both time levels. Also, all schemes except for the explicit formulation (ε=0) lead to an asymmetric banded matrix [G]. The associated set of algebraic equations is solved using either a standard asymmetric matrix equation solver [e.g., *Neuman*, 1972], or the ORTHOMIN method [*Mendoza et al.*, 1991], depending upon the size of final matrix. By contrast, the explicit scheme leads to a diagonal matrix [G], which is much easier to solve (but generally requires smaller time steps).

Since the heat transport equation is linear, there is no need for an iterative solution process for heat flow. The same is true for the transport of solutes undergoing only linear sorption reactions. On the other hand, iteration is needed when a nonlinear reaction between the solid and liquid phase is considered. The iteration procedure for solute transport is very similar to that for water flow. The coefficients in (6.10) are re-evaluated using each iteration, and the new equations are again solved using results of the previous iteration. The iterative process continues until a satisfactory degree of convergence is obtained, i.e., until at all nodes the absolute change in concentration between two successive iterations becomes less than some small value determined by the imposed relative and absolute concentration tolerances.

6.4.2. *Upstream Weighted Formulation*

Upstream weighing is provided as an option in the HYDRUS2 to minimize some of the problems with numerical oscillations when relatively steep concentration fronts are being simulated. For this purpose the second (flux) term of equation (6.3) is not weighted by regular linear basis functions ϕ_n, but instead using the nonlinear functions ϕ_n^u [*Yeh and Tripathi*, 1990]

$$\begin{aligned} \phi_1^u &= L_1 - 3\,\alpha_3^w\,L_2\,L_1 + 3\,\alpha_2^w\,L_3\,L_1 \\ \phi_2^u &= L_2 - 3\,\alpha_1^w\,L_3\,L_2 + 3\,\alpha_3^w\,L_1\,L_2 \\ \phi_3^u &= L_3 - 3\,\alpha_2^w\,L_1\,L_3 + 3\,\alpha_1^w\,L_2\,L_3 \end{aligned} \tag{6.21}$$

where α_i^w is a weighing factor associated with the length of the element size opposite to node *i*, and L_i are the local coordinates. The weighing factors are evaluated using the equation of *Christie et al.* [1976]:

$$\alpha_i^w = \coth\left(\frac{uL}{2D}\right) - \frac{2D}{uL} \tag{6.22}$$

where *u, D* and *L* are the flow velocity, dispersion coefficient and length associated with side *i*. The weighing functions ϕ^u ensure that relatively more weight is placed on the flow velocities of nodes located at the upstream side of an element. Evaluating the integrals in (6.3) shows that the following additional terms must be added to the entries of the global matrix S_{nm} in equation (6.6):

$$\begin{aligned} S_{1j}^{e'} = S_{1j}^{e} &- \frac{b_j}{40}[2q_{x1}(\alpha_2^w - \alpha_3^w) + q_{x2}(\alpha_2^w - 2\alpha_3^w) + q_{x3}(2\alpha_2^w - \alpha_3^w)] - \\ &- \frac{c_j}{40}[2q_{z1}(\alpha_2^w - \alpha_3^w) + q_{z2}(\alpha_2^w - 2\alpha_3^w) + q_{z3}(2\alpha_2^w - \alpha_3^w)] \end{aligned} \tag{6.23}$$

and

$$S_{2j}^{e'} = S_{2j}^{e} - \frac{b_j}{40}[q_{x1}(2\alpha_3^w - \alpha_1^w) + 2q_{x2}(\alpha_3^w - \alpha_1^w) + q_{x3}(\alpha_3^w - 2\alpha_1^w)] - \\ - \frac{c_j}{40}[q_{z1}(2\alpha_3^w - \alpha_1^w) + 2q_{z2}(\alpha_3^w - \alpha_1^w) + q_{z3}(\alpha_3^w - 2\alpha_1^w)] \tag{6.24}$$

$$S_{3j}^{e'} = S_{3j}^{e} - \frac{b_j}{40}[q_{x1}(\alpha_1^w - 2\alpha_2^w) + q_{x2}(2\alpha_1^w - \alpha_2^w) + 2q_{x3}(\alpha_1^w - \alpha_2^w)] - \\ - \frac{c_j}{40}[q_{z1}(\alpha_1^w - 2\alpha_2^w) + q_{z2}(2\alpha_1^w - \alpha_2^w) + 2q_{z3}(\alpha_1^w - \alpha_2^w)] \tag{6.25}$$

The weighing factors are applied only to those element sides that are inclined within 10 degrees from the flow direction.

6.4.3. *Implementation of First-Type Boundary Conditions*

Individual equations in the global matrix equation which correspond to nodes at which the concentration is prescribed are replaced by new equations:

$$\delta_{nm} c_m = c_{n0} \tag{6.26}$$

where c_{n0} is the prescribed value of the concentration at node *n*. This is done only when Gaussian elimination is used to solve the matrix equation. A similar procedure as for water flow (described in Section 5.3.4) is applied when the ORTHOMIN method is used. Because of asymmetry of the global matrix [*G*], no additional manipulations are needed in the resulting system of equations as was the case for the water flow solution.

The total material flux, Q_n^T, through a boundary at node *n* consists of the dispersive flux, Q_n^D, and the convective flux, Q_n^A:

$$Q_n^T = Q_n^D + Q_n^A \tag{6.27}$$

The dispersive boundary nodal flux is not known explicitly but must be calculated from equation (6.4). Hence, the dispersion flux, Q_n^D, for node *n* can be calculated as

$$Q_n^D = -[\varepsilon S_{nm}^{j+1} + (1-\varepsilon) S_{nm}^{j}] c_m^j - \varepsilon f_n^{j+1} - (1-\varepsilon) f_n^j - Q_{nn}^{j+\varepsilon} \frac{c_n^{j+1} - c_n^j}{\Delta t} \tag{6.28}$$

The convective flux is evaluated as

$$Q_n^A = Q_n c_n \tag{6.29}$$

where the fluid flux Q_n is known from the solution of the water flow equation.

6.4.4. *Implementation of Third-Type Boundary Conditions*

Equation (3.24) is rewritten as follows

$$\theta D_{ij} \frac{\partial c'}{\partial x_j} n_i = q_i n_i (c - c_0) \tag{6.30}$$

When substituted into the last term of (6.3), the boundary integral becomes

$$\sum_e \int_{\Gamma_e} \theta D_{ij} \frac{\partial c'}{\partial x_j} n_i \phi_n d\Gamma = Q_n c_n - Q_n c_{n0} \tag{6.31}$$

The first term on the right-hand side of (6.31) represents the convective flux. This term is incorporated into the coefficient matrix [*S*] of (6.4). The last term of (6.31) represents the total material flux, which is added to the known vector {*f*}.

At nodes where free outflow of water and its dissolved solutes takes place, the exit concentration c_0 is equal to the local (nodal) concentration c_n. In this case the dispersive flux becomes zero and the total material flux through the boundary is evaluated as

$$Q_n^T = Q_n c_n \tag{6.32}$$

The Cauchy boundary condition for volatile solutes is treated in a similar way. Equation (3.26) is rewritten as follows

$$\theta D_{ij}\frac{\partial c}{\partial x_j}n_i = q_i n_i(c-c_0) - \frac{D_g}{d}(k_g c - g_{atm}) \tag{6.33}$$

Again, when substituted into the last term of (6.3), the boundary integral becomes

$$\sum_e \int_{\Gamma_e} \theta D_{ij}\frac{\partial c'}{\partial x_j}n_i\phi_n d\Gamma = Q_n c_n - Q_n c_{n0} - \frac{D_g \Delta L}{d}(k_g c_n - g_{atm}) \tag{6.34}$$

where *ΔL* is the length of the boundary associated with node *n*. The last term of (6.34) representing the gas diffusion flux through the stagnant boundary layer at the soil surface is directly added to the vector {*f*} in equation (6.9), whereas the term containing k_g and unknown concentration c_n is incorporated into the coefficient matrix [*S*]. The other terms on the right-hand side of (6.34) are treated in the same way as described above for equation (6.31).

6.4.5. *Mass Balance Calculations*

The total amount of mass in the entire flow domain, or in a preselected subregion, is given by

$$M = \sum_e \int_{\Omega_e} (\theta c + a_v g + \rho s) d\Omega = \sum_e \int_{\Omega_e} \left[\left(\theta + a_v k_g + \rho f \frac{k_s c^{\beta\text{-}1}}{1+\eta c^{\beta}} \right) c + \rho s^k \right] d\,\Omega \tag{6.35}$$

The summation is taken over all elements within the specified region. The equations in this section pertain only to the equilibrium or chemical nonequilibrium models; the mass balance equations for physical nonequilibrium transport are very similar as those for chemical nonequilibirum.

The cumulative amounts M^0 and M^1 of solute removed from the flow region by zero- and first-order reactions, respectively, are calculated as follows

$$M_1^0 = -\int_0^t \sum_e \int_{\Omega_e} (\gamma_{w,1}\theta + \gamma_{s,1}\rho + \gamma_{g,1}a_v) d\Omega dt$$

$$M_k^0 = -\int_0^t \sum_e \int_{\Omega_e} [\left(\mu'_{w,k-1}\theta + \mu'_{s,k-1}\rho f \frac{k_{s,k-1} c_{k-1}^{\beta_{k-1}-1}}{1 + \eta_{k-1} c_{k-1}^{\beta_{k-1}}} + \mu'_{g,k-1} a_v k_{g,k-1} \right) c_{k-1} + \\ + \mu'_{s,k-1}\rho s_{k-1}^k + \gamma_{w,k}\theta + \gamma_{s,k}\rho + \gamma_{g,k}a_v] d\Omega dt \tag{6.36}$$

$$M^1 = \int_0^t \sum_e \int_{\Omega_e} \{ \left[(\mu_w + \mu'_w)\theta + (\mu_s + \mu'_s)\rho f \frac{k_s c^{\beta-1}}{1 + \eta c^{\beta}} + (\mu_g + \mu'_g) a_v k_g \right] c + \\ + (\mu_s + \mu'_s)\rho s^k \} d\Omega dt \tag{6.37}$$

whereas the cumulative amount M_r of solute taken up by plant roots is given by

$$M_r = \int_0^t \sum_{e_R} \int_{\Omega_e} S c_r d\Omega dt \tag{6.38}$$

where e_R represents the elements making up the root zone.

Finally, when all boundary material fluxes, decay reactions, and root uptake mass fluxes have been computed, the following mass balance should hold for the flow domain as a whole:

$$M_t - M_0 = + \int_0^t \sum_{n_\Gamma} Q_n^T dt - M^0 - M^1 - M_r \tag{6.39}$$

where M_t and M_0 are the amounts of solute in the flow region at times t and zero, respectively, as calculated with (6.35), and n_Γ represents nodes located along the boundary of the flow domain or at internal sinks and/or sources. The difference between the left- and right-hand sides of (6.39) represents the absolute error, ε_a^c, in the solute mass balance. Similarly as for water flow, the accuracy of the numerical solution for solute transport is evaluated by using the relative error, ε_r^c [%], in the solute mass balance as follows

$$\varepsilon_r^c = \frac{100|\varepsilon_a^c|}{\max\left(\sum_e |M_t^e - M_0^e|,\ |M^0| + |M^1| + |M_r| + \int_0^t \sum_{n\Gamma} |Q_n^T| dt\right)} \qquad (6.40)$$

where M_0^e and M_t^e are the amounts of solute in element e at times 0 and t, respectively. Note again that HYDRUS2 does not relate the absolute error to the total amount of mass in the flow region. Instead, the program uses as a reference the maximum value of (1) the absolute change in element concentrations as summed over all elements, and (2) the sum of the absolute values of all cumulative solute fluxes across the flow boundaries including those resulting from sources and sinks in the flow domain.

The total amount of heat energy in the entire flow domain, or in a preselected subregion, is given by

$$W = \sum_e \int_{\Omega_e} (C_n \theta_n + C_o \theta_o + C_w \theta + C_g a_v) T^A d\Omega \qquad (6.41)$$

where T^A is the absolute temperature [K]. The summation is taken over all elements within the specified region.

6.4.6. *Oscillatory Behavior*

Numerical solutions of the transport equation often exhibit oscillatory behavior and/or excessive numerical dispersion near relatively sharp concentration fronts. These problems can be especially serious for convection-dominated transport characterized by small dispersivities. One way to partially circumvent numerical oscillations is to use upstream weighing as discussed in Section 6.4.2. Undesired oscillations can often be prevented also by selecting an appropriate combination of space and time discretizations. Two dimensionless numbers may be used to characterize the space and time discretizations. One of these is the grid Peclet number, Pe_i^e, which defines the predominant type of the solute transport (notably the ratio of the convective and dispersive transport terms) in relation to coarseness of the finite element grid:

$$Pe_i^e = \frac{q_i \Delta x_i}{\theta D_{ii}} \tag{6.42}$$

where Δx_i is the characteristic length of a finite element. The Peclet number increases when the convective part of the transport equation dominates the dispersive part, i.e., when a relatively steep concentration front is present. To achieve acceptable numerical results, the spatial discretization must be kept relatively fine to maintain a low Peclet number. Numerical oscillation can be virtually eliminated when the local Peclet numbers do not exceed about 5. However, acceptably small oscillations may be obtained with local Peclet numbers as high as 10 [*Huyakorn and Pinder*, 1983]. Undesired oscillation for higher Peclet numbers can be effectively eliminated by using upstream weighing (see Section 6.4.2).

A second dimensionless number which characterizes the relative extent of numerical oscillations is the Courant number, Cr_i^e. The Courant number is associated with the time discretization as follows

$$Cr_i^e = \frac{q_i \Delta t}{\theta R \, \Delta x_i} \tag{6.43}$$

Three stabilizing options are used in HYDRUS2 to avoid oscillations in the numerical solution of the solute transport equation [*Šimůnek and van Genuchten*, 1994]. One option is upstream weighing (see Section 6.4.2), which effectively eliminates undesired oscillations at relatively high Peclet numbers. A second option for minimizing or eliminating numerical oscillations uses the criterion developed by *Perrochet and Berod* [1993]

$$Pe \bullet Cr \leq \omega_s \quad (=2) \tag{6.44}$$

where ω_s is the performance index [-]. This criterion indicates that convection-dominated transport problems having large *Pe* numbers can be safely simulated provided *Cr* is reduced according to (6.44) [*Perrochet and Berod*, 1993]. When small oscillations in the solution can be tolerated, ω_s can be increased to about 5 or 10.

A third stabilization option implemented in HYDRUS2 also utilizes criterion (6.44). However, instead of decreasing *Cr* to satisfy equation (6.44), this option introduces artificial dispersion to decrease the Peclet number. The amount of additional longitudinal dispersion, $\overline{D}_L$ [L], is given by [*Perrochet and Berod*, 1993]

$$\overline{D}_L = \frac{|q|\Delta t}{R\,\theta\,\omega_s} - D_L - \frac{\theta\,D_w\tau}{|q|} \tag{6.45}$$

The maximum permitted time step is calculated for all three options, as well as with the additional requirement that the Courant number must remain less than or equal to 1. The time step calculated in this way is subsequently used as one of the time discretization rules (rule No. B) discussed in section 5.3.3.

7. PARAMETER OPTIMIZATION

Parameter optimization is an indirect approach for estimating the unsaturated soil hydraulic and/or solute transport parameters from transient flow and/or transport data. Inverse methods are typically based upon the minimization of a suitable objective function, which expresses the discrepancy between the observed values and the predicted system response. Soil hydraulic properties for this purpose are assumed to be described by an analytical model with unknown parameter values (see Section 2.3). The system response is represented by a numerical solution of the flow equation, augmented with the parameterized hydraulic functions, selected transport parameters, and suitable initial and boundary conditions. Initial estimates of the optimized system parameters are then iteratively improved during the minimization process until a desired degree of precision is obtained. This methodology was originally applied to one-step and multi-step column outflow data generated in the laboratory [see for example *Kool et al.*, 1985; *van Dam et al.*, 1994], and laboratory or field transport data during steady-state water flow [*van Genuchten*, 1981; *Toride et al.*, 1995]. HYDRUS2 implements parameter optimization also for estimating the solute transport and reaction parameters from transient water flow and/or solute transport experiments.

7.1. *Definition of the Objective Function*

The objective function Φ to be minimized during the parameter estimation process may be defined as [*Šimůnek et al.*, 1998]:

$$\begin{aligned}\Phi(\boldsymbol{b},\boldsymbol{q},\boldsymbol{p}) = &\sum_{j=1}^{m_q} v_j \sum_{i=1}^{n_{qj}} w_{i,j}[q_j^*(\boldsymbol{x},t_i) - q_j(\boldsymbol{x},t_i,\boldsymbol{b})]^2 + \\ &+ \sum_{j=1}^{m_p} \overline{v}_j \sum_{i=1}^{n_{pj}} \overline{w}_{i,j}[p_j^*(\theta_i) - p_j(\theta_i,\boldsymbol{b})]^2 + \\ &+ \sum_{j=1}^{n_b} \hat{v}_j[b_j^* - b_j]^2\end{aligned} \qquad (7.1)$$

where the first term on the right-hand side represents deviations between the measured and calculated space-time variables (e.g., observed pressure heads, water contents, and/or concentrations at different locations and/or time in the flow domain, or the actual or cumulative flux versus time across a boundary of specified type). In this term, m_q is the number of different sets of measurements, n_{qj} is the number of measurements in a particular measurement set, $q_j^*(\boldsymbol{x},t_i)$ represents specific measurements at time t_i for the jth measurement set at location $\boldsymbol{x}(r,z)$, $q_j(\boldsymbol{x},t_i\ ,\boldsymbol{b})$ are the corresponding model predictions for the vector of optimized parameters $\boldsymbol{b}$ (e.g., θ_r, θ_s, α, n, K_s, D_l, $k_{g,k}$, ...), and v_j and $w_{i,j}$ are weights associated with a particular measurement set or point, respectively. The second term of (7.1) represents differences between independently measured and predicted soil hydraulic properties (e.g., retention, $\theta(h)$ and/or hydraulic conductivity, $K(\theta)$ or $K(h)$ data), while the terms m_p, n_{pj}, $p_j^*(\theta_i)$, $p_j(\theta_i\ ,\boldsymbol{b})$, $\overline{v_{jj}}$ and $\overline{w_{i,j}}$ have similar meanings as for the first term but now for the soil hydraulic properties. The last term of (7.1) represents a penalty function for deviations between prior knowledge of the soil hydraulic parameters, b_j^*, and their final estimates, b_j, with n_b being the number of parameters with prior knowledge and $\hat{v}_j$ representing pre-assigned weights. Estimates, which make use of prior information (such as those used in the third term of (7.1)) are known as Bayesian estimates. We note that the covariance (weighting) matrices, which provide information about the measurement accuracy, as well as any possible correlation between measurement errors and/or parameters, are assumed to be diagonal in this study. The weighting coefficients v_j , which minimize differences in weighting between different data types because of different absolute values and numbers of data involved, are given by [*Clausnitzer and Hopmans*, 1995]:

$$v_j = \frac{1}{n_j \sigma_j^2} \tag{7.2}$$

which causes the objective function to become the average weighted squared deviation normalized by the measurement variances σ_j^2.

7.2. *Marquardt-Levenberg Optimization Algorithm*

Minimization of the objective function Φ is accomplished by using the Levenberg-Marquardt nonlinear minimization method (a weighted least-squares approach based on Marquardt's maximum neighborhood method) [*Marquardt*, 1963]. This method combines the Newton and steepest descend methods, and generates confidence intervals for the optimized parameters. The method was found to be very effective and has become a standard in nonlinear least-squares fitting among soil scientists and hydrologists [*van Genuchten*, 1981; *Kool et al.*, 1985, 1987].

7.3. *Statistics of the Inverse Solution*

As part of the inverse solution, HYDRUS2 produces a correlation matrix which specifies degree of correlation between the fitted coefficients. The correlation matrix quantifies changes in model predictions caused by small changes in the final estimate of a particular parameter, relative to similar changes as a result of changes in the other parameters. The correlation matrix reflects the nonorthogonality between two parameter values. A value of ±1 suggests a perfect linear correlation whereas 0 indicates no correlation at all. The correlation matrix may be used to select which parameters, if any, are best kept constant in the parameter estimation process because of high correlation.

An important measure of the goodness of fit is the r^2 value for regression of the observed, $\hat{y}_i$, versus fitted, $y_i(\boldsymbol{b})$, values:

$$r^2 = \frac{\left[\sum w_i \hat{y}_i y_i - \frac{\sum \hat{y}_i \sum y_i}{\sum w_i} \right]^2}{\left[\sum w_i \ \hat{y}_i^2 - \frac{(\sum \hat{y}_i)^2}{\sum w_i} \right] \left[\sum y_i^2 - \frac{(\sum y_i)^2}{\sum w_i} \right]} \tag{7.3}$$

The r^2 value is a measure of the relative magnitude of the total sum of squares associated with the fitted equation; a value of 1 indicates a perfect correlation between the fitted and observed values.

HYDRUS2 provides additional statistical information about the fitted parameters such as the mean, standard error, T-value, and the lower and upper confidence limits (given in output file FIT.OUT). The standard error, s(b_j), is estimated from knowledge of the objective function, the number of observations, the number of unknown parameters to be fitted, and an inverse matrix [*Daniel and Wood*, 1971]. The *T*-value is obtained from the mean and standard error using the equation

$$T = \frac{b_j}{s(b_j)} \tag{7.4}$$

The values for *T* and s(b_j) provide absolute and relative measures of the deviations around the mean. HYDRUS2 also specifies the upper and lower bounds of the 95% confidence level around each fitted parameter b_j. It is desirable that the real value of the target parameter always be located in a narrow interval around the estimated mean as obtained with the optimization program. Large confidence limits indicate that the results are not very sensitive to the value of a particular parameter.

Finally, because of possible problems related to convergence and parameter uniqueness, we recommend to routinely rerun the program with different initial parameter estimates to verify that the program indeed converges to the same global minimum in the objective function. This is especially important for field data sets, which often exhibit considerable scatter in the measurements, or may cover only a narrow range of soil water contents, pressure heads, and/or concentrations. Whereas HYDRUS2 will not accept initial estimates that are out of range, it is ultimately the user's responsibility to select meaningful initial estimates.

8. PROBLEM DEFINITION

8.1. *Construction of Finite Element Mesh*

The finite element mesh is constructed by dividing the flow region into quadrilateral and/or triangular elements whose shapes are defined by the coordinates of the nodes that form the element corners. The program automatically subdivides the quadrilaterals into triangles, which are then treated as subelements.

Transverse lines [*Neuman*, 1974] formed by element boundaries must transect the mesh along the general direction of its shortest dimension. These transverse lines should be continuous and non-intersecting, but need not be straight. The nodes are numbered sequentially from 1 to *NumNP* (total number of nodes) by proceeding along each transverse line in the same direction. Elements are numbered in a similar manner. The maximum number of nodes on any transverse line, *IJ*, is used to determine the effective size of the finite element matrix (i.e., its band width). To minimize memory and time requirements, *IJ* should be kept as small as possible. The above rules for defining the finite element mesh apply only when Gaussian elimination is used to solve the matrix equations. Iterative methods (such as the conjugate gradient and ORTHOMIN methods) are not so restrictive since only non-zero entries in the coefficient matrix are stored in memory, and since the computational efficiency is less dependent upon the bandwidth of the matrix as compared to direct equation solvers.

The finite element dimensions must be adjusted to a particular problem. They should be made relatively small in directions where large hydraulic gradients are expected. Region with sharp gradients are usually located in the vicinity of the internal sources or sinks, or close to the soil surface where highly variable meteorological factors can cause fast changes in pressure head. Hence, we recommend normally using relatively small elements at and near the soil surface. The size of elements can gradually increase with depth to reflect the generally much slower changes in pressure heads at deeper depths. The element dimensions should also depend upon the soil hydraulic properties. For example, coarse-textured soils having relatively high n-values and small α-values (see Eqs. (2.19) and (2.21)) generally require a finer discretization than fine-textured soils. We also recommend using elements having approximately equal sizes to decrease numerical errors. For axisymmetric three-dimensional flow systems, the vertical axis must coincide with, or be to the left of, the left boundary of the mesh. No special restrictions are necessary to facilitate the soil root zone.

8.2. *Coding of Soil Types and Subregions*

Soil Types - An integer code beginning with 1 and ending with *NMat* (the total number of soil materials) is assigned to each soil type in the flow region. The appropriate material code is subsequently assigned to each nodal point *n* of the finite element mesh.

Interior material interfaces do not coincide with element boundaries. When different material numbers are assigned to the corner nodes of a certain element, material properties of this element will be averaged automatically by the finite element algorithm. This procedure will somewhat smooth soil interfaces.

A set of soil hydraulic parameters and solute transport characteristics must be specified for each soil material. Also, the user must define for each element the principal components of the conductivity anisotropy tensor, as well as the angle between the local and global coordinate systems.

As explained in Section 2.3, one additional way of changing the unsaturated soil hydraulic properties in the flow domain is to introduce scaling factors associated with the water content, the pressure head and the hydraulic conductivity. The scaling factors are assigned to each nodal point *n* in the flow region.

Subregions - Water and solute mass balances are computed separately for each specified subregion. The subregions may or may not coincide with the material regions. Subregions are characterized by an integer code which runs from 1 to *NLay* (the total number of subregions). A subregion code is assigned to each element in the flow domain.

8.3. *Coding of Boundary Conditions*

Flow boundary conditions were programmed in a fairly similar way as done in the UNSAT1 and UNSAT2 models of *Neuman* [1972] and *Neuman et al.* [1974], and in the SWMS_2D [*Šimůnek et al.*, 1992] and CHAIN_2D [*Šimůnek and van Genuchten*, 1994] codes. A boundary code, *Kode*(*n*), must be assigned to each node, *n*. If node *n* is to have a prescribed pressure head during a time step (Dirichlet boundary condition), *Kode*(*n*) must be set positive during that time step. If the volumetric flux of water entering or leaving the system at node *n* is prescribed during a time step (Neumann boundary condition), *Kode*(*n*) must be negative or zero.

Constant Boundary Conditions - The values of constant boundary conditions for a particular node, *n*, are given by the initial values of the pressure head, *h*(*n*), in case of Dirichlet boundary conditions, or by the initial values of the recharge/discharge flux, *Q*(*n*), in case of Neumann boundary conditions. Table 8.1 summarizes the use of the variables *Kode*(*n*), *Q*(*n*) and *h*(*n*) for various types of nodes.

Table 8.1. Initial settings of *Kode*(*n*), *Q*(*n*), and *h*(*n*) for constant boundary conditions.

Node Type	*Kode*(*n*)	*Q*(*n*)	*h*(*n*)
Internal; not sink/source	0	0.0	Initial Value
Internal; sink/source (Dirichlet condition)	1	0.0	Prescribed
Internal; sink/source (Neumann condition)	-1	Prescribed	Initial Value
Impermeable Boundary	0	0.0	Initial Value
Specified Head Boundary	1[+]	0.0	Prescribed
Specified Flux Boundary	-1[*]	Prescribed	Initial Value

[+] 5 and/or 6 may also be used
[*] -5 and/or -6 may also be used

Variable Boundary Conditions - Three types of variable boundary conditions can be imposed:

1. Atmospheric boundary conditions for which *Kode*(*n*)=±4,
2. Variable pressure head boundary conditions for which *Kode*(*n*)=+3, and
3. Variable flux boundary conditions for which *Kode*(*n*)=-3.

These conditions can be specified along any part of the boundary. It is not possible to specify more than one time-dependent boundary condition for each type. Initial settings of the variables *Kode*(*n*), *Q*(*n*) and *h*(*n*) for the time-dependent boundary conditions are given in Table 8.2.

Table 8.2. Initial settings of *Kode*(*n*), *Q*(*n*), and *h*(*n*) for variable boundary conditions.

Node Type	*Kode*(*n*)	*Q*(*n*)	*h*(*n*)
Atmospheric Boundary	-4	0.0	Initial Value
Variable Head Boundary	+3	0.0	Initial Value
Variable Flux Boundary	-3	0.0	Initial Value

Atmospheric boundary conditions are implemented when *Kode*(*n*)= ±4, in which case time-dependent input data for the precipitation, *Prec*, and evaporation, *rSoil*, rates must be specified in the input file ATMOSPH.IN. The potential fluid flux across the soil surface is determined by *rAtm*= *rSoil-Prec*. The actual surface flux is calculated internally by the program. Two limiting values of surface pressure head must also be provided: *hCritS* which specifies the maximum allowed pressure head at the soil surface (usually 0.0), and *hCritA* which specifies the minimum allowed surface pressure head (defined from equilibrium conditions between soil water and atmospheric vapor). The program automatically switches the value of *Kode*(*n*) from -4 to +4 if one of these two limiting points is reached. Table 8.3 summarizes the use of the variables *rAtm*, *hCritS* and *hCritA* during program execution. *Width*(*n*) in the table denotes the length of the boundary segment associated with node *n*.

Table 8.3. Definition of the variables *Kode*(*n*), *Q*(*n*) and *h*(*n*) when an atmospheric boundary condition is applied.

Kode(*n*)	*Q*(*n*)	*h*(*n*)	Event
-4	-*Width*(*n*)**rAtm*	Unknown	*rAtm*=*rSoil-Prec*
+4	Unknown	*hCritA*	Evaporation capacity is exceeded
+4	Unknown	*hCritS*	Infiltration capacity is exceeded

Variable head and flux boundary conditions along a certain part of the boundary are implemented when *Kode*(*n*)=+3 and -3, respectively. In that case, the input file ATMOSPH.IN must contain the prescribed time-dependent values of the pressure head, *ht*, or the flux, *rt*, imposed along the boundary. The values of *ht* or *rt* are assigned to particular nodes at specified times according to rules given in Table 8.4.

Table 8.4. Definition of the variables *Kode*(*n*), *Q*(*n*) and *h*(*n*) when variable head or flux boundary conditions are applied.

Node Type	*Kode*(*n*)	*Q*(*n*)	*h*(*n*)
Variable Head Boundary	+3	Unknown	*ht*
Variable Flux Boundary	-3	-*Width*(*n*)**rt*	Unknown

Water Uptake by Plant Roots - The program calculates the rate at which plants extract water from the soil root zone by evaluating the term *D* (equation (5.9)) in the finite element formulation. The code requires that *Kode*(*n*) be set equal to 0 or negative for all nodes in the root zone. Values of the potential transpiration rate, *rRoot*, must be specified at preselected times in the input file ATMOSPH.IN. Actual transpiration rates are calculated internally by the program as discussed in Section 2.2. The root uptake parameters are taken from an input file SELECTOR.IN. Values of the function *Beta*(*n*), which describes the potential water uptake distribution over the root zone (equation (2.9)), must be specified for each node in the flow domain (see the description of input Block I in Table 10.8 of Section 10). All parts of the flow region where *Beta*(*n*)>0 are treated as the soil root zone.

Deep Drainage from the Soil Profile - Vertical drainage, *q*(*h*), across the lower boundary of the soil profile is sometimes approximated by a flux which depends on the position of groundwater level (e.g., *Hopmans and Stricker*, 1989). If available, such a relationship can be implemented in the form of a variable flux boundary condition for which *Kode*(*n*)=-6. This boundary condition is implemented in HYDRUS2 by setting the logical variable *qGWLF* in the input file BOUNDARY.IN equal to ".true.". The discharge rate *Q*(*n*) assigned to node *n* is determined by the program as *Q*(*n*)=-*Width*(*n*)**q*(*h*) where *h* is the local value of the pressure head, and *q*(*h*) is given by

$$q(h) = -A_{qh} \exp(B_{qh} \,|\, h - GWL0L \,|) \qquad (8.1)$$

where A_{qh} and B_{qh} are empirical parameters which must be specified in the input file BOUNDARY.IN, together with *GWL0L* which represents the reference position of the groundwater level (usually set equal to the *z*-coordinate of the soil surface).

Free Drainage - Unit vertical hydraulic gradient boundary conditions can be implemented in the form of a variable flux boundary condition for which *Kode*(*n*)=-6. This boundary condition is implemented in HYDRUS2 by setting the logical variable *FreeD* in the input file BOUNDARY.IN equal to ".true.". The program determines the discharge rate *Q*(*n*) assigned to node *n* as *Q*(*n*)=-*Width*(*n*)**K*(*h*), where *h* is the local value of the pressure head, and *K*(*h*) is the hydraulic conductivity corresponding to this pressure head.

Seepage Faces - The initial settings of the variables *Kode*(*n*), *Q*(*n*) and *h*(*n*) for nodes along a seepage face are summarized in Table 8.5. All potential seepage faces must be identified before starting the numerical simulation. This is done by providing a list of nodes along each potential seepage face (see input Block J as defined in Table 10.10 of Section 10).

Table 8.5. Initial setting of *Kode*(*n*), *Q*(*n*), and *h*(*n*) for seepage faces.

Node Type	*Kode*(*n*)	*Q*(*n*)	*h*(*n*)
Seepage Face (initially saturated)	+2	0.0	0.0
Seepage Face (initially unsaturated)	-2	0.0	Initial Value

Drains - Table 8.6 summarizes the initial settings of the variables *Kode*(*n*), *Q*(*n*) and *h*(*n*) for nodes representing drains. All drains must be identified before starting the numerical simulation. This is done by providing a list of nodes representing drains, together with a list of elements around each drain whose hydraulic conductivities are to be adjusted according to discussion in Section 5.3.7 (see also input Block J as defined in Table 10.10 of Section 10).

Table 8.6. Initial setting of *Kode*(*n*), *Q*(*n*), and *h*(*n*) for drains.

Node Type	*Kode*(*n*)	*Q*(*n*)	*h*(*n*)
Drain (initially saturated)	+5	0.0	0.0
Drain (initially unsaturated)	-5	0.0	Initial Value

Solute and Heat Transport Boundary Conditions - The original version 1.1. of SWMS_2D [*Šimůnek et al*., 1992] assumed a strict relationship between the boundary conditions for water flow and solute transport. A first-type boundary condition for water flow forced the boundary condition for solute transport also to be of the first-type. Similarly, a second-type boundary condition for water flow induced a second- or third-type boundary condition for solute transport depending upon direction of the water flux. These strict relationships between the boundary conditions for water flow and solute transport have been abandoned in later versions. Selection of the type of boundary condition for the solute transport is now much more independent of the boundary condition implemented for water flow. The type of boundary condition to be invoked for solute or heat transport is specified by the input variable *KodCB* or *KodTB*, respectively. A positive sign of this variable means that a first-type boundary condition will be used. When *KodCB* (*KodTB*) is negative, HYDRUS2 selects a third-type boundary condition when the calculated water flux is directed into the region, or a second-type boundary condition when the water flux is zero or directed out of the region. One exception to these rules occurs for atmospheric boundary conditions when *Kode*(*n*)=±4 and *Q*(*n*)<0. HYDRUS2 assumes that solutes can leave the flow region across atmospheric boundaries only by gas diffusion. The solute flux in this situation becomes zero, i.e., c_0=0 in equation (6.30). Cauchy and Neumann boundary conditions are automatically applied to internal sinks/sources depending upon the direction of water flow. The dependence or independence of the solute and heat boundary conditions on time or the system is still defined through the variable *Kode*(*n*) as discussed above.

Although HYDRUS2 can implement first-type boundary conditions, we recommend users to invoke third-type conditions where possible. This is because third-type conditions, in general, are physically more realistic and preserve solute mass in the simulated system (e.g., *van Genuchten and Parker* [1984]; *Leij et al.* [1991]).

8.4. *Program Memory Requirements*

One single parameter statement is used at the beginning of the code to define the problem dimensions. All major arrays in the program are adjusted automatically according to these dimensions. This feature makes it possible to change the dimensions of the problem to be simulated without having to recompile all program subroutines. Different problems can be investigated by changing the dimensions in the parameter statement at the beginning of the main program, and subsequently linking all previously compiled subroutines with the main program when creating an executable file. Table 8.7 lists the array dimensions, which must be defined in the parameter statement.

Table 8.7. List of array dimensions in HYDRUS2.

Dimension	Description
NumNPD	Maximum number of nodes in finite element mesh
NumElD	Maximum number of elements in finite element mesh
MBandD	Maximum dimension of the bandwidth of matrix *A* when Gaussian elimination is used. Maximum number of nodes adjacent to a particular node, including itself, when iterative matrix solvers are used.
NumBPD	Maximum number of boundary nodes for which *Kode*(*n*)≠0
NSeepD	Maximum number of seepage faces
NumSPD	Maximum number of nodes along a seepage face
NDrD	Maximum number of drains
NElDrD	Maximum number of elements surrounding a drain
NMatD	Maximum number of materials
NTabD	Maximum number of items in the table of hydraulic properties generated by the program for each soil material
NumKD	Maximum number of available code number values (equals 6 in present version)
NSD	Maximum number of solutes (equals 6 in present version)
NObsD	Maximum number of observation nodes for which the pressure head, the water content, temperature and concentration are printed at each time level
MNorth	Maximum number of orthogonalizations performed when iterative solvers are used

8.5. *Matrix Equation Solvers*

Discretization of the governing partial differential equations for water flow (2.1), solute transport (3.11) and heat movement (4.1) leads to the system of linear equations

$$[A]\{x\} = \{b\} \tag{8.2}$$

in which matrix [*A*] is symmetric for water flow and asymmetric for solute and heat transport.

The original version of SWMS_2D [*Šimůnek et al.*, 1992] uses Gaussian elimination to solve both systems of linear algebraic equations. The invoked solvers took advantage of the banded nature of the coefficient matrices and, in the case of water flow, of the symmetric properties of the matrix. Such direct solution methods have several disadvantages as compared to iterative methods. Direct methods require a fixed number of operations (depending upon the size of the matrix), which increases approximately by the square of the number of nodes [*Mendoza et al.,* 1991]. Iterative methods, on the other hand, require a variable number of repeated steps, which increases at a much smaller rate (about 1.5) with the size of a problem [*Mendoza et al.,* 1991]. A similar reduction also holds for the memory requirement since iterative methods do not require one to store non-zero matrix elements. Memory requirements, therefore, increase at a much smaller rate with the size of the problem when iterative solvers are used [*Mendoza et al.,* 1991]. Round-off errors also represent less of a problem for iterative methods as compared to direct methods. This is because round-off errors in iterative methods are self-correcting [*Letniowski*, 1989]. Finally, for time-dependent problems, a reasonable approximation of the solution (i.e., the solution at the previous time step) exists for iterative methods, but not for direct methods [*Letniowski*, 1989]. In general, direct methods are more appropriate for relatively small problems, while iterative methods are more suitable for larger problems.

Many iterative methods have been used in the past for handling large sparse matrix equations. These methods include Jacobi, Gauss-Seidel, alternating direction implicit (ADI), block successive over-relaxation (BSSOR), successive line over-relaxation (SLOR), and strongly implicit procedures (SIP), among others [*Letniowski*, 1989]. More powerful preconditioned accelerated iterative methods, such as the preconditioned conjugate gradient method (PCG) [*Behie and Vinsome*, 1982], were introduced more recently. *Sudicky and Huyakorn* [1991] gave three advantages of the PCG procedure as compared to other iterative methods: PCG can be readily modified for finite element methods with irregular grids, the method does not require iterative parameters, and PCG usually outperforms its iterative counterparts for situations involving relatively stiff matrix conditions.

HYDRUS2 implements both direct and iterative methods for solving the system of linear algebraic equations given by (8.2). Depending upon the size of matrix [*A*], we use either direct Gaussian elimination or the preconditioned conjugate gradient method [*Mendoza et al.,* 1991] for water flow and the ORTHOMIN (preconditioned conjugate gradient squared) procedure [*Mendoza et al.,* 1991] for solute transport. Gaussian elimination is used if either the bandwidth of matrix [*A*] is smaller than 20, or the total number of nodes is smaller than 500. The iterative methods used in

HYDRUS2 were adopted from the ORTHOFEM software package of *Mendoza et al.* [1991].

The preconditioned conjugate gradient and ORTHOMIN methods consist of two essential parts: initial preconditioning, and the iterative solution with either conjugate gradient or ORTHOMIN acceleration [*Mendoza et al.,* 1991]. Incomplete lower-upper (ILU) preconditioning is used in ORTHOFEM when matrix [*A*] is factorized into lower and upper triangular matrices by partial Gaussian elimination. The preconditioned matrix is subsequently repeatedly inverted using updated solution estimates to provide a new approximation of the solution. The orthogonalization-minimization acceleration technique is used to update the solution estimate. This technique insures that the search direction for each new solution is orthogonal to the previous approximate solution, and that either the norm of the residuals (for conjugate gradient acceleration [*Meijerink and van der Vorst*, 1981]) or the sum of the squares of the residuals (for ORTHOMIN [*Behie and Vinsome*, 1982]) is minimized. More details about the two methods is given in the user's guide of ORTHOFEM [*Mendoza et al.,* 1991] or in *Letniowski* [1989]. *Letniowski* [1989] also gives a comprehensive review of accelerated iterative methods, as well as of the preconditional techniques.

9. EXAMPLE PROBLEMS

9.1. *Direct Example Problems*

Seven example problems are presented in this section. The first three examples are identical to those provided previously with the first version of SWMS_2D [*Šimůnek et al.*, 1992]. The other four examples in this section are identical to those provided with CHAIN_2D [*Šimůnek and van Genuchten*, 1994], and are included mainly for mathematical verification purposes, and for demonstrating new features of HYDRUS2, i.e., non-equilibrium and nonlinear adsorption, sequential first-order decay reactions, solute diffusion in the gas phase, and/or heat transport.

Examples 1 and 2 provide comparisons of the water flow part of HYDRUS2 code with results from both the UNSAT2 code of *Neuman* [1974] and the SWATRE code of *Belmans et al.* [1983]. Example 3 serves to verify the accuracy of the solute transport part of HYDRUS2 by comparing numerical results against those obtained with a two-dimensional analytical solution during steady-state groundwater flow. The results obtained with the HYDRUS2 code for these three examples are identical to the results obtained with SWMS_2D. Example 4 serves to verify the accuracy of HYDRUS2 by comparing numerical results for a problem with three solutes involved in a sequential first-order decay chain against results obtained with an analytical solution during one-dimensional steady-state water flow [*van Genuchten*, 1985]. Example 5 considers one-dimensional transport of a solute undergoing nonlinear cation adsorption. Numerical results are compared with experimental data and previous numerical solutions obtained with the MONOC code of *Selim et al.* [1987] and the HYDRUS code of *Kool and van Genuchten* [1991]. Example 6 serves to verify the accuracy of HYDRUS2 in describing nonequilibrium adsorption by comparing numerical results against experimental data and previous numerical predictions during one-dimensional steady-state water flow [*van Genuchten*, 1981]. Example 7 demonstrates numerical results for a field infiltration experiment involving a two-layered axisymmetric three-dimensional flow domain. The infiltrating water was assumed to have a higher temperature than the soil, and to contain an organic compound (parent pesticide) which in the soil profile degraded into two sequential daughter products.

9.1.1. *Example 1 - Column Infiltration Test*

This example simulates a one-dimensional laboratory infiltration experiment discussed by *Skaggs et al.* [1970]. The example was used later by *Davis and Neuman* [1983] as a test problem for the UNSAT2 code. Hence, the example provides a means of comparing results obtained with the HYDRUS2 and UNSAT2 codes.

Figure 9.1 gives a graphical representation of the soil column and the finite element mesh used for the numerical simulations. The soil water retention and relative hydraulic conductivity functions of the sandy soil are presented in Figure 9.2. The sand was assumed to be at an initial pressure head of -150 cm. The soil hydraulic properties were assumed to be homogenous and isotropic with a saturated hydraulic conductivity of 0.0433 cm/min. The column was subjected to ponded infiltration (a Dirichlet boundary condition) at the soil surface, resulting in one-dimensional vertical water flow. The open bottom boundary of the soil column was simulated by implementing a no-flow boundary condition during unsaturated flow (*h*<0), and a seepage face with *h*=0 when the bottom boundary becomes saturated (this last condition was not reached during the simulation). The impervious sides of the column were simulated by imposing no-flow boundary conditions.

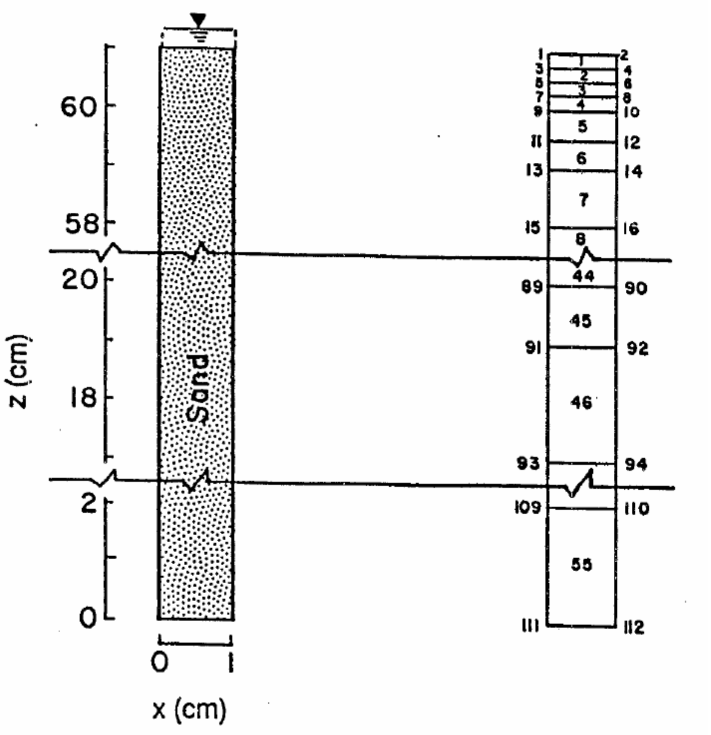


Fig. 9.1. Flow system and finite element mesh for example 1.

The simulation was carried out for 90 min, which corresponds to the total time duration of the experiment. Figure 9.3 shows the calculated instantaneous (q_0) and cumulative (I_0) infiltration rates simulated with HYDRUS2. The calculated results agree closely with those obtained by *Davis and Neuman* [1983] using their UNSAT2 code.

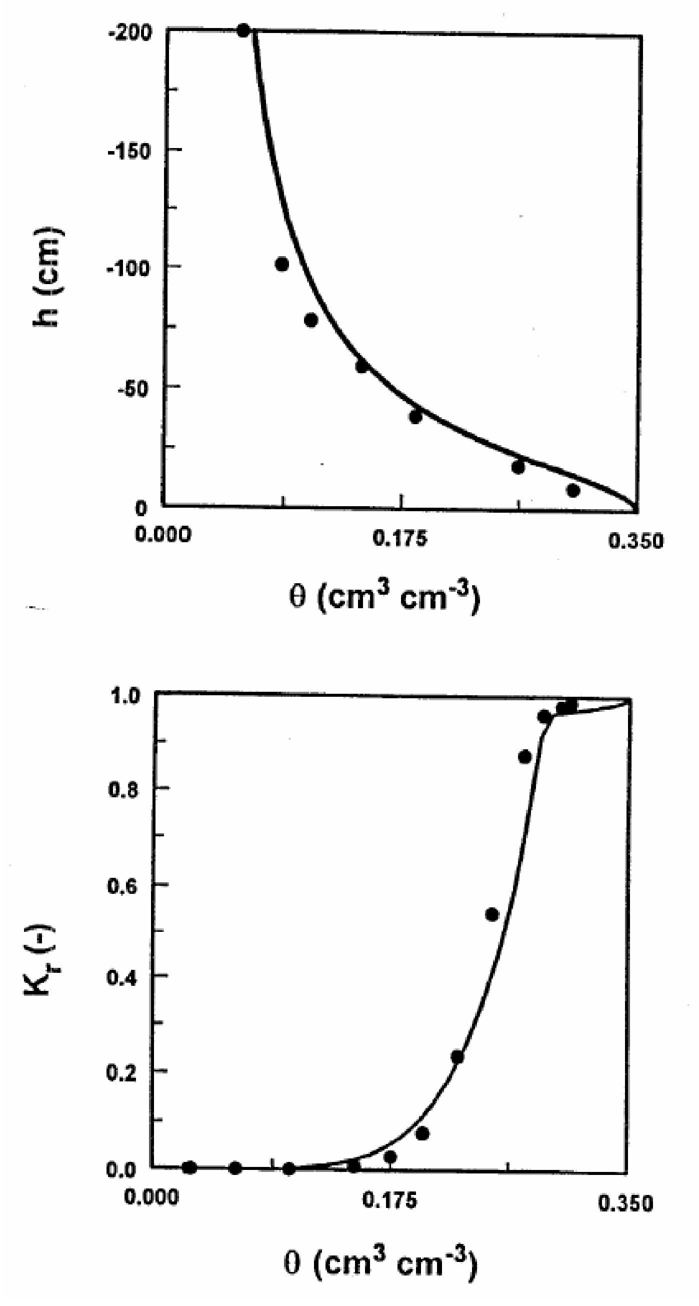


Fig. 9.2. Retention and relative hydraulic conductivity functions for example 1. The solid circles are UNSAT2 input data [*Davis and Neuman*, 1983].

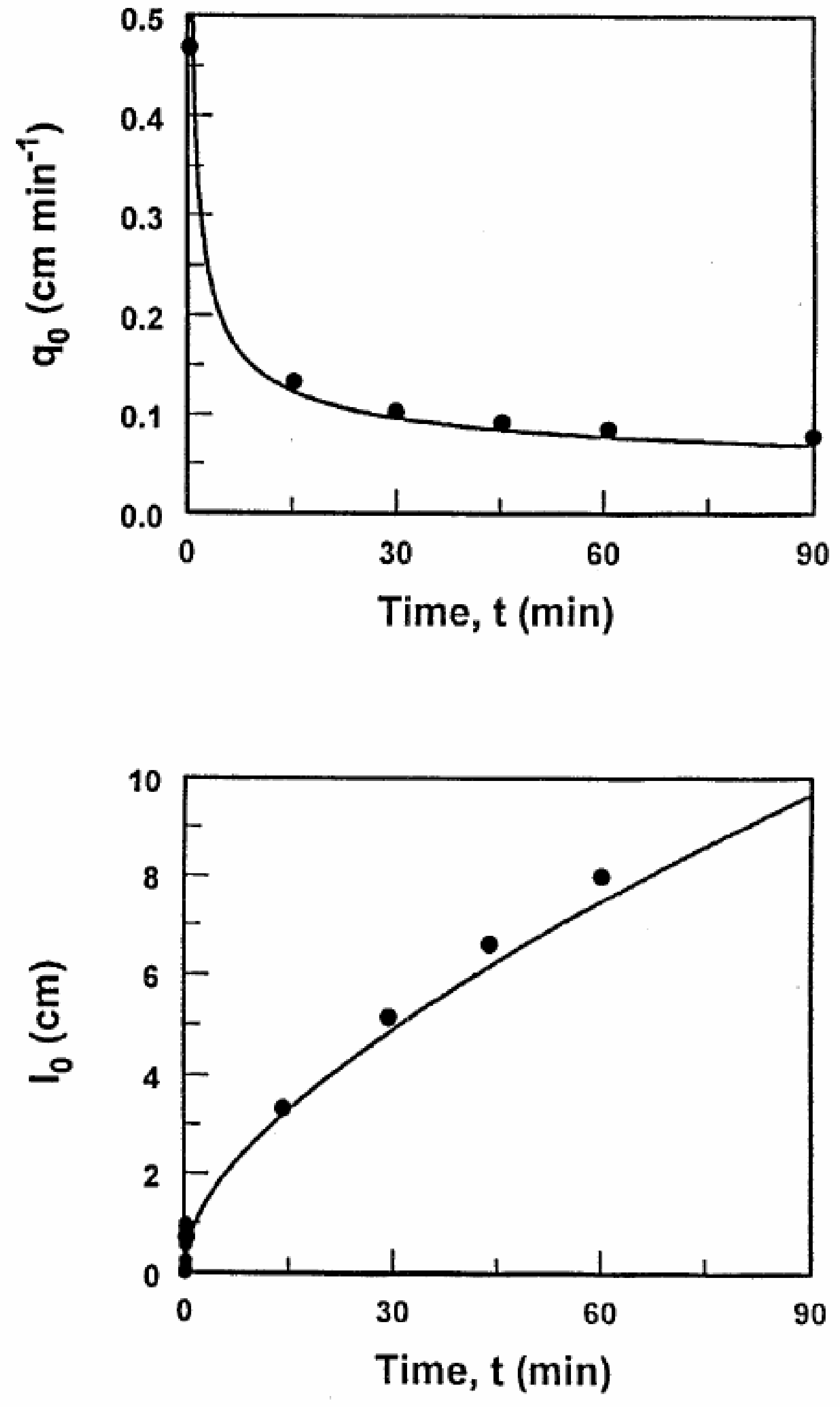


Fig. 9.3. Instantaneous, q_0, and cumulative, I_0, infiltration rates simulated with the HYDRUS2 (solid lines) and UNSAT2 (solid circles) codes for example 1.

9.1.2. *Example 2 - Water Flow in a Field Soil Profile Under Grass*

This example considers one-dimensional water flow in a field profile of the Hupselse Beek watershed in the Netherlands. Atmospheric data and observed ground water levels provided the required boundary conditions for the numerical model. Calculations were performed for the period of April 1 to September 30 of the relatively dry year 1982. Simulation results obtained with HYDRUS2 will be compared with those generated with the SWATRE computer program [*Feddes et al.*, 1978, *Belmans et al.*, 1983].

The soil profile (Fig. 9.4) consisted of two layers: a 40-cm thick A-horizon, and a B/C-horizon which extended to a depth of about 300 cm. The depth of the root zone was 30 cm. The mean scaled hydraulic functions of the two soil layers in the Hupselse Beek area [*Císlerová,* 1987; *Hopmans and Stricker*, 1989] are presented in Figure 9.5.

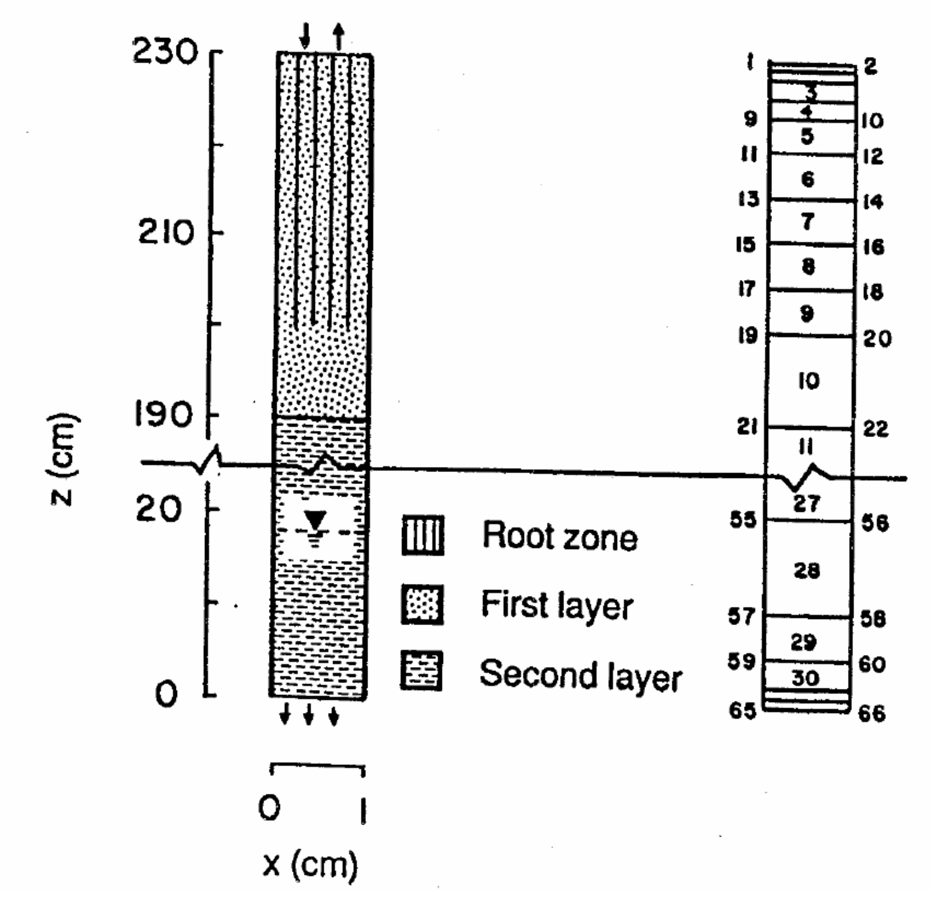


Fig. 9.4. Flow system and finite element mesh for example 2.

The soil surface boundary conditions involved actual precipitation and potential transpiration rates for a grass cover. The surface fluxes were incorporated by using average daily rates distributed uniformly over each day. The bottom boundary condition consisted of a prescribed drainage flux - groundwater level relationship, $q(h)$, as given by equation(8.1). The groundwater level was initially set at 55 cm below the soil surface. The initial moisture profile was taken to be in equilibrium with the initial ground water level.

Figure 9.6 presents input values of the precipitation and potential transpiration rates. Calculated cumulative transpiration and cumulative drainage amounts as obtained with the HYDRUS2 and SWATRE codes are shown in Figure 9.7. The pressure head at the soil surface and the arithmetic mean pressure head of the root zone during the simulated season are presented in Figure 9.8. Finally, Figure 9.9 shows variations in the calculated groundwater level with time.

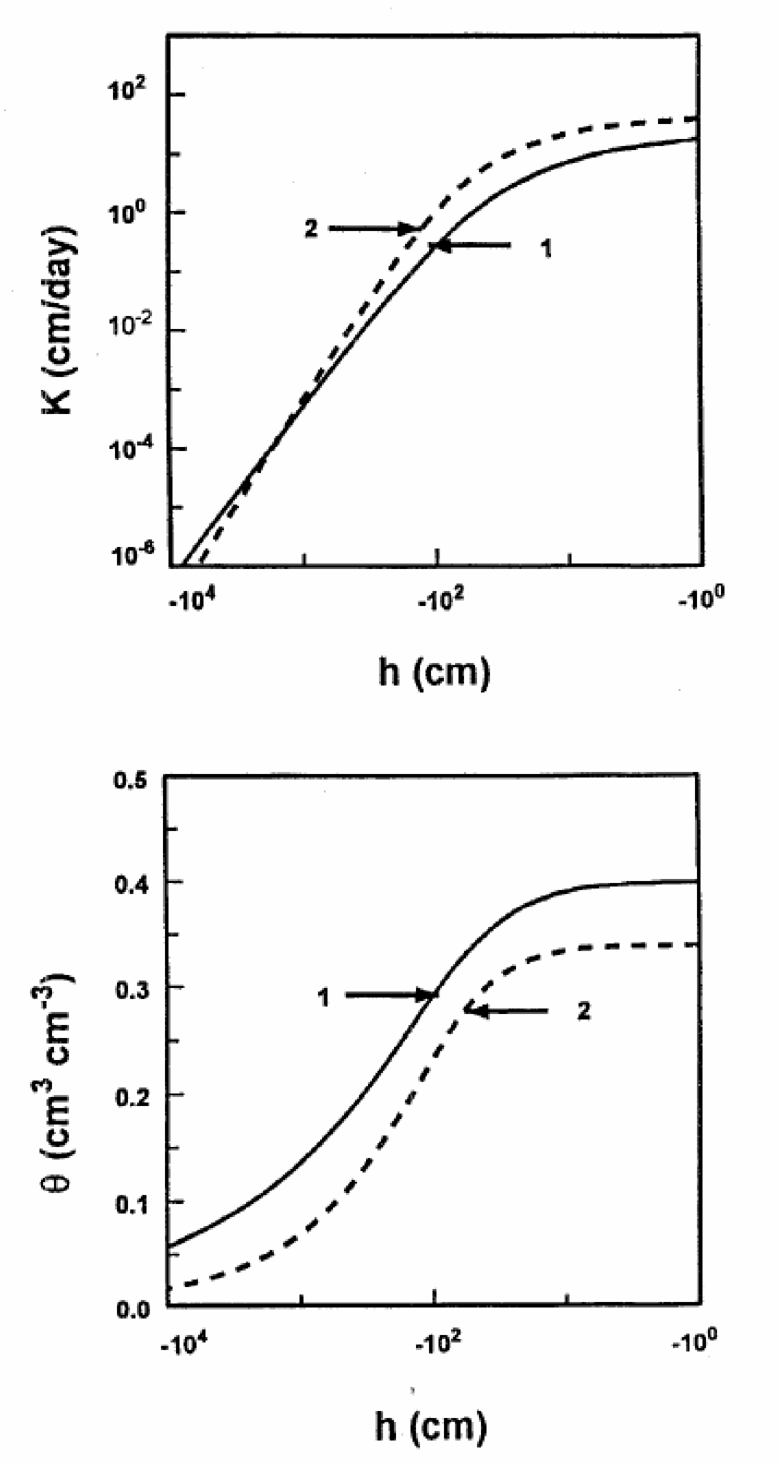
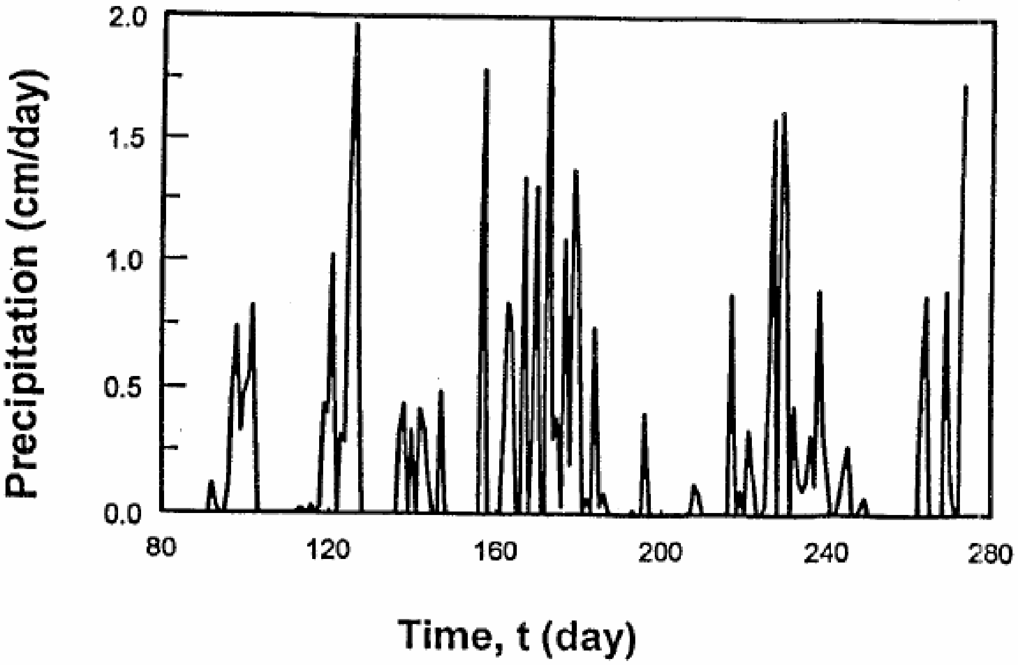


Fig. 9.5. Unsaturated hydraulic properties of the first and second soil layers for example 2.



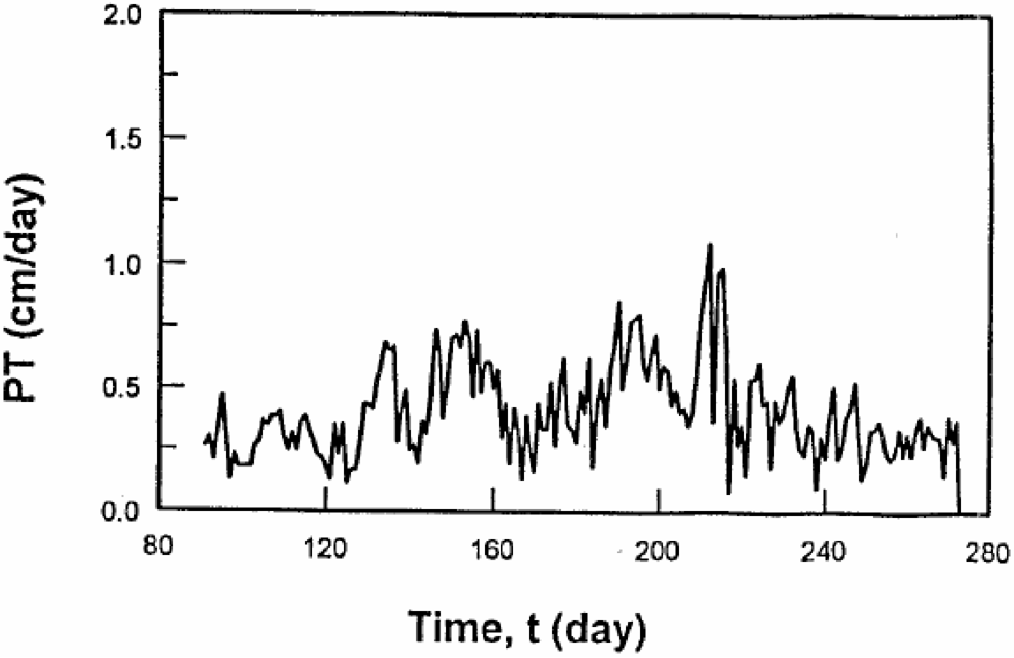


Fig. 9.6. Precipitation and potential transpiration rates for example 2.

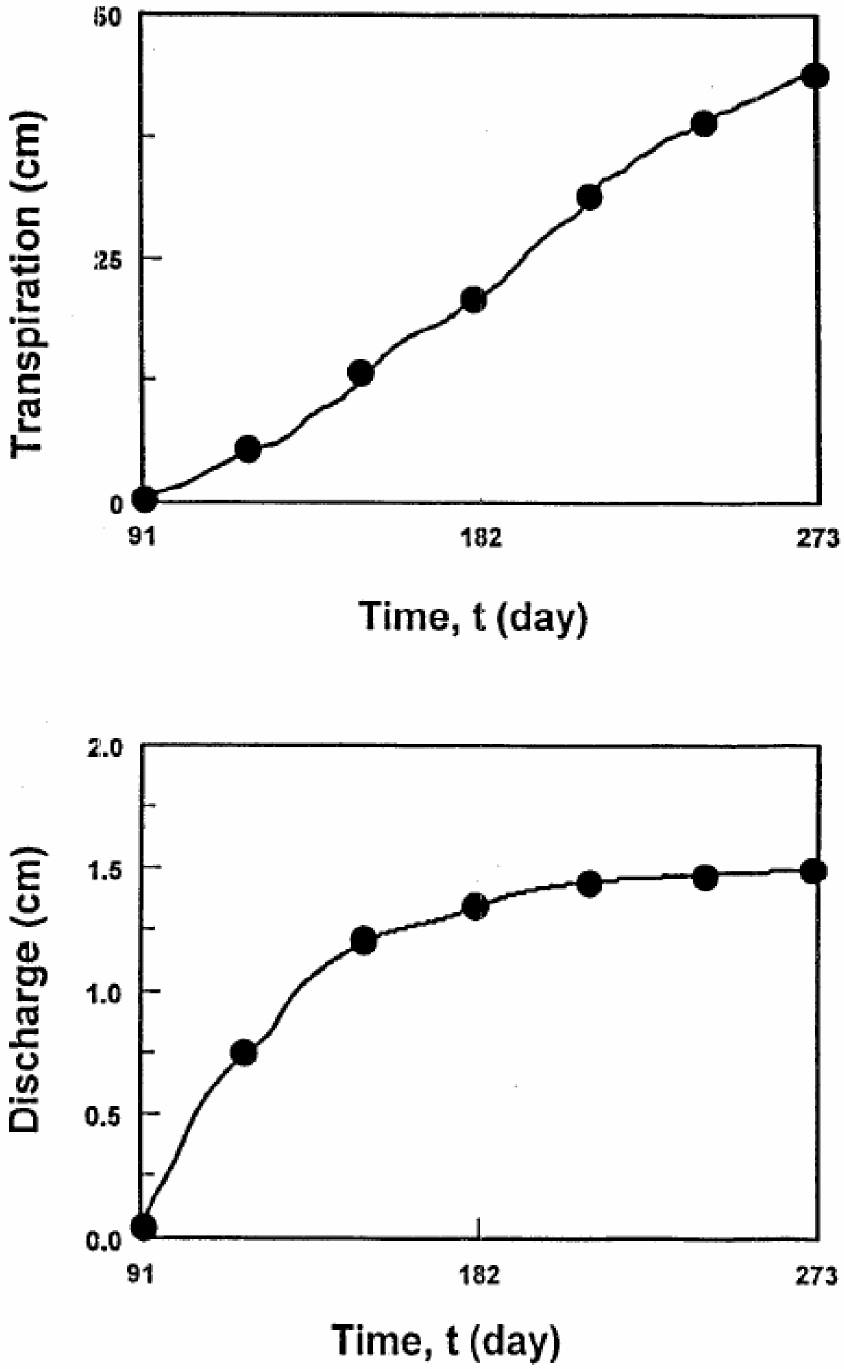


Fig. 9.7. Cumulative values for the actual transpiration and bottom discharge rates for example 2 as simulated with HYDRUS2 (solid line) and SWATRE (solid circles).

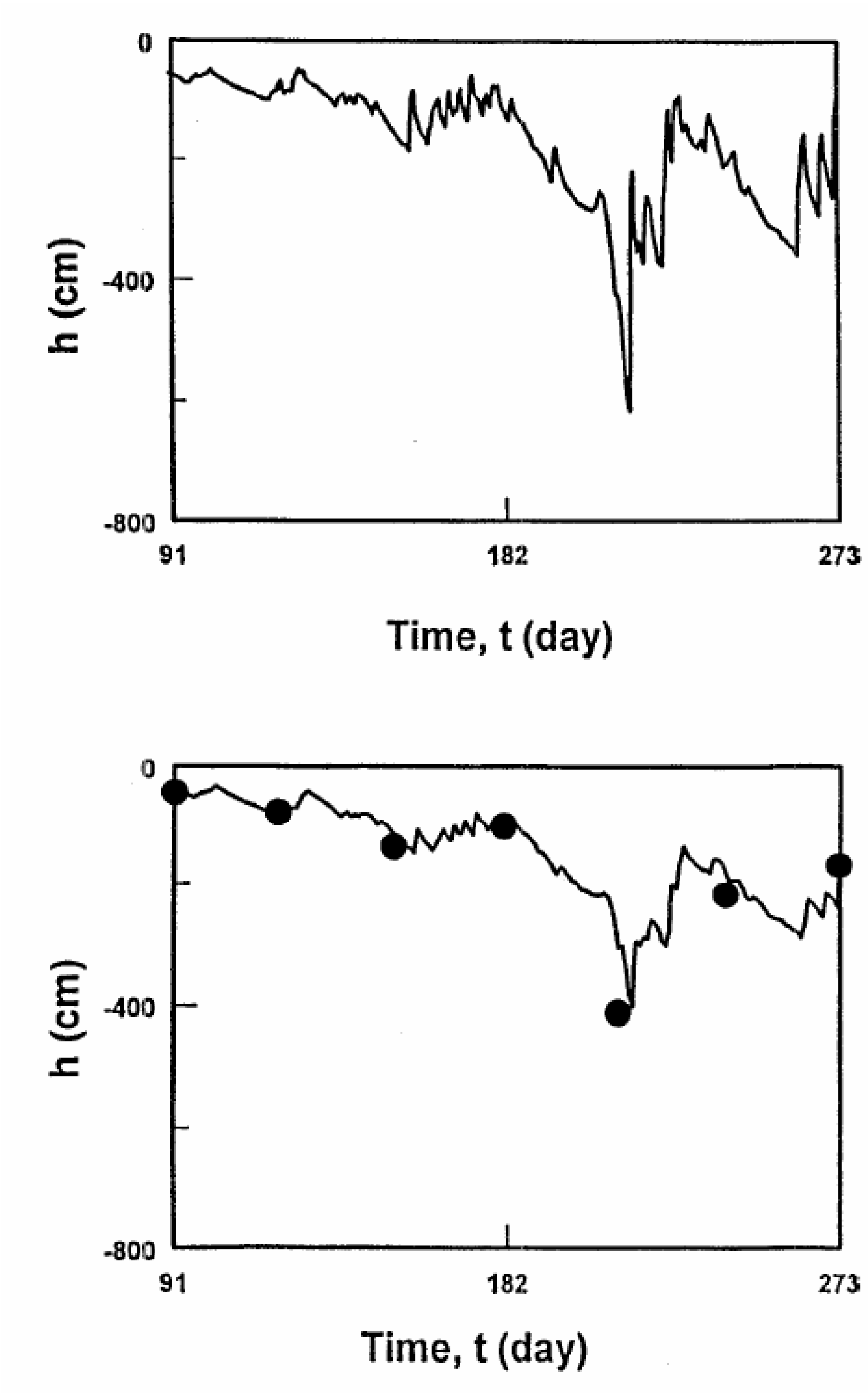


Fig. 9.8. Pressure head at the soil surface and mean pressure head of the root zone for example 2 as simulated with HYDRUS2 (solid lines) and SWATRE (solid circles).

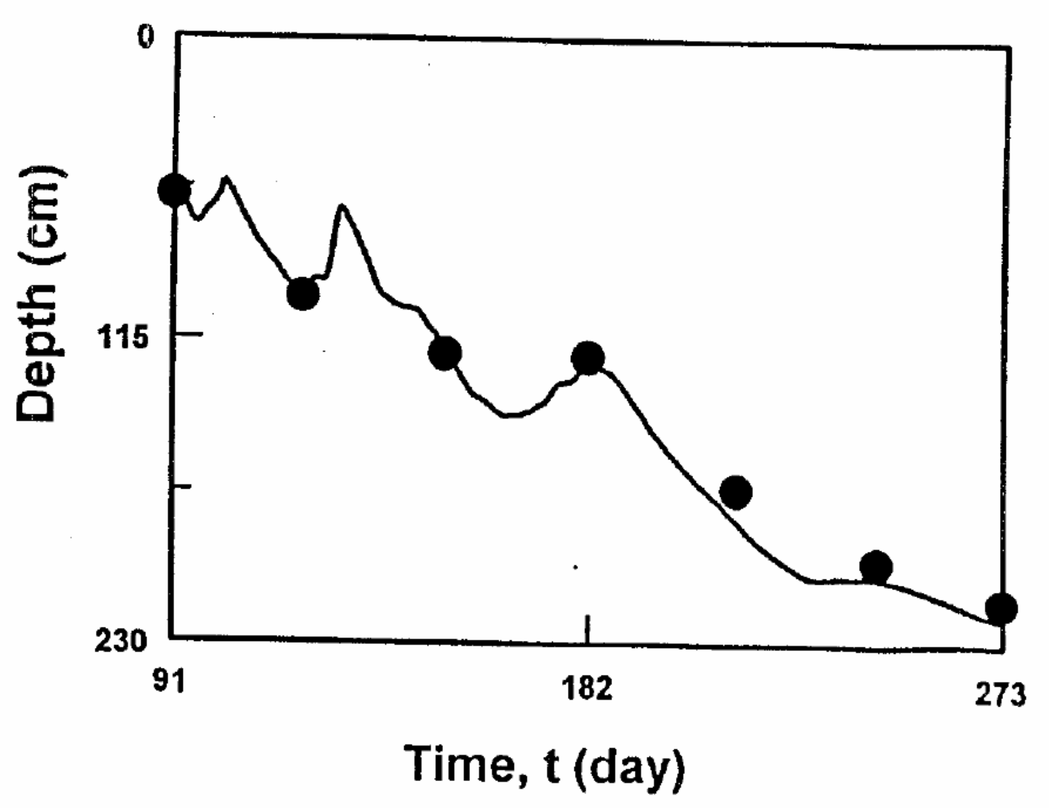


Fig. 9.9. Location of the groundwater table versus time for example 2 as simulated with the HYDRUS2 (solid line) and SWATRE (solid circles) computer programs.

9.1.3. *Example 3 - Two-Dimensional Solute Transport*

This example was used to verify the mathematical accuracy of the solute transport part of HYDRUS2. *Cleary and Ungs* [1978] published several analytical solutions for two-dimensional dispersion problems. One of these solutions holds for solute transport in a homogeneous, isotropic porous medium during steady-state unidirectional groundwater flow. The solute transport equation (3.11) for this situation reduces to

$$D_T \frac{\partial^2 c}{\partial x^2} + D_L \frac{\partial^2 c}{\partial z^2} - v \frac{\partial c}{\partial z} - \lambda R c = R \frac{\partial c}{\partial t} \tag{9.1}$$

where λ is a first-order degradation constant, D_L and D_T are the longitudinal and transverse dispersion coefficients, respectively; v is the average pore water velocity (q_z/θ) in the flow direction, and z and x are the spatial coordinates parallel and perpendicular to the direction of flow. The initially solute-free medium is subjected to a solute source, c_0, of unit concentration. The source covers a length $2a$ along the inlet boundary at z=0, and is located symmetrically about the coordinate x=0. The transport region of interest is the half- plane (z≥0; -∞≤x≤∞). The boundary conditions may be written as:

$$\begin{aligned} &c(x,0,t) = c_0 \qquad -a \le x \le a \\ &c(x,0,t) = 0 \qquad \text{other values of } x \\ &\lim_{z\to\infty} \frac{\partial c}{\partial z} = 0 \\ &\lim_{x\to\pm\infty} \frac{\partial c}{\partial x} = 0 \end{aligned} \tag{9.2}$$

The analytical solution of the above transport problem is [*Javandel et al.,* 1984, *Leij and Bradford*, 1994]

$$\begin{aligned} c(x,z,t) = &\frac{c_0\, z}{4\,(\pi\, D_L)^{1/2}} \exp\left(\frac{vz}{2D_L}\right) \int_0^{t/R} \exp\left[-\left(\lambda R + \frac{v^2}{4D_L}\right)\tau - \frac{z^2}{4\, D_L\, \tau}\right] \tau^{-3/2} \\ &\cdot \left[erf\left(\frac{a-x}{2(D_T\,\tau)^{1/2}}\right) + erf\left(\frac{a+x}{2(D_T\,\tau)^{1/2}}\right)\right] d\tau \end{aligned} \tag{9.3}$$

The input transport parameters for two simulations are listed in Table 9.1. The width of the source was assumed to be 100 m. Because of symmetry, calculations were carried out only for the quarter plane where x≥0 and z≥0.

Figure 9.10 shows the calculated concentration front (taken at a concentration of 0.1) at selected times for the first set of transport parameters in Table 9.1. Note the close agreement between the analytical and numerical results. Excellent agreement is also obtained for the calculated concentration distributions after 365 days at the end of the simulation (Fig. 9.11). Figures 9.12 and 9.13 show similar results for the second set of transport parameters listed in Table 9.1.

Table 9.1. Input parameters for example 3.

Parameter	Example 3a	Example 3b
v [m/day]	0.1	1.0
D_T [m^2/day]	1.0	0.5
D_L [m^2/day]	1.0	1.0
λ [day^{-1}]	0.0	0.01
R [-]	1.0	3.0
c_0 [-]	1.0	1.0

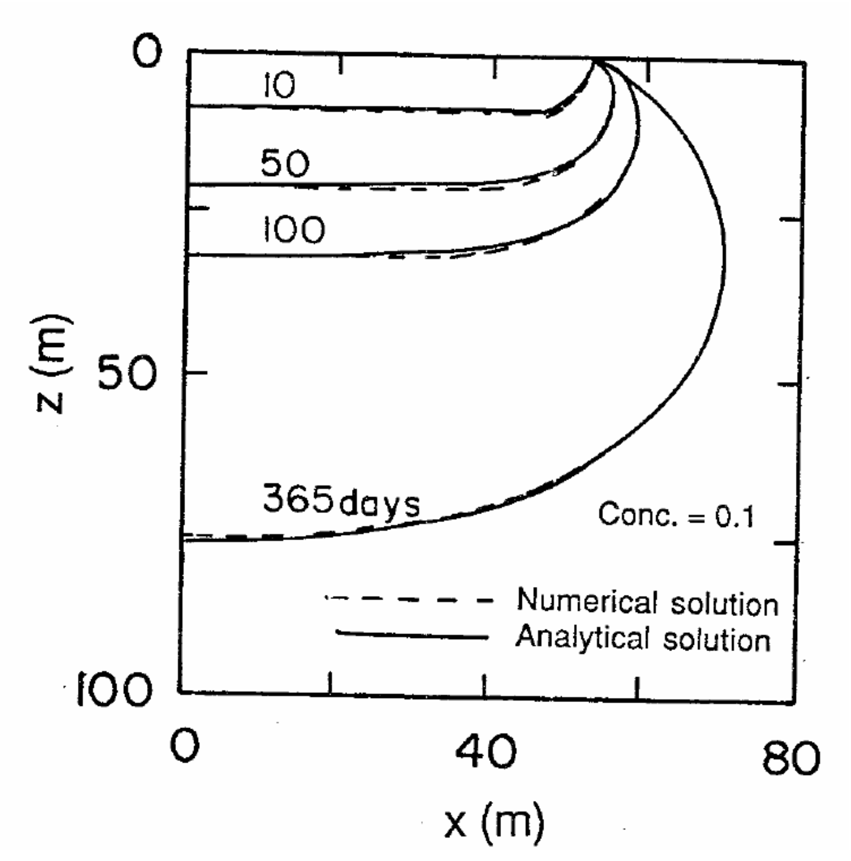


Fig. 9.10. Advancement of the concentration front (c=0.1) for example 3a as calculated with HYDRUS2 (dotted lines) and the analytical solution (solid lines).

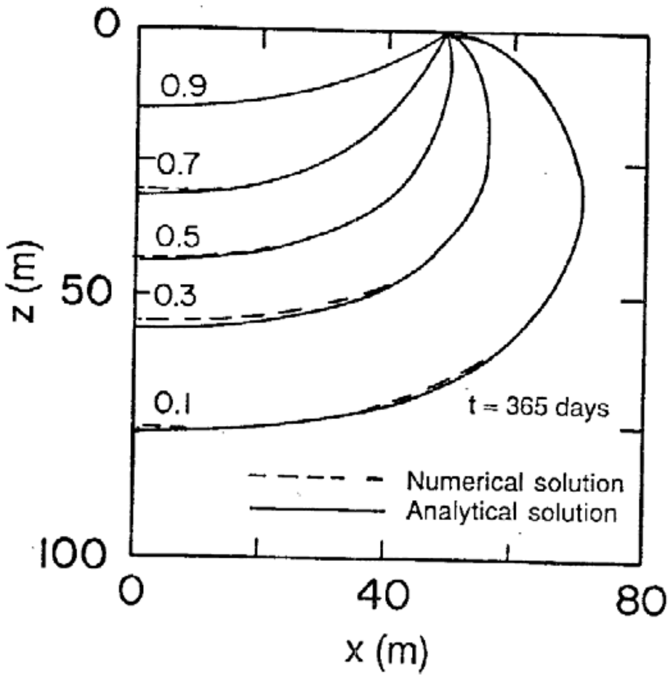


Fig. 9.11. Concentration profile at the end of the simulation (*t*=365 days) for example 3a calculated with HYDRUS2 (dotted lines) and the analytical solution (solid lines).

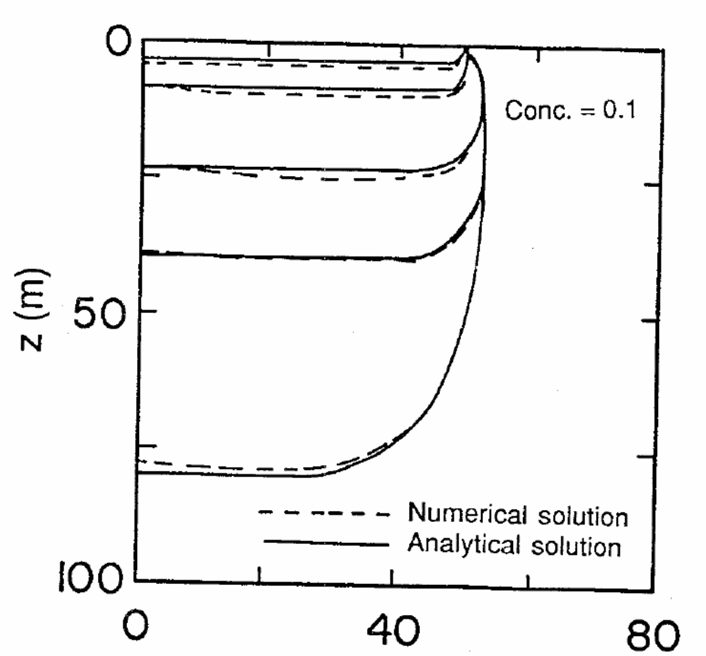


Fig. 9.12. Advancement of the concentration front (*c*=0.1) for example 3b as calculated with HYDRUS2 (dotted lines) and the analytical solution (solid lines).

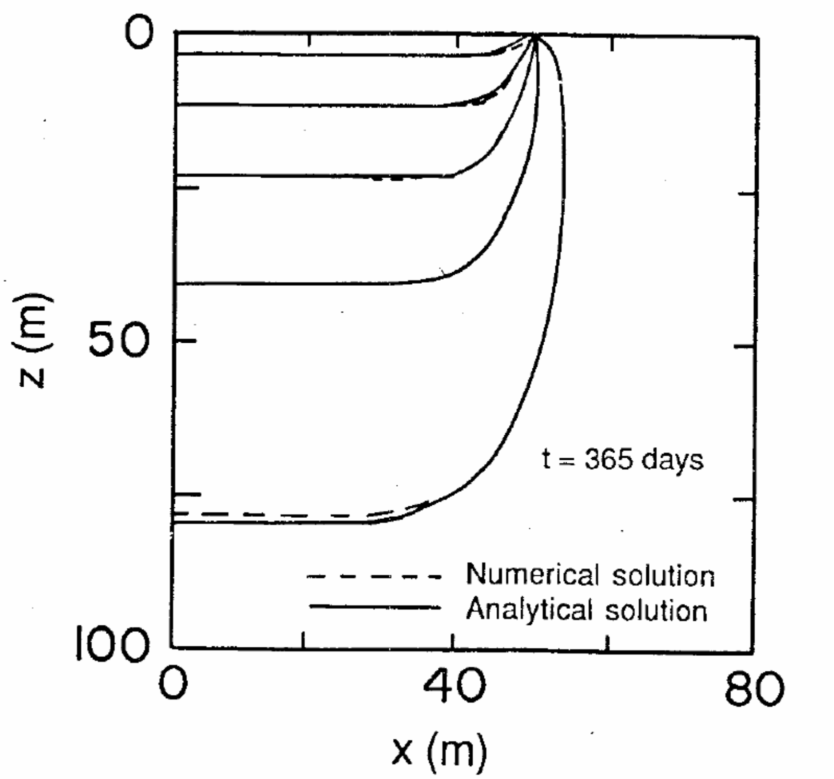


Fig. 9.13. Concentration profile at the end of the simulation (*t*=365 days) for example 3b as calculated with HYDRUS2 (dotted lines) and the analytical solution (solid lines).

9.1.4. *Example 4 - One-Dimensional Solute Transport with Nitrification Chain*

This example was used to verify in part the mathematical accuracy of the solute transport part of HYDRUS2. Numerical results will be compared with results generated with an analytical solution published by *van Genuchten* [1985] for one-dimensional convective-dispersive transport of solutes involved in sequential first-order decay reactions. The analytical solution holds for solute transport in a homogeneous, isotropic porous medium during steady-state unidirectional groundwater flow. Solute transport equations (3.1) and (3.2) for this situation reduce to

$$R_1\frac{\partial c_1}{\partial t} = D\frac{\partial^2 c_1}{\partial x^2} - v\frac{\partial c_1}{\partial x} - \mu_1 R_1 c_1 \qquad (9.4)$$

$$R_i \frac{\partial c_i}{\partial t} = D \frac{\partial^2 c_i}{\partial x^2} - v \frac{\partial c_i}{\partial x} + \mu_{i-1} R_{i-1} c_{i-1} - \mu_i R_i c_i \qquad i = 2, 3 \tag{9.5}$$

where μ is a first-order degradation constant, D is the dispersion coefficient, v is the average pore water velocity (q_x/θ) in the flow direction, x is the spatial coordinate in the direction of flow, and where it is assumed that 3 solutes participate in the decay chain. The specific example used here applies to the three-species nitrification chain

$$NH_4^+ \rightarrow NO_2^- \rightarrow NO_3^- \tag{9.6}$$

and is the same as described by *van Genuchten* [1985], and earlier by *Cho* [1971]. The boundary conditions may be written as:

$$\begin{aligned}
&\left(-D \frac{\partial c_1}{\partial x} + v c_1 \right) = v \, c_{0,1}(0, t) \\
&\left(-D \frac{\partial c_i}{\partial x} + v c_i \right) = 0 \qquad\qquad i = 2, 3 \\
&\lim_{x \to \infty} \frac{\partial c_i}{\partial x} = 0 \qquad\qquad\quad i = 1, 2, 3
\end{aligned} \tag{9.7}$$

The experiment involves the application of a NH_4^+ solution to an initially solute-free medium ($c_i = 0$). The input transport parameters for the simulation are listed in Table 9.2.

Figure 9.14 shows concentration profiles for all three solutes at time 200 hours, calculated both numerically with HYDRUS2 and analytically with the CHAIN code of *van Genuchten* [1985]. Figure 9.15 shows the concentration profiles at three different times (50, 100, and 200 hours) for NH_4^+, NO_2^-, and NO_3^-, respectively. The numerical results in each case duplicated the analytical results.

Table 9.2. Input parameters for example 4.

Parameter	Value
v [cm/hour]	1.0
D [cm^2/hour]	0.18
μ_1 [hour^{-1}]	0.005
μ_2 [hour^{-1}]	0.1
μ_3 [hour^{-1}]	0.0
R_1 [-]	2.0
R_2 [-]	1.0
R_3 [-]	1.0
c_i [-]	0.0
$c_{0,1}$ [-]	1.0

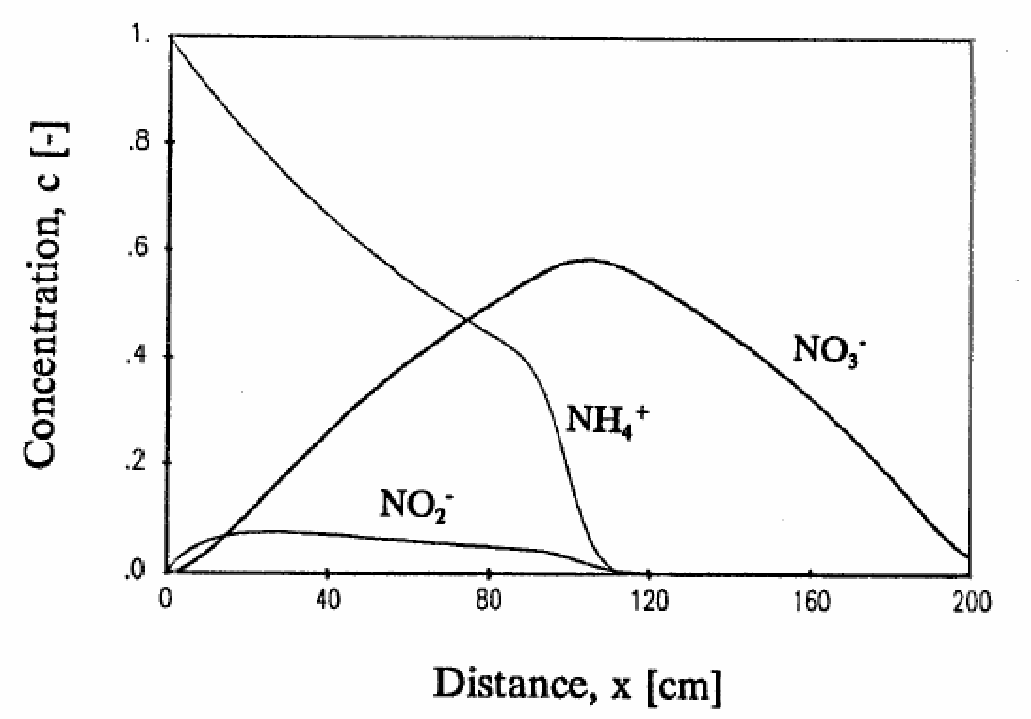


Fig. 9.14. Analytically and numerically calculated concentration profiles for NH_4^+, NO_2^-, and NO_3^- after 200 hours for example 4.

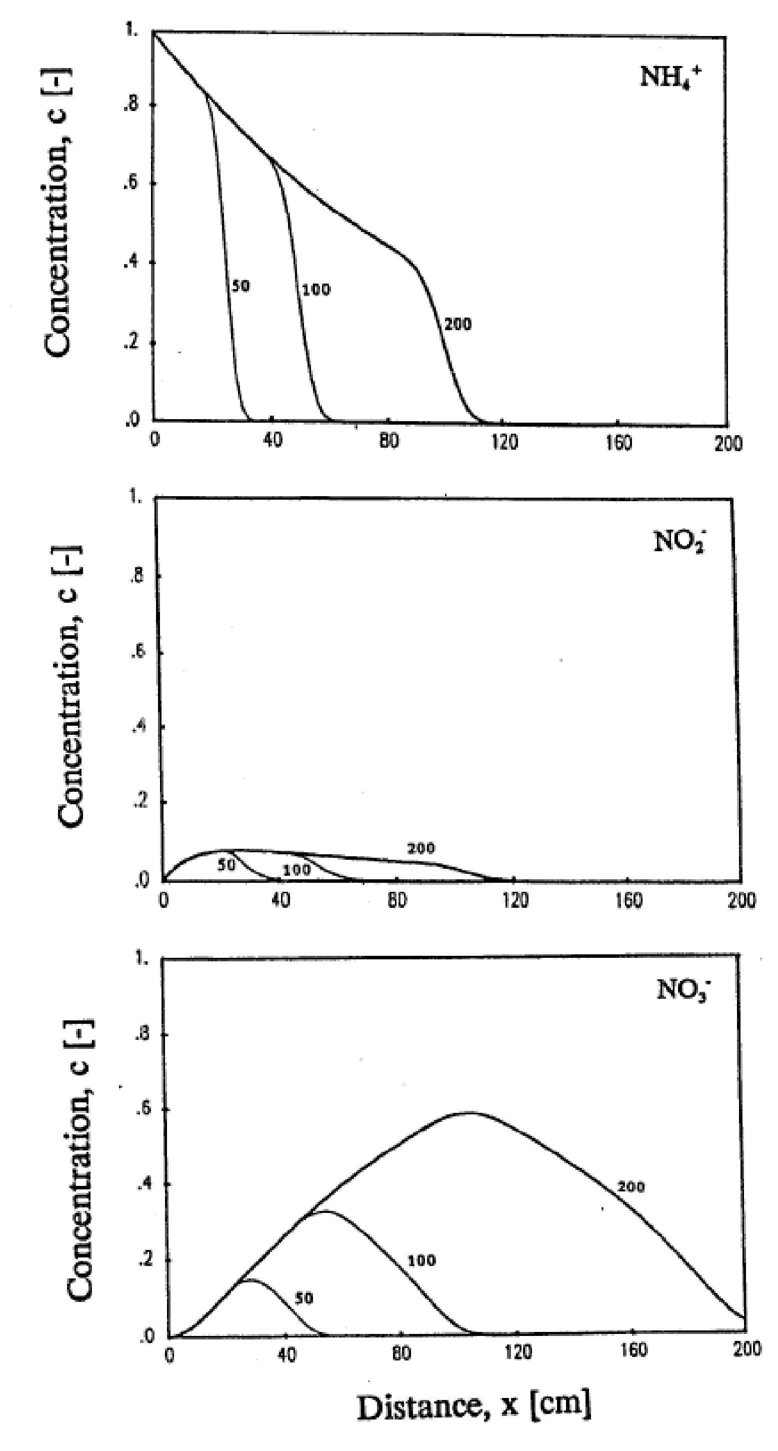


Fig. 9.15. Analytically and numerically calculated concentration profiles for NH_4^+ (top), NO_2^- (middle), NO_3^- (bottom) after 50, 100, and 200 hours for example 4.

9.1.5. *Example 5 - One-Dimensional Solute Transport with Nonlinear Cation Adsorption*

The experiment discussed in this example was conducted by *Selim et al.* [1987], and used later for validation of the HYDRUS code [*Kool and van Genuchten*, 1991]. The soil in this experiment was Abist loam. A 10.75-cm long soil column was first saturated with a 10 mmol$_c$L^{-1} CaCl$_2$ solution. The experiment consisted of applying a 14.26 pore volume pulse (t=358.05 hours) of 10 mmol$_c$L^{-1} MgCl$_2$ solution, followed by the original CaCl$_2$ solution. The adsorption isotherm was determined with the help of batch experiments [*Selim et al.*, 1987], and fitted with the Freundlich equation (3.3) [*Kool and van Genuchten*, 1991]. The Freundlich isotherm parameters, as well as other transport parameters for this problem, are listed in Table 9.3. First- and second-type boundary conditions were applied at the top and bottom of the soil column, respectively.

The observed Mg breakthrough curve is shown in Figure 9.16, together with simulated breakthrough curves obtained with HYDRUS2, the MONOC code of *Selim et al.* [1987] and the HYDRUS code of *Kool and van Genuchten* [1991]. The results indicate a reasonable prediction of the measured breakthrough curve using HYDRUS2, and close correspondence between the simulated results obtained with the HYDRUS2 and MONOC models. The HYDRUS2 results became identical to those generated with HYDRUS when a third-type boundary condition was invoked at the top of the soil column.

Table 9.3. Input parameters for example 5.

Parameter	Value
q [cm/hour]	0.271
D [cm^2/hour]	1.167
ρ [g/cm^3]	0.884
θ [-]	0.633
c_0 [mmol$_c$/L]	10.0
k_s [cm^3/g]	1.687
β [-]	1.615

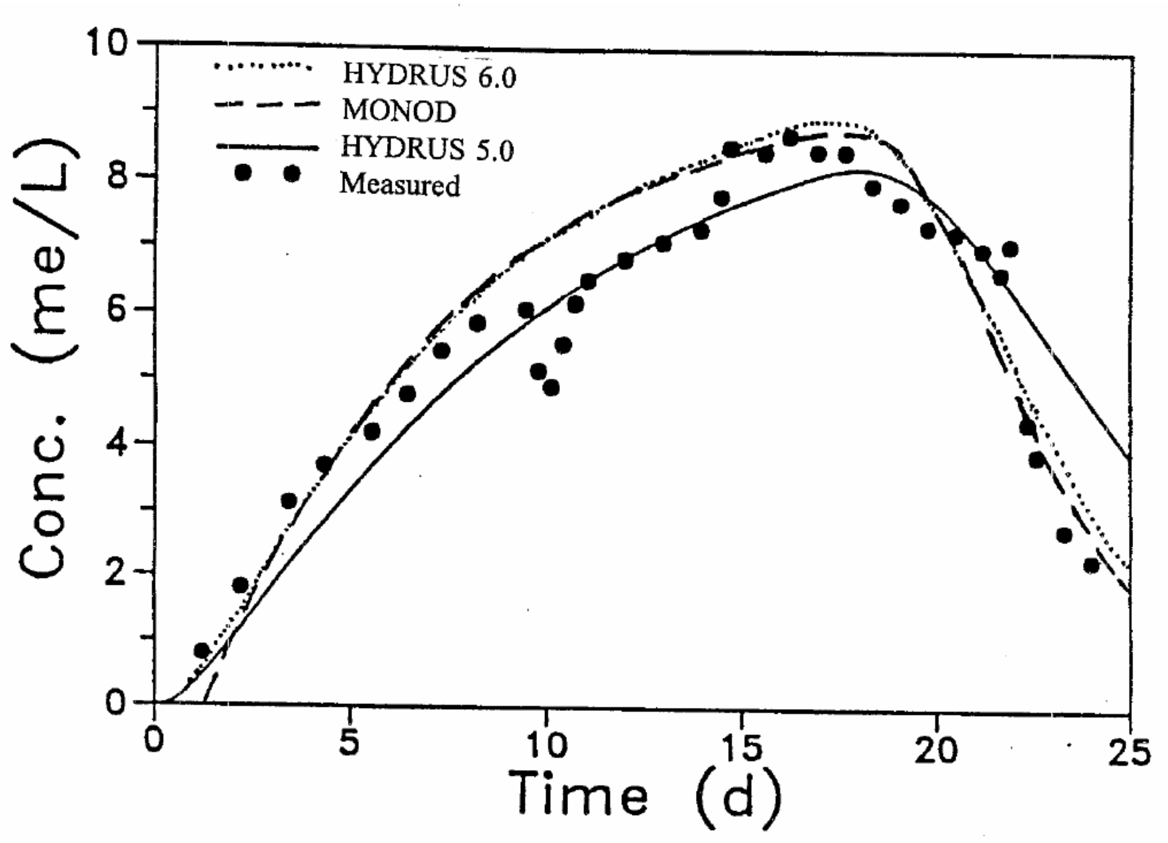


Fig. 9.16. Mg breakthrough curves for Abist loam calculated with the MONOD, HYDRUS, and HYDRUS2 codes (data points from *Selim et al.* [1978]).

The Langmuir adsorption isotherm can also be used to model the exchange of homovalent ions. Parameters in the Langmuir adsorption isotherm for homovalent ion exchange may be derived as follows. Ion exchange for two ions with valences n and m can be expressed in a generalized form as [*Sposito*, 1981]

$$K_{ex} = \left(\frac{\bar{a}_1}{a_1}\right)^m \left(\frac{a_2}{\bar{a}_2}\right)^n \qquad (9.8)$$

where K_{ex} is the dimensionless thermodynamic equilibrium constant, and a and $\bar{a}$ denote the ion activities in the soil solution and on the exchange surfaces [-], respectively:

$$a_i = \gamma_i \, c_i \qquad i = 1, 2$$

$$\overline{a}_i = \xi_i \, s_i \qquad i = 1, 2 \tag{9.9}$$

where c_i [ML^{-3}] (mmol/l) and s_i [MM^{-1}] (mmol/kg) are solution and exchangeable concentrations, respectively, and γ_i and ξ_i are activity coefficients in the soil solution [L^{3}M^{-1}] (l/mmol) and on the exchange surfaces [MM^{-1}] (kg/mmol), respectively. Substituting (9.9) into (9.8) gives

$$K_{12} = K_v \frac{\gamma_1^m}{\gamma_2^n} = K_{ex} \frac{\xi_2^n}{\xi_1^m} \frac{\gamma_1^m}{\gamma_2^n} = \frac{s_1^m}{s_2^n} \frac{c_2^n}{c_1^m} \tag{9.10}$$

where K_v denotes the Vanselow selectivity coefficient [-], while K_{12} will be simply referred to as the selectivity coefficient [-]. Assuming that both the total solution concentration, C_T [ML^{-3}] (mmol$_c$/l), and the cation exchange capacity, S_T [MM^{-1}] (mmol$_c$/kg), are time invariant, i.e.,

$$\begin{aligned} n c_1 + m c_2 &= C_T \\ n s_1 + m s_2 &= S_T \end{aligned} \tag{9.11}$$

the Langmuir parameters k_s and η in (3.3) for the incoming solute become

$$\begin{aligned} k_s &= \frac{K_{12} \, S_T}{C_T} \\ \eta &= \frac{\vartheta(K_{12} - 1)}{C_T} \end{aligned} \tag{9.12}$$

whereas for the solute initially in the soil column:

$$\begin{aligned} k_s &= \frac{S_T}{K_{12} \, C_T} \\ \eta &= \frac{\vartheta(1 - K_{12})}{K_{12} \, C_T} \end{aligned} \tag{9.13}$$

The parameter *υ* in (9.12) and (9.13) equals 1 for monovalent ions, and 2 for divalent ions.

The selectivity coefficient K_{12} for example 5 was measured by *Selim et al.* [1987] (K_{12}=0.51). From the total solution concentration (C_T=10 mmol$_c$/l) and the known cation exchange capacity (S_T=62 mmol$_c$/kg), it follows that the parameters in the Langmuir adsorption isotherm for the incoming solute (Mg) are k_s=3.126 and η=-0.098, while those for the solute initially in the soil profile (Ca) the parameters are k_s=12.157 and η=0.192. The observed Ca breakthrough curve is shown in Figure 9.17, together with the simulated breakthrough curves obtained with the HYDRUS2 and MONOC codes [*Selim et al.*, 1987]. Note the close agreement between the numerical results and the experimental data.

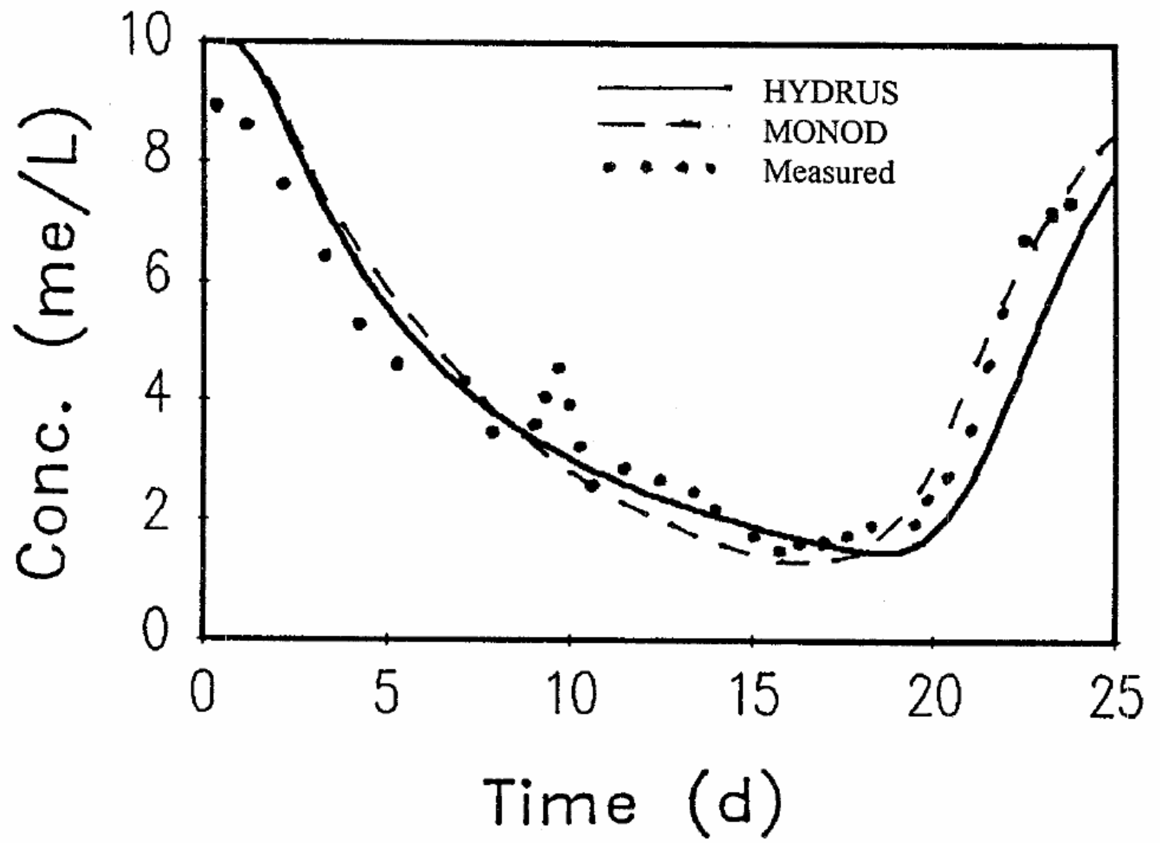


Fig. 9.17. Ca breakthrough curves for Abist loam calculated with the MONOD and HYDRUS2 codes (data points from *Selim et al.* [1978]) (example 5).

9.1.6. *Example 6 - One-Dimensional Solute Transport with Nonequilibrium Adsorption*

This example considers the movement of a boron (H_3BO_4) pulse through Glendale clay loam [*van Genuchten*, 1981]. The numerical simulation uses solute transport parameters that were fitted to the breakthrough curve with the CFITIM parameter estimation model [*van Genuchten*, 1981] assuming a two-site chemical nonequilibrium sorption model analogous to the formulation discussed in Section 3, but for steady-state water flow. Input parameters for example 6 are listed in Table 9.4. Figure 9.18 compares HYDRUS2 numerical results with the experimental data, and with a numerical simulation assuming physical non-equilibrium and nonlinear adsorption [*van Genuchten*, 1981].

Table 9.4. Input parameters for example 6.

Parameter	Value
q [cm/day]	17.12
D [cm^2/day]	49.0
θ [-]	0.445
ρ [g/cm$_3$]	1.222
c_0 [mmol$_c$/L]	20.0
k_s [cm^3/g]	1.14
β [-]	1.0
η [-]	0.0
f [-]	0.47
ω [1/day]	0.320
t_p [day]	5.06

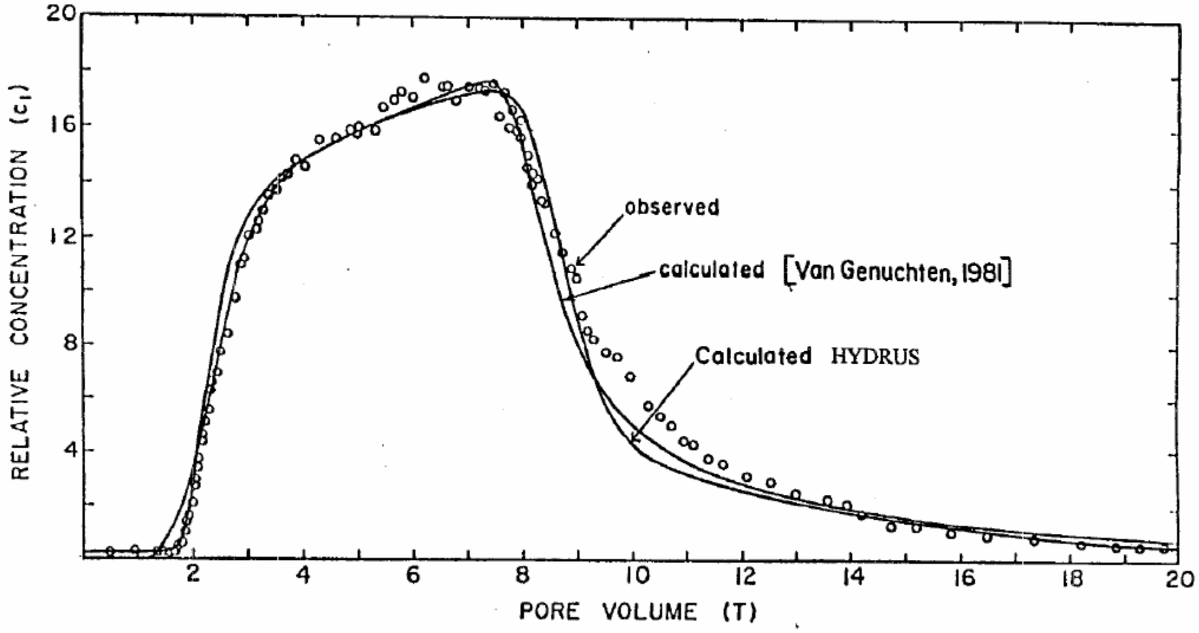
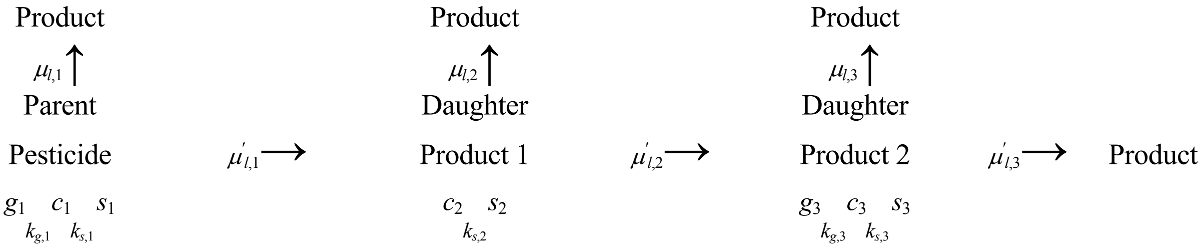


Fig. 9.18. Observed and calculated effluent curves for Boron movement through Glendale clay (data points from *van Genuchten* [1981]) (example 6).

9.1.7. *Example 7 - Water and Solute Infiltration Test*

This example corresponds to example 4 in the SWMS_2D manual [*Šimůnek et al.*, 1992]. The example concerns the movement of water and a dissolved solute from a single-ring infiltrometer into the soil profile consisting of two layers: a 40-cm thick A-horizon, and an underlying B/C-horizon. The hydraulic functions of the two soil layers are the same as those used in Example 2. The axisymmetric flow system and associated finite element mesh for the ponded infiltration experiment are shown in Figure 9.19. The soil profile had an initial temperature of 20°C, whereas the infiltrated water had a temperature of 30°C and contained an organic (parent) compound (the pesticide aldicarb) which is known to degrade by oxidation into two sequential daughter products (sulfone and sulfoxide) [*Ou et al.*, 1988]. Each of the three solutes also undergoes hydrolytical first-order decay that leads to products which are not simulated or monitored during their subsequent transport (i.e., oxime, sulfone oxime, and sulfoxide oxime) [*Ou et al.*, 1988]. All major solutes adsorb onto the solid phase, and volatilization is considered for the first and the third solutes. The reaction pathway is schematically given by:



The example is used here to illustrate variably-saturated water flow, heat movement, and solute transport in a layered and radially symmetric three-dimensional soil profile.

Calculations were carried out over a period of 10 days. The pressure head profile obtained in example problem 2 at the beginning of June 1982 was taken as the initial condition for the water flow equation, similarly as in example 4 of *Šimůnek et al.* [1992]. The soil profile was assumed to be initially free of any solutes. All sides of the flow region were considered to be impervious, except for a small portion around the origin at the surface (the ponded surface inside the ring infiltrometer) where constant pressure head and concentration flux boundary conditions were imposed, as well as the lower right corner where the groundwater level was kept constant. The concentration of the infiltrating water was free of any solute except for the first five days of the simulation during which time the infiltrating water contained the parent solute of unit concentration. Boundary condition

(3.27) was used for the non-ponded part of the soil surface, thereby assuming the existence of a stagnant boundary layer of thickness *d* at the soil surface through which volatile solutes moved to the atmosphere by gas diffusion only. Water and solute extraction by plant roots was not considered.

Tables 9.5, 9.6, and 9.7 list the unsaturated soil hydraulic, heat and solute transport parameters, respectively. The solute and heat transport parameters were assumed to be the same for the two soil layers. The heat transport parameters b_1, b_2, b_3, C_n, C_o, and C_w in Table 9.6 were taken from *Chung and Horton* [1987]. The solute parameters k_s, k_g, μ_w, μ_w', and D_g in Table 9.7 were taken from *Wagenet and Hutson* [1987].

Figure 9.20 presents the initial and steady-state pressure head profiles. The steady- state profile for water flow was reached after approximately 2 or 3 days. Temperature profiles at times 1 and 10 days are shown in Figure 9.21. The heat front moved relatively slowly into the soil profile in comparison with the solute front (shown later) because of the high volumetric heat capacity of the relatively wet soil. Figures 9.22, 9.23, and 9.24 show concentration profiles for all three solutes at times 2.5, 5, 7.5, and 10 days. After its application in the infiltrating water during the first five days of the simulation (Fig. 9.22a, b), the first (parent) solute is transported further by water flow and gaseous diffusion, as well as is being degraded by two first-order decay reactions, such that the soil profile is practically free of this solute after 10 days (Fig. 9.22c, d). The second solute exists exclusively because of first-order degradation of the first solute. Hence, the second solute initially corresponds mainly with the first solute, but subsequently moves faster through the soil profile because of less sorption. Note that the highest concentrations of the second solute were reached after complete application of the first solute (Fig. 9.23c, d). Similar features as for the second solute also apply to the third solute. In particular, notice that the soil profile is almost free of this solute after 2.5 days (Fig. 9.24a), and that the highest concentrations of the third solute (while being much smaller than those for the first two solutes) were reached at the end of the simulation (Fig. 9.24d).

Table 9.5. Hydraulic input parameters for example 7.

Parameter	1st layer	2nd layer
$\theta_s=\theta_m=\theta_k$	0.399	0.339
$\theta_r=\theta_a$	0.000	0.000
$K_s=K_k$ [m/day]	0.298	0.454
α [1/m]	1.74	1.39
n [-]	1.38	1.60

Table 9.6. Heat transport input parameters for example 7.

Parameter	Value
θ_n [-]	0.600^+ (0.660^*)
θ_o [-]	0.001
λ_L [m]	0.005
λ_T [m]	0.001
b_1 [$Wm^{-1}K^{-1}$]	0.243
b_2 [$Wm^{-1}K^{-1}$]	0.393
b_3 [$Wm^{-1}K^{-1}$]	1.534
C_n [$Jm^{-3}K^{-1}$]	$1.92*10^6$
C_o [$Jm^{-3}K^{-1}$]	$2.51*10^6$
C_w [$Jm^{-3}K^{-1}$]	$4.18*10^6$
T_i [°C]	20
T_0 [°C]	30

$^+$ for the first layer

* for the second layer

Table 9.7. Solute transport input parameters for example 7.

Parameter	Value
ρ [kg/m^3]	1300
D_w [m^2/day]	0.00374
D_g [m^2/day]	0.432
D_L [m]	0.005
D_T [m]	0.001
$k_{s,1}$ [m^3/kg]	0.0001
$k_{s,2}$ [m^3/kg]	0.00005
$k_{s,3}$ [m^3/kg]	0.0002
$k_{g,1}$ [-]	$1.33*10^{-7}$
$k_{g,2}$ [-]	0.0
$k_{g,3}$ [-]	$1.33*10^{-3}$
$\mu_{w,1}$ [1/day]	0.2
$\mu_{w,2}$ [1/day]	0.01
$\mu_{w,3}$ [1/day]	0.005
$\mu'_{w,1}$ [1/day]	0.36
$\mu'_{w,2}$ [1/day]	0.024
$\mu'_{w,3}$ [1/day]	0.0024
$c_{0,1}$ [-]	1.0
$c_{0,2}$ [-]	0.0
$c_{0,3}$ [-]	0.0
d [m]	0.005

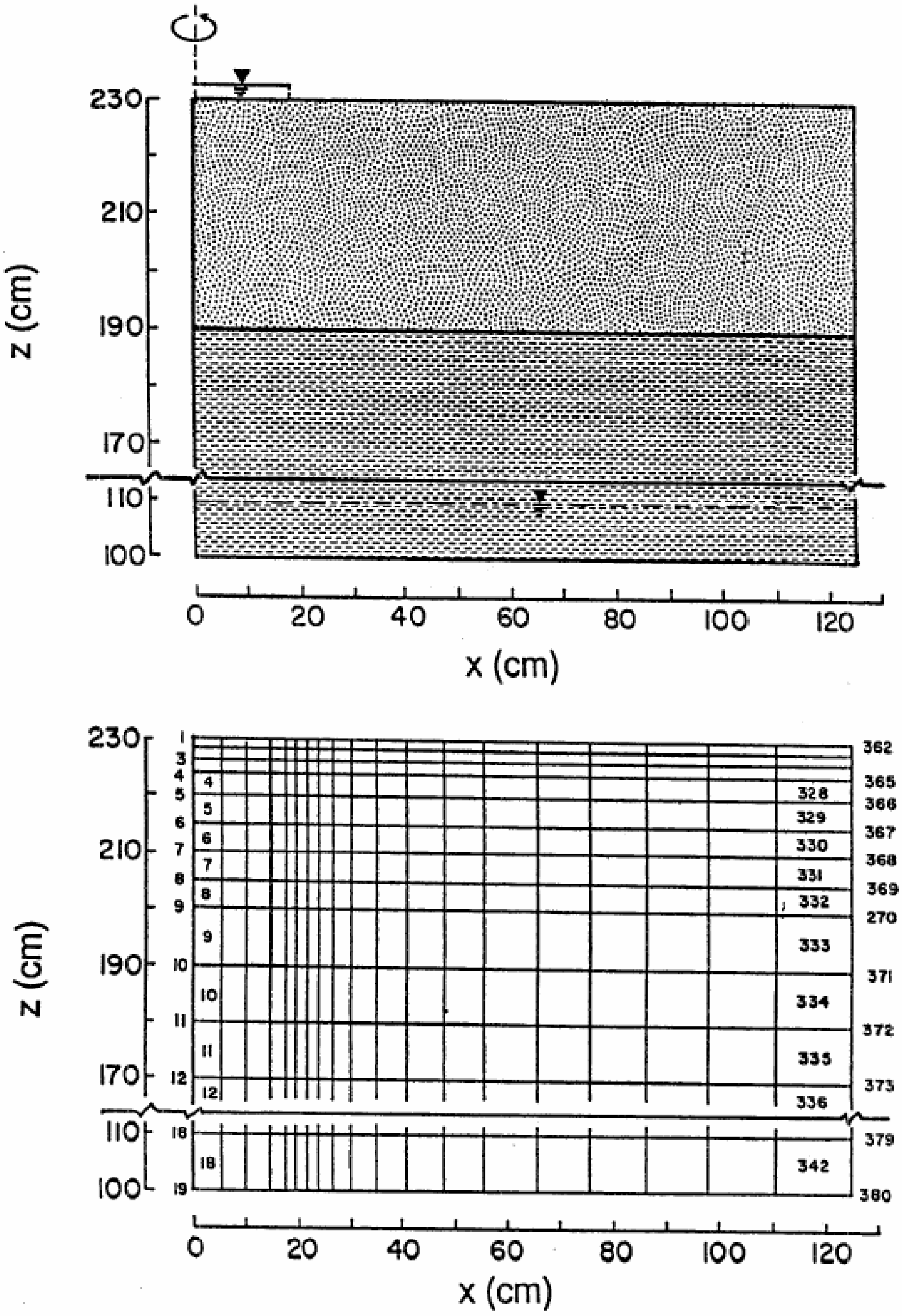


Fig. 9.19. Flow system and finite element mesh for example 7.

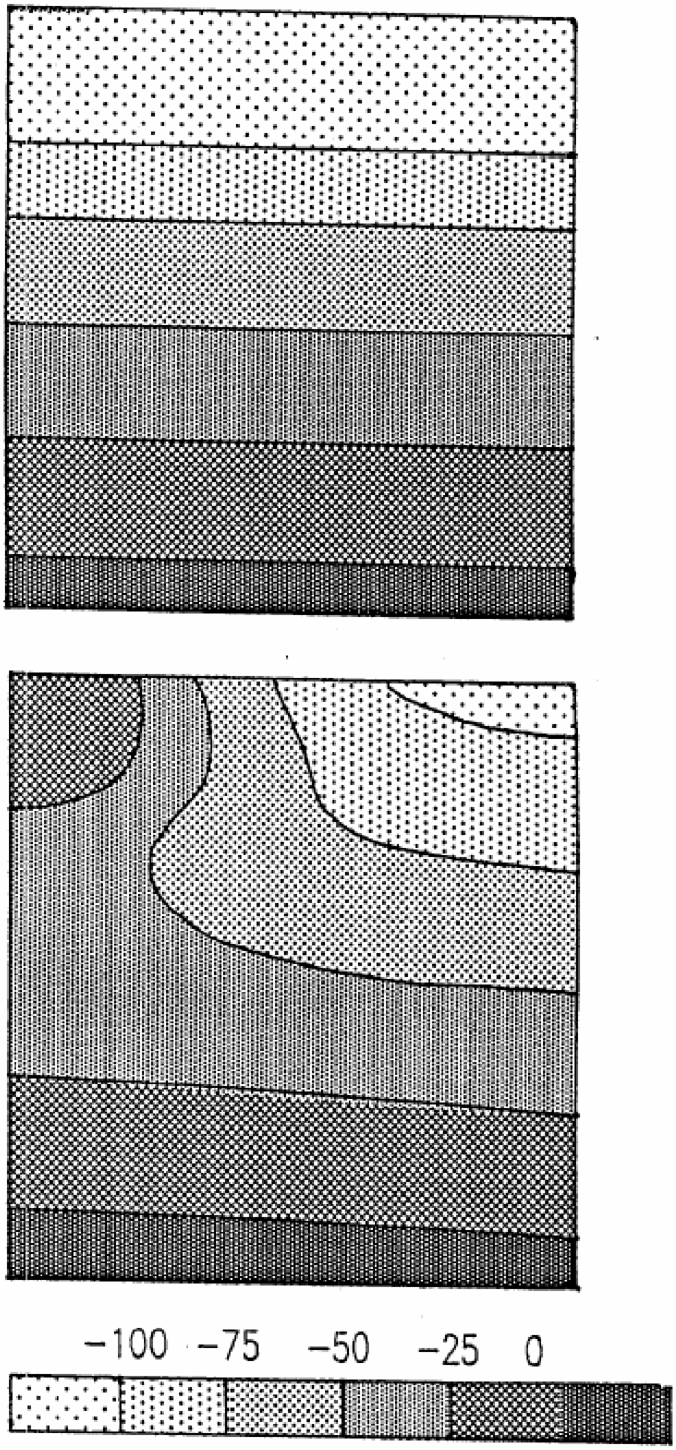


Fig. 9.20. Initial (top) and steady state (bottom) pressure head profiles for example 7.

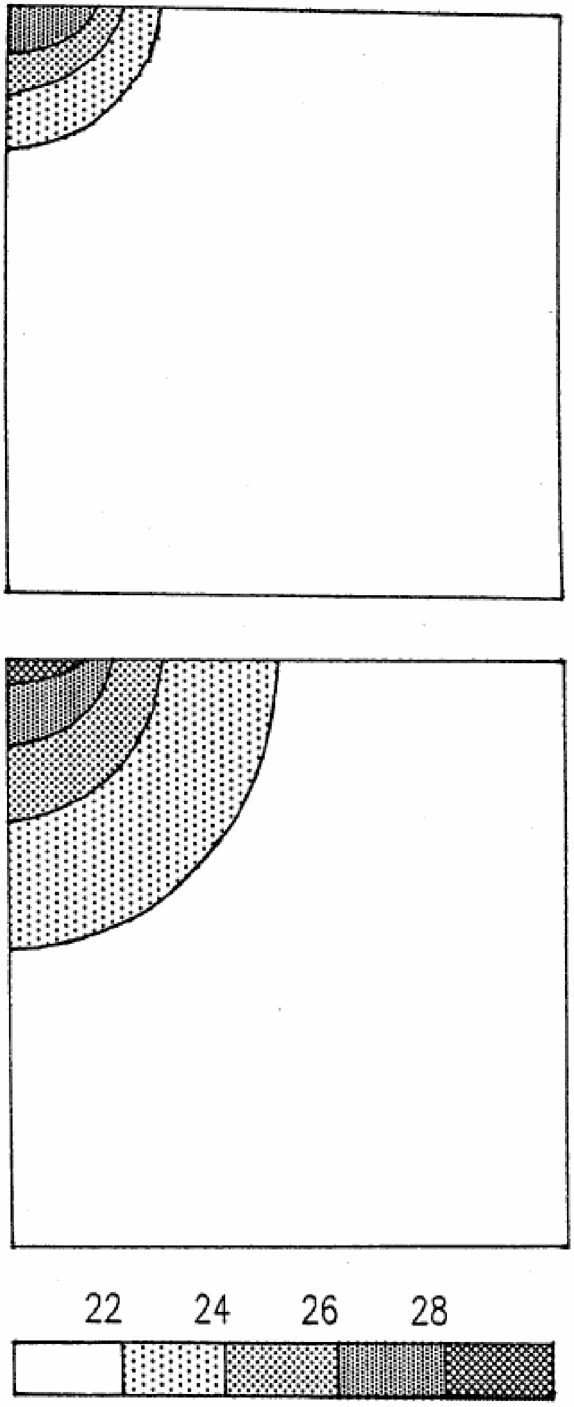


Fig. 9.21. Temperature profiles after 1 (top) and 10 days (bottom) for example 7.

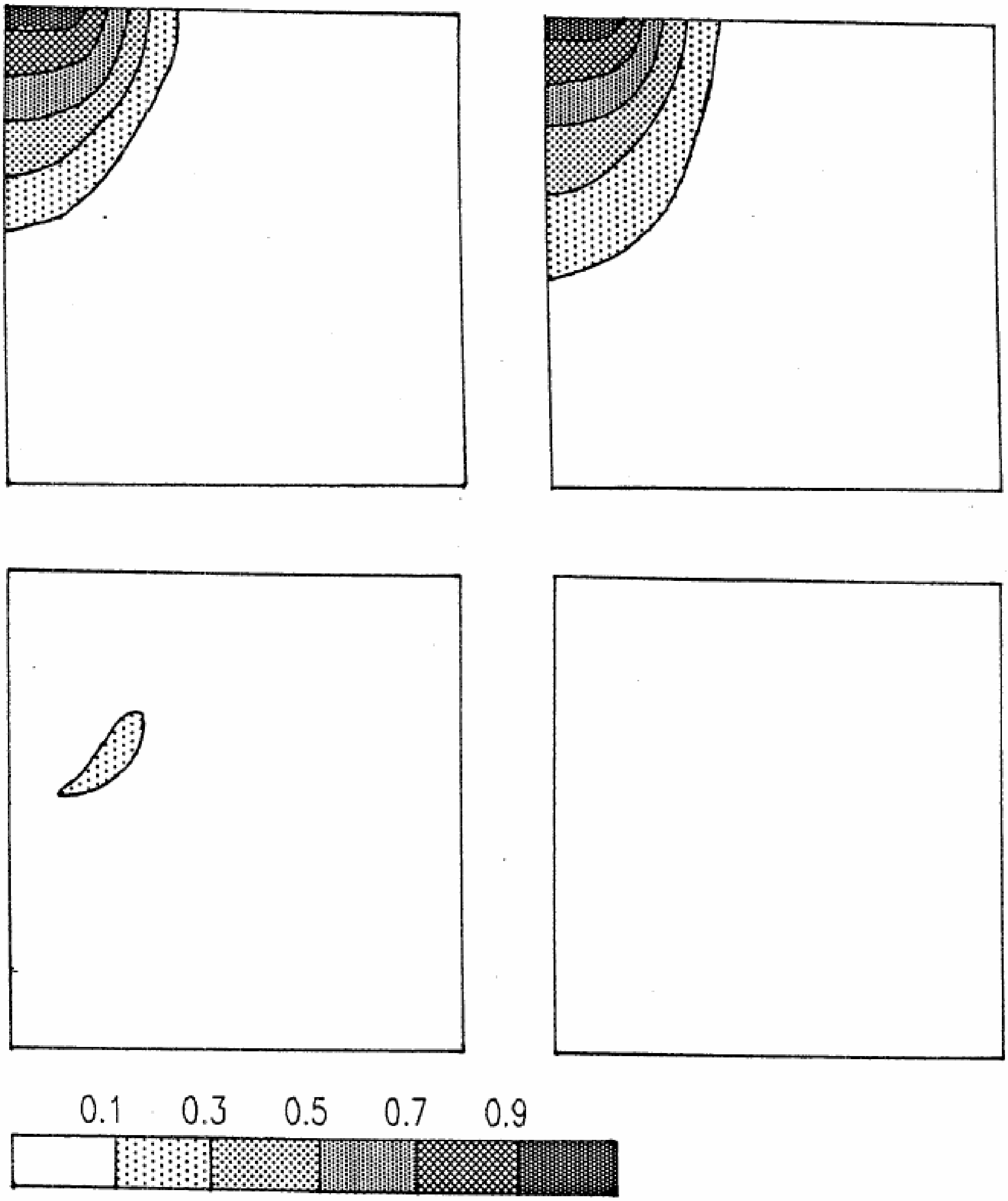


Fig. 9.22. Concentration profiles for the first solute after 2.5, 5, 7.5, and 10 days for example 7.

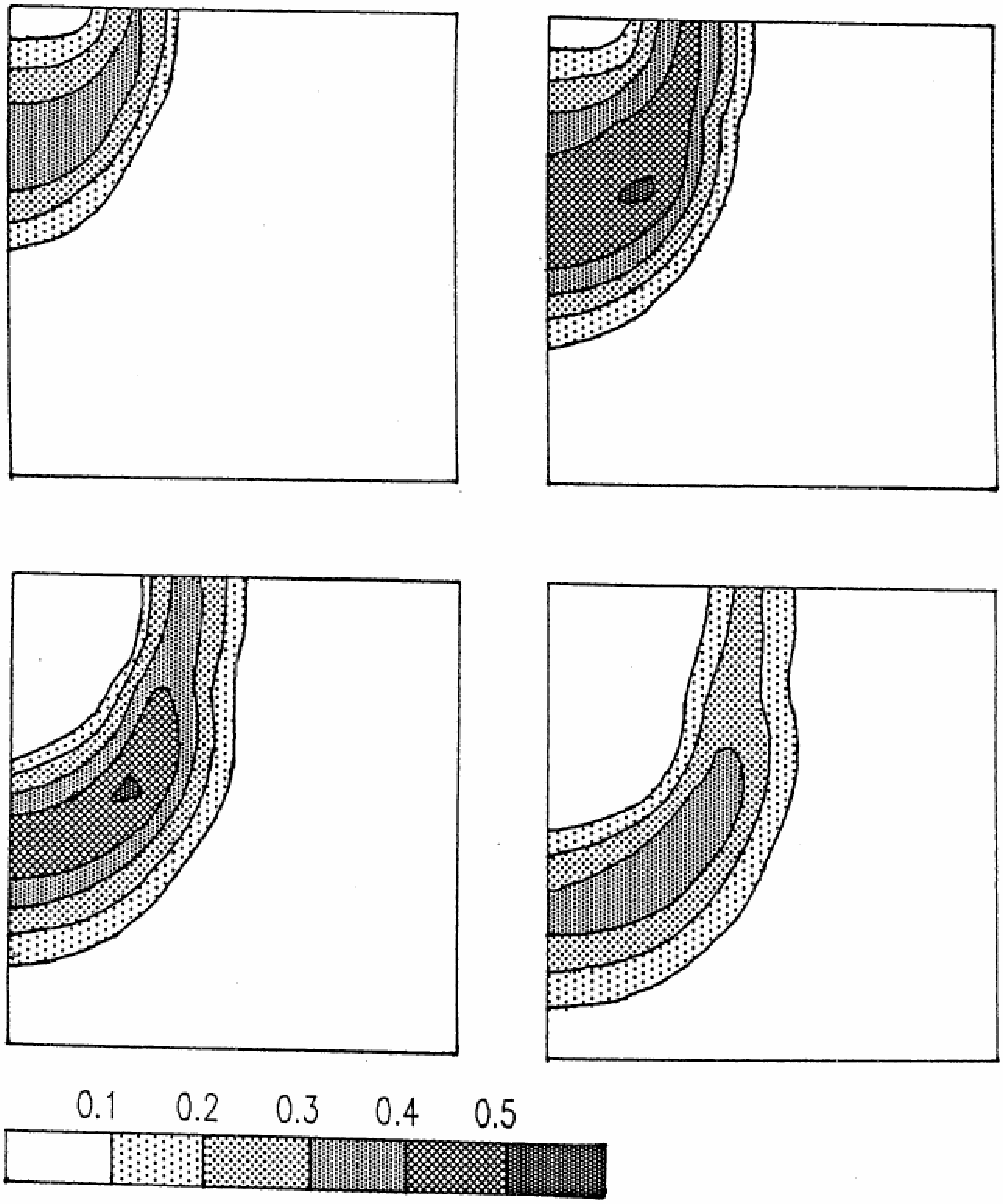


Fig. 9.23. Concentration profiles for the second solute after 2.5, 5, 7.5, and 10 days for example 7.

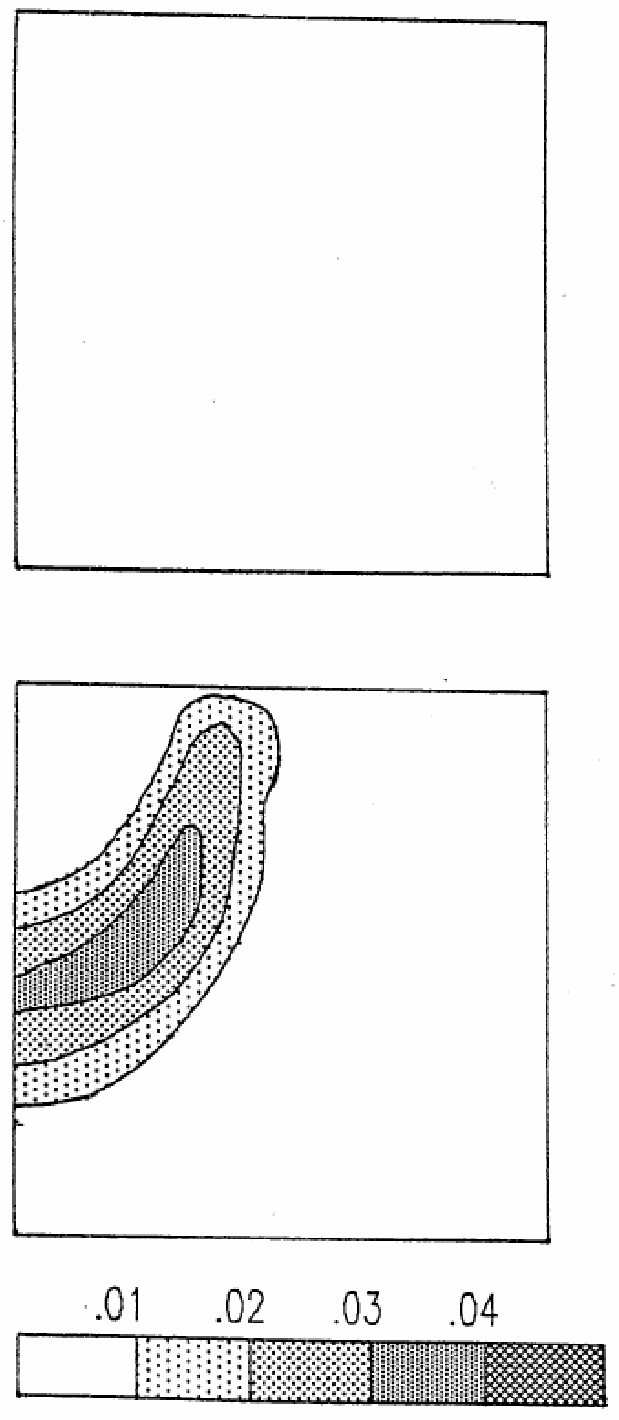


Fig. 9.24. Concentration profiles for the third solute after 2.5, 5, 7.5, and 10 days for example 7.

9.2. *Inverse Example Problems*

In this section we discuss three methods recently proposed for estimating the soil hydraulic properties by numerical inversion of the Richards' equation. One method currently being developed involves the use of tension disc permeameter data [*Šimůnek and van Genuchten*, 1996, 1997] (example 8), while a second method uses data collected with a modified cone penetrometer [*Gribb*, 1996; *Gribb et al.*, 1998] (example 9). The third method involves the use of a multiple extraction device [*Inoue et al.*, 1998] (example 10).

9.2.1. *Example 8 - Tension Disc Infiltrometer*

Šimůnek and van Genuchten [1997] suggested the use of multiple tension infiltration experiments in combination with knowledge of the initial and final water contents for estimating soil hydraulic properties. An evaluation of the numerical stability and parameter uniqueness using numerically generated data with superimposed stochastic and deterministic errors showed that a combination of the multiple cumulative tension infiltration data, a measured final water content, and an initial condition expressed in terms of the water content, provided the most promising parameter estimation approach for practical applications [*Šimůnek and van Genuchten*, 1997].

The experiment was used to estimate the soil hydraulic characteristics of a two-layered soil system involving a crusted soil in a Sahel region [*Šimůnek et al.*, 1998]. Here we will report only results for the sandy subsoil. Data were obtained with a tension disc diameter of 25 cm and with supply tensions of 11.5, 9, 6, 3, 1, and 0.1 cm. Figure 9.25 shows measured and optimized cumulative infiltration curves and their differences. The small breaks in the cumulative infiltration curve (Fig. 9.25) were caused by brief removal of the infiltrometer from the soil surface to resupply it with water and to adjust the tension for a new time interval. Very close agreement between the measured and optimized cumulative infiltration curves was obtained; the largest deviations were generally less than 60 ml, which constituted only about 0.5% of the total infiltration volume. Figure 9.26 shows a comparison of the parameter estimation results against results obtained using Wooding's analysis. Both methods give almost identical unsaturated hydraulic conductivities for pressure heads between -2 and -10.25 cm. However, the hydraulic conductivity at the highest pressure head interval was overestimated by a factor of two using Wooding's analysis. *Šimůnek et al.* [1998] further compared the numerical inversion results with hydraulic properties estimated from available soil textural information using a neural-network-based pedotransfer function approach.

Relatively good agreement between the inverse and neural network predictions was obtained.

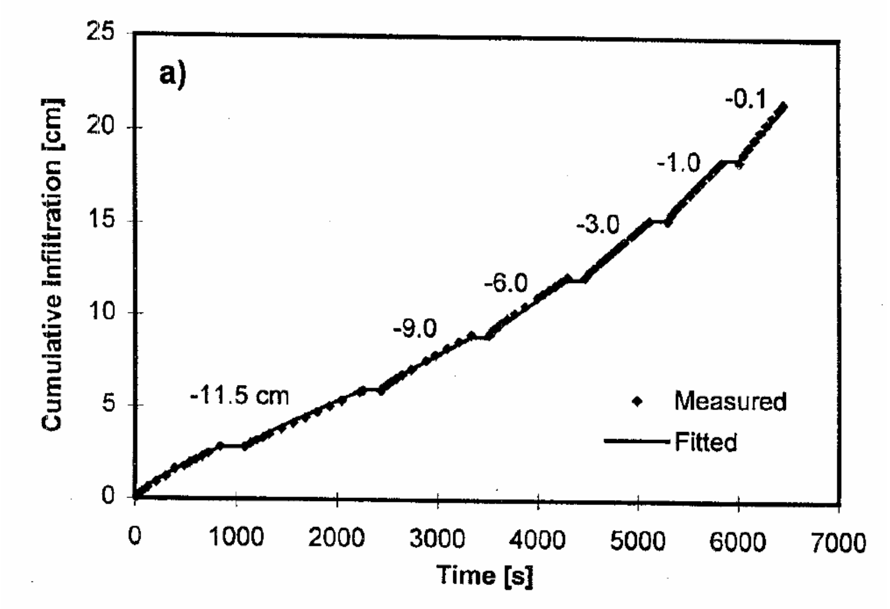


Fig. 9.25. Measured and optimized cumulative infiltration curves for a tension disc infiltrometer experiment (example 8).

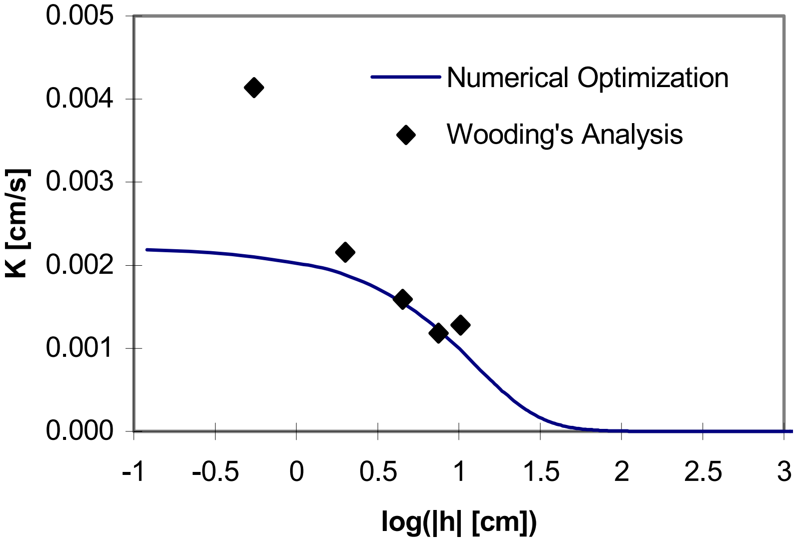


Fig. 9.26. Unsaturated hydraulic conductivities at particular pressure heads, calculated using Wooding's analytical solution and the complete function obtained by numerical inversion (example 8).

9.2.2. *Example 9 - Cone Penetrometer*

While tension infiltrometer experiments provide relatively quick estimates of the hydraulic properties, they can be used only at the soil surface. By comparison, a new cone penetrometer method currently under development [*Gribb*, 1996; *Gribb et al.*, 1998; *Kodešová et al.*, 1998] can be used at depth. Cone penetrometers were originally used to obtain soil strength characteristics by measuring the tip resistance and sleeve friction during penetration at a constant rate. To obtain the hydraulic properties, a modified cone penetrometer, instrumented with a porous filter close to the penetrometer tip and two tensiometer rings 5 and 9 cm above the filter, is used (Fig. 9.27). The device is pushed into a soil to the desired depth, and a constant head is applied to the 5-cm filter. The volume of water imbibed into the soil is monitored, as are tensiometer ring readings registering the advancement of the wetting front for a short period of time (300-500 s).

Gribb [1996] gave a detailed numerical analysis of this experiment, including a study of the identifiability of the soil hydraulic parameters. She showed that the inverse solution was least sensitive to n and θ_s, and most sensitive to K_s and α. The method was recently used to estimate the hydraulic parameters of a sandy soil in a laboratory aquifer system measuring 5 x 5 x 3 m [*Gribb et al.*, 1998; *Kodešová et al.*, 1998; *Šimůnek et al.*, 1999].

Figure 9.28 shows observed flow data, as well as results of the numerical inversion. Excellent agreement between measured and optimized values was obtained for the inverse solution with four optimized parameters. *Gribb et al.* [1998] and *Kodešová et al.* [1998] presented retention curves obtained with selected laboratory methods and the parameter estimation technique. The optimized curves were close to wetting curves determined in the laboratory. *Gribb et al.* [1998] showed that the estimated saturated hydraulic conductivities were similar to those obtained with other test methods, such as the Guelph permeameter, slug tests, and laboratory constant head tests. Finally, *Šimůnek et al.* [1999] further used the redistribution part of the experiment to also estimate hysteresis in the soil water retention curve.

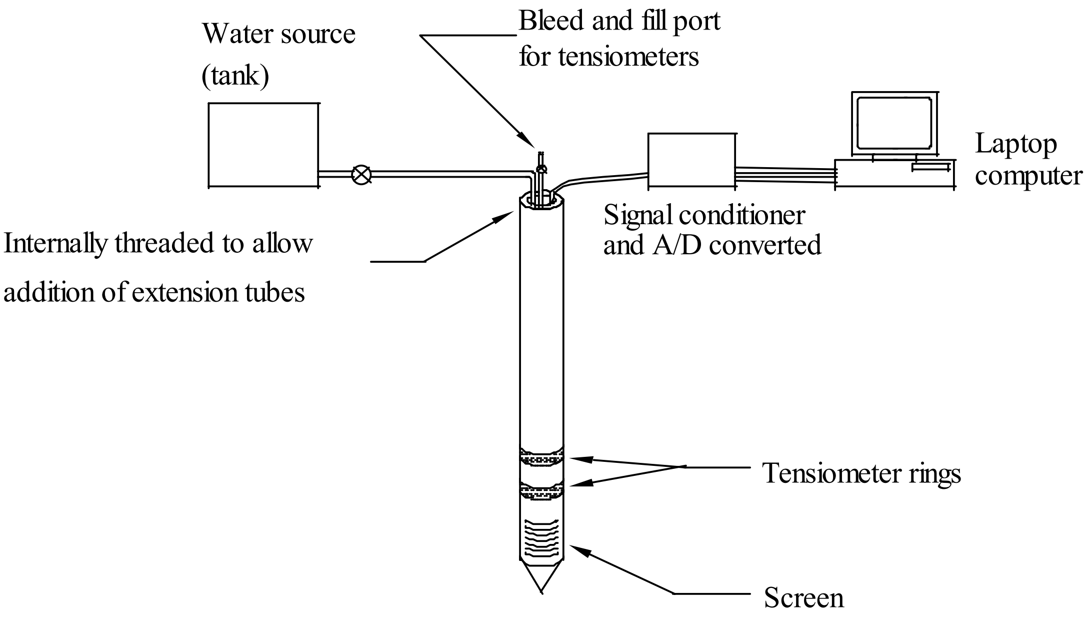


Fig. 9.27. Schematic of the modified cone penetrometer (example 9).

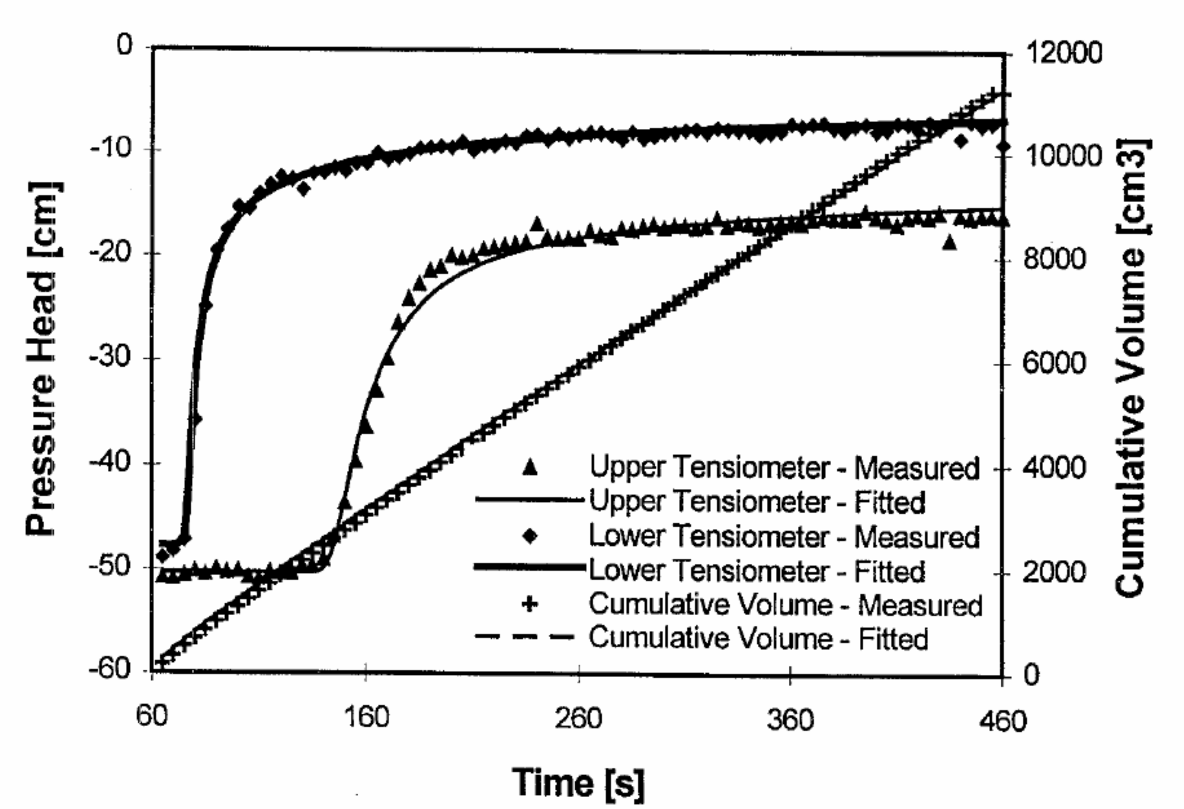


Fig. 9.28. Comparison of observed and optimized cumulative infiltration curves and tensiometer readings for the modified cone penetrometer test (example 9).

9.2.3. *Example 10 - Multiple-Step Extraction Experiment*

The tension infiltrometer and cone penetrometer methods provide information about the wetting branches of the soil hydraulic properties. By comparison, a multiple step extraction device can be used to obtain draining branches. The device consists of a ceramic soil solution sampler (Fig. 9.29), which is inserted into an initially wet soil profile and subjected to a series of vacuum extraction pressures. The cumulative amount of soil solution extracted during an experiment, as well as pressure heads at various locations near the extraction device, are monitored during the experiment and subsequently used in an objective function for the nonlinear minimization problem. *Inoue et al.* [1998] first evaluated the feasibility of the vacuum extraction technique using numerically generated data, and concluded that the method is well suited for loamy-textured soils, but not necessarily for sandy soils. They tested the method in the laboratory and in the field. Here we briefly discuss one laboratory application.

The experiment [*Inoue et al.*, 1998] was carried out on a Columbia fine sandy loam. The center of the ceramic ring (with a radius of 3 cm, and a length of 3 cm) was located 6.3 cm below the soil surface. The tensiometers were installed at the following positions: $T_1(r,z)$=(4,-6.3 cm), T_2=(6,-6.3), T_3=(6,-24). Five vacuum extraction steps were applied: h_{ex}=-35 cm for 25 h, h_{ex}=-65 cm for 25 < t < 73 h, h_{ex}=-125 cm for 73 < t < 217 h, h_{ex}=-240 cm for 217 < t < 339 h, and finally h_{ex}=-480 cm for 339 < t < 605 h. Experimental data, as well as the final results of the numerical inversion, are presented in Figure 9.30. The saturated hydraulic conductivity of the ceramic ring was optimized simultaneously with van Genuchten's hydraulic parameters. Agreement between measured and calculated values in Figure 9.30 was relatively good. *Inoue et al.* [1998] compared the optimized soil hydraulic functions with independently measured data.

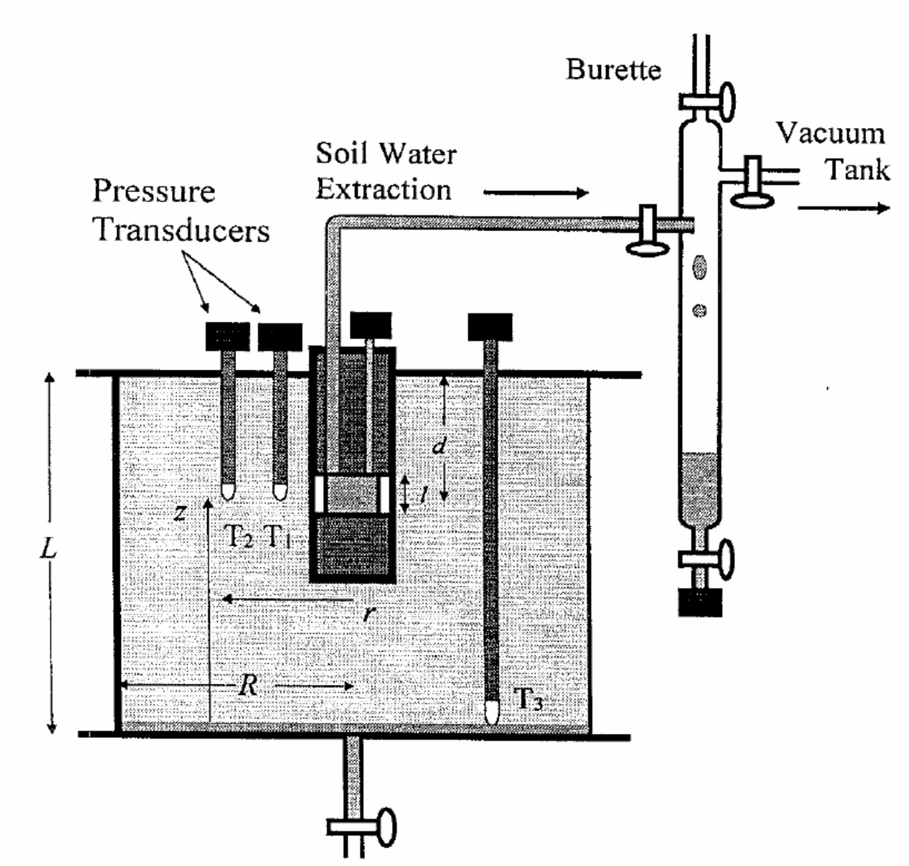


Fig. 9.29. Layout of laboratory multistep extraction experiment (example 10).

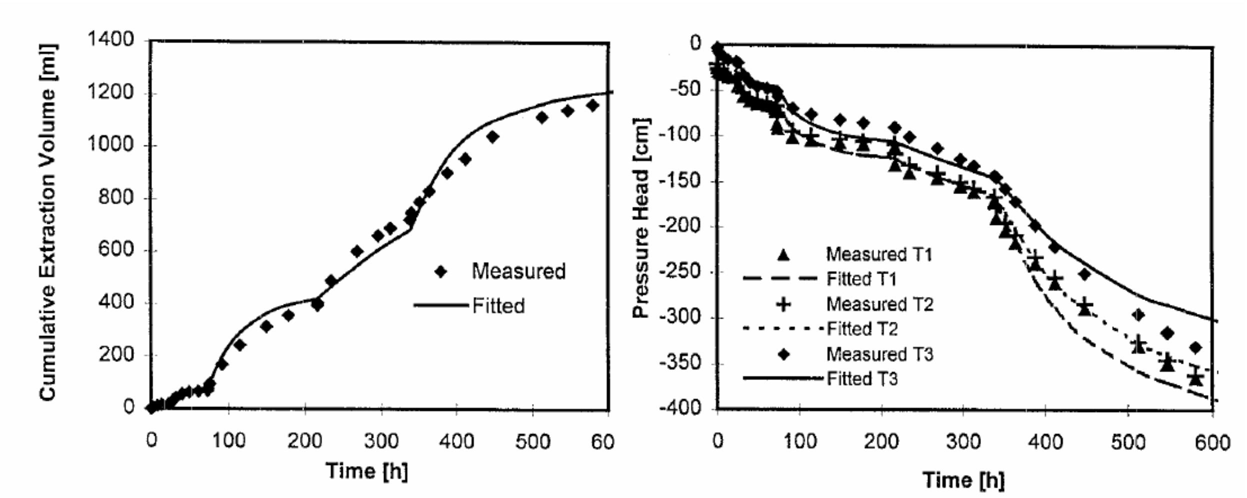


Fig. 9.30. Comparison of measured (symbols) and optimized (lines) cumulative extraction (a) and pressure head (b) values (example 10).

10. INPUT DATA

The input data for HYDRUS2 are given in seven separate input files. These input files consist of one or more input blocks identified by the letters from A through M. The input files and blocks must be arranged as follows:

SELECTOR.IN

A. Basic Information
B. Material Information
C. Time Information
D. Root Water Uptake Information
E. Solute Transport Information
F. Heat Transport Information

MESHTRIA.TXT

G. Finite Element Mesh Information

DOMAIN.DAT

H. Nodal Information
I. Element Information

BOUNDARY.IN

J. Boundary Information

ATMOSPH.IN

K. Atmospheric Information

DIMENSIO.IN

L. Dimension Information

FIT.IN

M. Inverse Solution Information

The various input blocks are described in detail below. Tables 10.1 through 10.13 describe the data required for each input block. All data are read in using list-directed formatting (free format). Comment lines are provided at the beginning of, and within, each input block to facilitate, among other things, proper identification of the function of the block and the input variables. The comment lines are ignored during program execution; hence, they may be left blank but should not be omitted. All input files must be placed into one subdirectory. Output files are printed into the same subdirectory. An additional file **Level_01.dir**, which specifies the path to the input and output file subdirectory must be given in the same directory as the executable **HYDRUS2** code. The program assumes that all input data are specified in a consistent set of units for mass M, length L, and time T. The values of temperature should be specified in degrees Celsius.

Most of the information in Tables 10.1 through 10.13 should be self-explanatory. Several input files can be alternatively entered using binary files, e.g. the DOMAIN.DAT file can be replaced with the binary DOMAIN.IN file, and the MESHTRIA.TXT file can be replaced with the MESHTRIA.000 (or MESHGEN2.PMG) binary file. Binary files are utilized by default when the HYDRUS2D user interface is used.

Table 10.1. Block A - Basic information.

Record	Type	Variable	Description
1	Char	*cVersion*	Set text equal to "Pcp_File_Version=".
1	Integer	*iVersion*	Version of the Selector.in file; set equal to 2 for current version.
2,3	-	-	Comment lines.
4	Char	*Hed*	Heading.
5	-	-	Comment line.
6	Char	*LUnit*	Length unit (e.g., 'cm').
7	Char	*TUnit*	Time unit (e.g., 'min').
8	Char	*MUnit*	Mass unit for concentration (e.g., 'g', 'mol', '-').
9	-	-	Comment line.
10	Integer	*Kat*	Type of flow system to be analyzed: 0 for a horizontal (areal) system 1 for axisymmetric flow 2 for vertical flow in a cross-section
11	-	-	Comment line.
12	Integer	*MaxIt*	Maximum number of iterations allowed during any time step (usually 20).
12	Real	*TolTh*	Absolute water content tolerance for nodes in the unsaturated part of the flow region [-] (its recommended value is 0.0001). *TolTh* represents the maximum desired absolute change in the value of the water content, θ, between two successive iterations during a particular time step.
12	Real	*TolH*	Absolute pressure head tolerance for nodes in the saturated part of the flow region [L] (its recommended value is 0.1 cm). *TolH* represents the maximum desired absolute change in the value of the pressure head, *h*, between two successive iterations during a particular time step.
12	Logical	*lInitW*	Set this logical variable equal to **.true.** when the initial condition is specified in terms of the pressure head. Set this logical variable equal to **.false.** when the initial condition is specified in terms of the water content.
13	-	-	Comment line.
14	Logical	*lWat*	Set this logical variable equal to **.true.** when transient water flow is considered. Set this logical variable equal to **.false.** when steady-state water flow is to be calculated.
14	Logical	*lChem*	Set this logical variable equal to **.true.** if solute transport is to be considered.
14	Logical	*SinkF*	Set this variable equal to **.true.** if water extraction from the root zone is imposed.

Table 10.1. (continued)

Record	Type	Variable	Description
14	Logical	*ShortF*	**.true.** if information is to be printed only at preselected times, but not at each time step (T-level information, see Section 11.1), **.false.** if information is to be printed at each time step.
14	Logical	*FluxF*	**.true.** if detailed information about the element fluxes and discharge/recharge rates is to be printed.
14	Logical	*lScreen*	Set this logical variable equal to **.true.** if information is to be printed on the screen during code execution.
14	Logical	*AtmInf*	**.true.** if atmospheric boundary conditions are supplied via the input file ATMOSPH.IN, **.false.** if the file ATMOSPH.IN is not provided (i.e., in case of time independent boundary conditions).
14	Logical	*lTemp*	Set this logical variable equal to **.true.** if heat transport is to be considered.
14	Logical	*lWDep*	**.true.** if hydraulic properties are to be considered as temperature dependent. **.false.** otherwise (see Section 2.5).
14	Logical	*lEquil*	**.true.** if equilibrium solute transport is considered. **.false.** if nonequilibrium solute transport is considered for at least one solute species.
14	Logical	*lExter*	Set this logical variable equal to **.true.** if an external mesh generator is to be used to generate the finite element mesh (i.e., when using MESHGEN-2D). Set this logical variable equal to **.false.** if an internal mesh generator for simple quadrilateral domains is to be used to generate the finite element mesh.
14	Logical	*lInv*	Set this logical variable equal to **.true.** if the soil hydraulic or solute transport parameters are to be estimated from measured data. Set this logical variable equal to **.false.** if only the direct solution for a particular problem is to be carried out.

Table 10.2. Block B - Material information.

Record	Type	Variable	Description
1,2	-	-	Comment lines.
3	Integer	*NMat*	Number of soil materials. Materials are identified by the material number, *MatNum*, specified in Block H.
3	Integer	*NLay*	Number of subregions for which separate water balances are being computed. Subregions are identified by the subregion number, *LayNum*, specified in Block I.
3	Real	*ha*	Absolute value of the upper limit [L] of the pressure head interval below which a table of hydraulic properties will be generated internally for each material (h_a must be greater than 0.0; e.g. 0.001 cm) (see Section 5.3.11).
3	Real	*hb*	Absolute value of the lower limit [L] of the pressure head interval for which a table of hydraulic properties will be generated internally for each material (e.g. 1000 m). One may assign to h_b the highest (absolute) expected pressure head to be expected during a simulation. If the absolute value of the pressure head during program execution lies outside of the interval [h_a ,h_b], then appropriate values for the hydraulic properties are computed directly from the hydraulic functions (i.e., without interpolation in the table).
4	-	-	Comment line.
5	Integer	*iModel*	Soil hydraulic properties model: = 0; *van Genuchten*'s [1980] model with five parameters. = 1; modified van Genuchten's model with nine parameters, *Vogel and Císlerová* [1988]. = 2; *Brooks and Corey*'s [1964] model with five parameters.
5	Integer	*iHyst*	Hysteresis in the soil hydraulic properties: = 0; No hysteresis = 1; Hysteresis in the retention curve only = 2; Hysteresis in both the retention and hydraulic conductivity functions
6	-	-	Comment line.
7	Integer	*iKappa*	= -1 if the initial condition is to be calculated from the main drying branch. = 1 if the initial condition is to be calculated from the main wetting branch.
			Records 6 and 7 are provided only when *iHyst* > 0.
8	-	-	Comment line.
9	Real	*Par*(1,*M*)	Parameter θ_r for material *M* [-].
9	Real	*Par*(2,*M*)	Parameter θ_s for material *M* [-].
9	Real	*Par*(3,*M*)	Parameter α for material *M* [L^{-1}].
9	Real	*Par*(4,*M*)	Parameter *n* for material *M* [-].
9	Real	*Par*(5,*M*)	Parameter K_s for material *M* [LT^{-1}].
9	Real	*Par*(6,*M*)	Parameter *l* for material *M* [-].

Table 10.2. (continued)

Record	Type	Variable	Description
			The following four parameters are specified only when *iModel*=1.
9	Real	*Par*(7,*M*)	Parameter θ_m for material *M* [-].
9	Real	*Par*(8,*M*)	Parameter θ_a for material *M* [-].
9	Real	*Par*(9,*M*)	Parameter θ_k for material *M* [-].
9	Real	*Par*(10,*M*)	Parameter K_k for material *M* [LT^{-1}].
			The following four parameters are specified only when *iModel*=0 and *iHyst*>1.
9	Real	*Par*(7,*M*)	Parameter θ_m for material *M* [-].
9	Real	*Par*(8,*M*)	Parameter θ_s^w for material *M* [-].
9	Real	*Par*(9,*M*)	Parameter α^w for material *M* [L^{-1}].
9	Real	*Par*(10,*M*)	Parameter K_s^w for material *M* [LT^{-1}].
			Record 9 information is provided for each material *M* (from 1 to *NMat*). If *lWDep*=**.true.** (Block A) then the soil hydraulic parameters *Par*(*i,M*) must be specified at reference temperature T_{ref}=20°C.

Table 10.3. Block C - Time information.

Record	Type	Variable	Description
1,2	-	-	Comment lines.
3	Real	*dt*	Initial time increment, Δt [T]. Initial time step should be estimated in dependence on the problem being solved. For problems with high pressure gradients (e.g. infiltration into an initially dry soil), Δt should be relatively small.
3	Real	*dtMin*	Minimum permitted time increment, Δt_{min} [T].
3	Real	*dtMax*	Maximum permitted time increment, Δt_{max} [T].
3	Real	*dMul*	If the number of required iterations at a particular time step is less than or equal to *ItMin*, then Δt for the next time step is multiplied by a dimensionless number *dMul* ≥1.0 (its value is recommended not to exceed 1.3).
3	Real	*dMul*2	If the number of required iterations at a particular time step is greater than or equal to *ItMax*, then Δt for the next time step is multiplied by *dMul*2≤1.0 (e.g. 0.33).
3	Integer	*ItMin*	If the number of required iterations at a particular time step is less than or equal to *ItMin*, then Δt for the next time step is multiplied by a dimensionless number *dMul*≥ 1.0 (its value is recommended not to exceed 1.3).
3	Integer	*ItMax*	If the number of required iterations at a particular time step is greater than or equal to *ItMax*, then Δt for the next time step is multiplied by *dMul*2 ≤1.0 (e.g. 0.33).
3	Integer	*MPL*	Number of specified print-times at which detailed information about the pressure head, water content, flux, temperature, concentrations, and the water and solute balances will be printed.
4	-	-	Comment line.
5	Real	*tInit*	Initial time of the simulation [T].
5	Real	*tMax*	Final time of the simulation [T].
6	-	-	Comment line.
7	Real	*TPrint*(1)	First specified print-time [T].
7	Real	*TPrint*(2)	Second specified print-time [T].
.	.	.	.
.	.	.	.
7	Real	*TPrint*(*MPL*)	Last specified print-time [T]. (Maximum six values on one line.)

Table 10.4. Block D - Root water uptake information.+

Record	Type	Variable	Description
1,2	-	-	Comment lines.
3	Integer	*iMoSink*	Type of root water uptake stress response function. = 0; *Feddes et al.* [1978] = 1; S-shaped, *van Genuchten* [1987]
4	-	-	Comment line.
			The following records (records 5a, 6a, 7a) are given only if *iMoSink*=0.
5a	Real	*P0*	Value of the pressure head, h_1 (Fig. 2.1), below which roots start to extract water from the soil.
5a	Real	*P2H*	Value of the limiting pressure head, h_3, below which the roots cannot extract water at the maximum rate (assuming a potential transpiration rate of *r2H*).
5a	Real	*P2L*	As above, but for a potential transpiration rate of *r2L*.
5a	Real	*P3*	Value of the pressure head, h_4, below which root water uptake ceases (usually equal to the wilting point).
5a	Real	*r2H*	Potential transpiration rate [LT^{-1}] (currently set at 0.5 cm/day).
5a	Real	*r2L*	Potential transpiration rate [LT^{-1}] (currently set at 0.1 cm/day). The above input parameters permit one to make the variable h_3 a function of the potential transpiration rate, T_p (h_3 presumably decreases at higher transpiration rates). HYDRUS2 currently implements the same linear interpolation scheme as used in several versions of the SWATRE code (e.g., *Wesseling and Brandyk* [1985]) and in the SWMS_2D [*Šimůnek et al.*, 1992] and HYDRUS 5.0 [*Vogel et al.*, 1996] codes. The scheme is based on the following interpolation: $h_3 = P2H + \frac{P2L - P2H}{r2H - r2L}(r2H - T_p) \quad \text{for} \quad r2L < T_p < r2H$ $h_3 = P2L \quad \text{for} \; T_a \leq r2L$ $h_3 = P2H \quad \text{for} \; T_a \geq r2H$
6a	-	-	Comment line.
7a	Real	*POptm*(1)	Value of the pressure head, h_2, below which roots start to extract water at the maximum possible rate (material number 1).
7a	Real	*POptm*(2)	As above (material number 2).
.	.	.	.
.	.	.	.
7a	Real	*POptm*(*NMat*)	As above (for material number *NMat*).
			The following record (record 5b) is given only if *iMoSink*=1.
5b	Real	*P50*	Value of the pressure head, h_{50} (Fig. 2.1), at which the root water uptake is reduced by 50%.

Table 10.4. (continued)

Record	Type	Variable	Description
5b	Real	*P3*	Exponent p in the S-shaped root water uptake stress response function. Recommended value is 3.
			The following records are given only if *lChem*=**.true.** .
8	-	-	Comment line.
9	Logical	*lSolRed*	=**.true.** : root water uptake is reduced due to salinity. =**.false.**: otherwise.
			The following records are given only if *lSolRed*=**.true.** .
10	-	-	Comment line.
11	Logical	*lSolAdd*	=**.true.** if the effect of salinity stress is additive to the pressure head stress. =**.false.** if the effect of salinity stress is multiplicative to the pressure head stress.
12	-	-	Comment line.
			The following two values are specified when the root water uptake salinity stress response function is described with the S-shaped function (2.6) or (2.7), i.e., *lMsSink*=.**true.**.
13a	Real	*c50*	Value of the osmotic head $h_{\phi 50}$, at which the root water uptake is reduced by 50%. This value is specified only when *lSolAdd*=**.false.**.
13a	Real	*P3c*	Exponent, p, in the S-shaped root water uptake salinity stress response function. Recommended value is 3. This value is specified only when *lSolAdd*=**.false.**.
			The following two values are specified when the root water uptake salinity stress response function is described with the threshold-slope function of *Maas* [1990], i.e., *lMsSink*=.**false.**. $\alpha(h_\phi) = 1 \qquad h_\phi^M \le h_\phi < 0$ $\alpha(h_\phi) = \max[0, 1. - 0.01(h_\phi - h_\phi^M) s_\phi] \qquad h_\phi < h_\phi^M$
13a	Real	*c50*	Value of the minimum osmotic head (the salinity threshold) h_ϕ^M, above which root water uptake is not reduced. This value is specified only when *lSolAdd*=**.false.**.
13a	Real	*P3c*	Slope, s_φ, of the curve determining fractional root water uptake decline per unit increase in salinity below the threshold. This value is specified only when *lSolAdd*=**.false.**.
13	Real	*aOsm*(1)	Osmotic coefficient, a_1, for the first solute [L^4M^{-1}].

Table 10.4. (continued)

Record	Type	Variable	Description
13	Real	*aOsm*(2)	Osmotic coefficient, a_2, for the second solute [L^4M^{-1}].
	.	.	.
	.	.	.
13	Real	*aOsm*(*NSD*)	Osmotic coefficient, a_n, for the last solute [L^4M^{-1}].
13	Logical	*lMsSink*	=**.true.** : S-shaped root water uptake salinity stress response function. =**.false.**: threshold function according *Maas* [1990].

+ Block D is not needed when the logical variable *SinkF* (Block A) is set equal to **.false.** .

Table 10.5. Block E - Solute transport information.$^+$

Record	Type	Variable	Description
1,2	-	-	Comment lines.
3	Real	*Epsi*	Temporal weighing coefficient. =0.0 for an explicit scheme. =0.5 for a Crank-Nicholson implicit scheme. =1.0 for a fully implicit scheme.
3	Logical	*lUpW*	**.true.** if upstream weighing formulation is to be used. **.false.** if the original Galerkin formulation is to be used.
3	Logical	*lArtD*	**.true.** if artificial dispersion is to be added in order to fulfill the stability criterion *PeCr* (see Section 6.4.6). **.false.** otherwise.
3	Logical	*lTDep*	**.true.** if at least one transport or reaction coefficient (*ChPar*) is temperature dependent. **.false.** otherwise. If *lTDep*=**.true.**, then all values of *ChPar*(*i,M*) should be specified at a reference temperature T_r=20^{o}C.
3	Real	*cTolA*	Absolute concentration tolerance [ML^{-3}], the value is dependent on the units used (set equal to zero if nonlinear adsorption is not considered).
3	Real	*cTolR*	Relative concentration tolerance [-] (set equal to zero if nonlinear adsorption is not considered).
3	Integer	*MaxItC*	Maximum number of iterations allowed during any time step for solute transport - usually 20 (set equal to zero if nonlinear adsorption is not considered).
3	Real	*PeCr*	Stability criteria (see Section 6.4.6). Set equal to zero when *lUpW* is equal to **.true.**.
3	Integer	*NS*	Number of solutes in a chain reaction.
3	Logical	*lTort*	**.true.** if the tortuosity factor [*Millington and Quirk,* 1961] is to be used. **.false.** if the tortuosity factor is assumed to be equal to one.
4	-	-	Comment line.
5	Real	*ChPar*(1,*M*)	Bulk density of material *M*, ρ [ML^{-3}].
5	Real	*ChPar*(2,*M*)	Longitudinal dispersivity for material type *M*, D_L [L].
5	Real	*ChPar*(3,*M*)	Transverse dispersivity for material type *M*, D_T [L].
5	Real	*ChPar*(4,*M*)	Dimensionless fraction of the sorption sites classified as type-1, i.e., sites with instantaneous sorption, when the **chemical nonequilibrium** option is considered. Set equal to 1 if equilibrium transport is to be considered. Dimensionless fraction of the sorption sites in contact with mobile water when the **physical nonequilibrium** option is considered. Set equal to 1 if all sorption sites are in contact with mobile water.
5	Real	*ChPar*(5,*M*)	Immobile water content. Set equal to 0 when the **physical nonequilibrium** option is not considered.

Table 10.5. (continued)

Record	Type	Variable	Description
			Record 5 information is provided for each material *M* (from 1 to *NMat*).
6	-	-	Comment line.
7	Real	*ChPar*(6,*M*)	Ionic or molecular diffusion coefficient in free water, D_w [L^{2}T^{-1}].
7	Real	*ChPar*(7,*M*)	Ionic or molecular diffusion coefficient in gas phase, D_g [L^{2}T^{-1}].
8	-	-	Comment line.
9	Real	*ChPar*(8,*M*)	Adsorption isotherm coefficient, k_s, for material type *M* [L^{3}M^{-1}]. Set equal to zero if no adsorption is to be considered.
9	Real	*ChPar*(9,*M*)	Adsorption isotherm coefficient, η, for material type M [L^{3}M^{-1}]. Set equal to zero if Langmuir adsorption isotherm is not to be considered.
9	Real	*ChPar*(10,*M*)	Adsorption isotherm coefficient, β, for material type M [-]. Set equal to one if Freundlich adsorption isotherm is not to be considered.
9	Real	*ChPar*(11,*M*)	Equilibrium distribution constant between liquid and gas phases, k_g, material type *M* [-].
9	Real	*ChPar*(12,*M*)	First-order rate constant for dissolved phase, μ_w, material type *M* [T^{-1}].
9	Real	*ChPar*(13,*M*)	First-order rate constant for solid phase, μ_s, material type *M* [T^{-1}].
9	Real	*ChPar*(14,*M*)	First-order rate constant for gas phase, μ_g, material type *M* [T^{-1}].
9	Real	*ChPar*(15,*M*)	Rate constant, μ_w', representing a first-order decay for the first solute and zero-order production for the second solute in the dissolved phase, material type *M* [T^{-1}].
9	Real	*ChPar*(16,*M*)	Same as above for the solid phase, μ_s', material type *M* [T^{-1}].
9	Real	*ChPar*(17,*M*)	Same as above for the gas phase, μ_g', material type *M* [T^{-1}].
9	Real	*ChPar*(18,*M*)	Zero-order rate constant for dissolved phase, γ_w, material type *M* [ML^{-3}T^{-1}].
9	Real	*ChPar*(19,*M*)	Zero-order rate constant for solid phase, γ_s, material type *M* [T^{-1}].
9	Real	*ChPar*(20,*M*)	Zero-order rate constant for gas phase, γ_g, material type *M* [ML^{-3}T^{-1}].
9	Real	*ChPar*(21,*M*)	First-order rate coefficient for nonequilibrium adsorption, or the mass transfer coefficient for solute exchange between mobile and immobile liquid regions, ω, material type M [T^{-1}].
			Record 9 information is provided for each material *M* (from 1 to *NMat*).
			Record 6 through 9 information are provided for each solute (from 1 to *NS*).
10,11	-	-	Comment lines.
12	Real	*TDep*(6)	Activation energy for parameter *ChPar*(6,*M*) [ML^{2}T^{-2}M^{-1}] (See Section 3.4). This parameter should be specified in J mol^{-1}. Set equal to 0 if *ChPar*(6,*M*) is temperature independent.
12	Real	*TDep*(7)	Same for parameter *ChPar*(7,*M*) [ML^{2}T^{-2}M^{-1}].
13	-	-	Comment line.
14	Real	*TDep*(8)	Same for parameter *ChPar*(8,*M*) [ML^{2}T^{-2}M^{-1}].
.	.	.	.
.	.	.	.
14	Real	*TDep*(21)	Same for parameter *ChPar*(21,*M*) [ML^{2}T^{-2}M^{-1}].

Table 10.5. (continued)

Record	Type	Variable	Description
			Records 11 through 14 information is provided only when the logical variable *lTDep* of record 3 is set equal to **.true.**.
15	-	-	Comment line.
16	Real	*cBound*(1,1)	Value of the concentration for the first time-independent BC [ML^{-3}]. Set equal to zero if no *KodCB*(*n*)=±1 is specified.
16	Real	*cBound*(1,2)	Value of the concentration for the second time-independent BC [ML^{-3}]. Set equal to zero if no *KodCB*(*n*)= ±2 is specified.
.	.	.	.
.	.	.	.
16	Real	*cBound*(1,5)	Value of the concentration for the fifth time-independent BC [ML^{-3}]. If water uptake is specified, then *cBound*(1,5) is automatically used for the concentration of water removed from the flow region by root water uptake [ML^{-3}]. Set equal to zero if no *KodCB*(*n*)=±5 and no root solute uptake is specified.
16	Real	*cBound*(1,6)	Value of the concentration for the sixth time-independent BC [ML^{-3}]. If internal sources are specified, then *cBound*(1,6) is automatically used for the concentration of water injected into the flow region through internal sources [ML^{-3}]. Set equal to zero if no *KodCB*(*n*)=±6 and no internal sources are specified.
16	Real	*cBound*(1,7)	Concentration of the incoming fluid in equation (3.26) [ML^{-3}]. Set equal to zero if no *KodCB*(*n*)=-7 is specified.
16	Real	*cBound*(1,8)	Concentration above the stagnant boundary layer, g_{atm} (see equation (3.26)) [ML^{-3}]. Set equal to zero if no *KodCB*(*n*)=-7 is specified.
16	Real	*cBound*(1,9)	Thickness of the stagnant boundary layer, *d* [L] (see equation (3.26)). Set equal to zero if no *KodCB*(*n*)=-7 is specified.
			Record 16 information is provided for each solute (from 1 to *NS*).
17	-	-	Comment line.
18	Real	*tPulse*	Time duration of the concentration pulse [T].

$^{+}$Block E is not needed when the logical variable *lChem* in Block A is set equal to **.false.** .

Table 10.6. Block F - Heat transport information.[+]

Record	Type	Symbol	Description
1,2	-	-	Comment lines.
3	Real	*TPar*(1,*M*)	Volumetric solid phase fraction of material *M*, θ_n [-].
3	Real	*TPar*(2,*M*)	Volumetric organic matter fraction of material *M*, θ_o [-].
3	Real	*TPar*(3,*M*)	Longitudinal thermal dispersivity of material *M*, λ_L [L].
3	Real	*TPar*(4,*M*)	Transverse thermal dispersivity of material *M*, λ_T [L].
3	Real	*TPar*(5,*M*)	Coefficient b_1 in the thermal conductivity function [$MLT^{-3}K^{-1}$] (e.g.$Wm^{-1}K^{-1}$) (see equation (4.5)).
3	Real	*TPar*(6,*M*)	Coefficient b_2 in the thermal conductivity function [$MLT^{-3}K^{-1}$] (e.g.$Wm^{-1}K^{-1}$) (see equation (4.5)).
3	Real	*TPar*(7,*M*)	Coefficient b_3 in the thermal conductivity function [$MLT^{-3}K^{-1}$] (e.g.$Wm^{-1}K^{-1}$) (see equation (4.5)).
3	Real	*TPar*(8,*M*)	Volumetric heat capacity of solid phase of material *M*, C_n [$ML^{-1}T^{-2}K^{-1}$] (e.g. $Jm^{-3}K^{-1}$).
3	Real	*TPar*(9,*M*)	Volumetric heat capacity of organic matter of material *M*, C_o [$ML^{-1}T^{-2}K^{-1}$] (e.g. $Jm^{-3}K^{-1}$).
3	Real	*TPar*(10,*M*)	Volumetric heat capacity of liquid phase of material *M*, C_w [$ML^{-1}T^{-2}K^{-1}$] (e.g. $Jm^{-3}K^{-1}$).
			Record 3 is required for each soil material *M* (from 1 to *NMat*).
4	-	-	Comment line.
5	Real	*TBound*(1)	Value of the first time-independent thermal boundary condition [^{0}C]. Set equal to zero if no *KodTB*(*n*)=±1 is specified.
5	Integer	*TBound*(2)	Value of the second time-independent thermal boundary condition [^{0}C]. Set equal to zero if no *KodTB*(*n*)=±2 is specified.
.	.	.	.
.	.	.	.
5	Integer	*TBound*(6)	Value of the sixth time-independent thermal boundary condition [^{0}C]. Set equal to zero if no *KodTB*(*n*)=±6 is specified. If internal sources are specified, then *TBound*(6) is automatically used for the temperature of water injected into the flow region through internal source.
6	-	-	Comment line.
7	Real	*Ampl*	Temperature amplitude at the soil surface [K] prescribed for nodes where *Kode*(*n*)=±4. Set equal to zero when no *Kode*(*n*)=±4 is specified.
7	Real	*tPeriod*	Time interval for the completion of one temperature cycle (usually 1 day) [T].

[+] Block F need not be supplied if logical variables *lTemp* (Block A) is set equal to **.false.**

Table 10.7. Block G - Finite element mesh information.

Record	Type	Variable	Description
1	Char	*cVersion*	Set text equal to "Pcp_File_Version=".
1	Integer	*iVersion*	Version of the Meshtria.txt file; set equal to 2 for current version.
2	Integer	*NumNP*	Number of nodal points.
2	Integer	*nEdges*	Number of mesh edges.
2	Integer	*NumEl*	Number of finite elements (only triangles are allowed).
2	Integer	*nEBdr*	Number of mesh edges which form the boundary of the transport domain.
3	Integer	*n*	Nodal number.
3	Real	*x*(*n*)	*x*-coordinate of node *n* [L] (always a horizontal coordinate).
3	Real	*z*(*n*)	*z*-coordinate of node *n* [L]. *z* is the vertical coordinate for problems involving vertical planar or axisymmetric flow. For axisymmetric flow, *z* coincides with the vertical axis of symmetry.
3	Real	*rDummy*	Any real number, e.g., 0.0.
3	Integer	*nDummy*	Any integer number, e.g., 0.
			Record 3 information is required for each node *n*. Numbering of the nodes is arbitrary except that the first *nEBdr*+1 nodes must be boundary nodes in a counter-clock wise direction.
4	-	-	Comment line.
5	Integer	*e*	Edge number.
5	Integer	*iEpb*(*e*)	Number of the beginning node of edge *e*.
5	Integer	*iEpe*(*e*)	Number of the ending node of edge *e*.
5	Integer	*nDummy*	Any integer number, e.g., 0.
5	Integer	*nDummy*	Any integer number, e.g., 0.
5	Integer	*nDummy*	Any integer number, e.g., 0.
5	Integer	*nDummy*	Any integer number, e.g., 0.
			Record 5 information is required for each edge *e*.
6	-	-	Comment line.
7	Integer	*t*	Triangular element number.
7	Integer	*KX*(*t*,1)	Global nodal number of the first triangle's vertex.
7	Integer	*KX*(*t*,2)	Global nodal number of the second triangle's vertex.
7	Integer	*KX*(*t*,3)	Global nodal number of the third triangle's vertex.
			Record 7 information is required for each triangle *t*.

Table 10.8. Block H - Nodal information.

Record	Type	Variable	Description
1	Char	*cVersion*	Set text equal to "Pcp_File_Version=".
1	Integer	*iVersion*	Version of the Domain.dat file; set equal to 2 for current version.
2,3	-	-	Comment lines.
4	Integer	*NS*	Number of solutes in a chain reaction.
4	Integer	*iEquil*	This variable is read only if the user interface is used. = 1; Equilibrium solute transport is considered. = 0; Nonequilibrium solute transport is considered. Set equal to 1 if *lChem* is equal to **.false.**.
5,6	-	-	Comment line.
7	Integer	*n*	Nodal number.
7	Integer	*Kode*(*n*)	Code specifying the type of boundary condition applied to a particular node. Permissible values are 0,1,±2, ±3, ±4,..., ±*NumKD* (see Section 8.3).
7	Real	*hNew*(*n*)	Initial value of the pressure head at node *n* [L] when variable *lInitW* in Block A is equal to .**true.**. Initial value of the water content at node *n* [L] when variable *lInitW* in Block A is equal to .**false.**. If *lWat*=.**false.** in Block A, then *hNew*(*n*) represents the initial guess of the pressure head (water content) for steady state conditions.
7	Real	*Q*(*n*)	Prescribed recharge/discharge rate at node *n*; [L^2T^{-1}] for planar flow, [L^3T^{-1}] for axisymmetric flow. *Q*(*n*) is negative when directed out of the system. When no value for *Q*(*n*) is needed, set *Q*(*n*) equal to zero.
7	Integer	*MatNum*(*n*)	Index for material whose hydraulic and transport properties are assigned to node *n*.
7	Real	*Beta*(*n*)	Value of the water uptake distribution, *b*(*x,z*) [L^{-2}], in the soil root zone at node *n*. Set *Beta*(*n*) equal to zero if node *n* lies outside the root zone.
7	Real	*Axz*(*n*)	Nodal value of the dimensionless scaling factor α_h [-] associated with the pressure head.
7	Real	*Bxz*(*n*)	Nodal value of the dimensionless scaling factor α_K [-] associated with the saturated hydraulic conductivity.
7	Real	*Dxz*(*n*)	Nodal value of the dimensionless scaling factor α_θ [-] associated with the water content.
7	Real	*Temp*(*n*)	Initial value of temperature at node *n* [^oC] (if *lTemp*=**.false.** then set equal to 0 or any initial value, which is to be used with temperature dependent water flow and solute transport).
7	Real	*Conc*(1,*n*)	Initial value of the concentration of the first solute at node *n* [ML^{-3}] (do not have to be specified if *lChem*=**.false.**).

Table 10.8. (continued)

Record	Type	Variable	Description
7	Real	*Conc*(2,*n*)	Initial value of the concentration of the second solute at node *n* [ML^{-3}] (must not be specified if *lChem*=**.true.** and *NS* < 2).
.	.	.	.
.	.	.	.
7	Real	*Conc*(*i,n*)	Initial value of the concentration of the last solute at node *n* [ML^{-3}] (must not be specified if *lChem*=**.true.** and *NS* < *i*)..
7	Real	*Sorb*(1,*n*)	Initial value of the adsorbed concentration on type-2 sites or initial concentration in the immobile zone of the first solute at node *n* [ML^{-3}]. This variable does not have to be specified if *lChem*=**.false.**. or *lEquil*=**.true.** .
7	Real	*Sorb*(2,*n*)	Initial value of the adsorbed concentration on type-2 sites or initial concentration in the immobile zone of the second solute at node *n* [ML^{-3}]. This variable does not have to be specified if *lChem*=**.false.** or *lEquil*=**.true.** or *NS* < 2.
.	.	.	.
.	.	.	.
7	Real	*Sorb*(*NS,n*)	Initial value of the adsorbed concentration on type-2 sites or initial concentration in the immobile zone of the *NS*th solute at node *n* [ML^{-3}]. This variable does not have to be specified if *lChem*=**.false.** or *lEquil*=**.true.** .

Table 10.9. Block I - Element information.

Record	Type	Variable	Description
1,2	-	-	Comment lines.
3	Integer	*e*	Element number.
3	Real	*Angle*(*e*)	Angle in degrees between K_1^A and the *x*-coordinate axis assigned to each element *e*.
3	Real	*ConA*1(*e*)	First principal component, K_1^A, of the dimensionless tensor $\mathbf{K}^A$ which describes the local anisotropy of the hydraulic conductivity assigned to element *e*.
3	Real	*ConA*2(*e*)	Second principal component, K_2^A.
3	Integer	*LayNum*(*e*)	Subregion number assigned to element *e*. In general, record 3 information is required for each element *e*, starting with *e*=1 and continuing sequentially until *e*=*NumEl*.

Table 10.10. Block J - Boundary information.

Record	Type	Variable	Description
1	Char	*cVersion*	Set text equal to "Pcp_File_Version=".
1	Integer	*iVersion*	Version of the Boundary.in file; set equal to 2 for current version.
2,3	-	-	Comment lines.
4	Integer	*NumBP*	Number of boundary nodes for which *Kode*(*n*) is not equal to 0.
4	Integer	*NObs*	Number of observation nodes for which values of the pressure head, the water content, temperature (for *lTemp*=.true.), and the solution and sorbed concentrations (for *lChem*=.true.) are printed at each time level.
4	Logical	*SeepF*	**.true.** if one or more seepage faces is to be considered.
4	Logical	*FreeD*	Set this variable equal to **.true.** if a unit vertical hydraulic gradient boundary condition (free drainage) is used at the bottom boundary. Otherwise set equal to **.false.** .
4	Logical	*DrainF*	Set this logical variable equal to **.true.** if a drain is to be simulated by means of a boundary condition. Otherwise set equal to **.false.**. Section 5.3.7 explains how tile drains can be represented as boundary conditions in a regular finite element mesh.
4	Logical	*qGWLF*	Set this variable equal to **.true.** if the discharge-groundwater level relationship *q*(*GWL*) given by equation (8.1) is used as the bottom boundary condition; *GWL*=*h*-*GWL0L*, where *h* is the pressure head at the boundary.
5	-	-	Comment line.
6	Integer	*KXB*(1)	Global node number of the first of a set of sequentially numbered boundary nodes for which *Kode*(*n*) is not equal to zero.
6	Integer	*KXB*(2)	As above for the second boundary node.
.	.	.	.
.	.	.	.
6	Integer	*KXB*(*NumBP*)	As above for the last boundary node.
7	-	-	Comment line.
8	Real	*Width*(1)	Width of the boundary [L] associated with boundary node *KXB*(1). *Width*(*n*) includes half the boundary length of each element connected to node *KXB*(*n*) along the boundary. The type of boundary condition assigned to *KXB*(*n*) is determined by the value of *Kode*(*n*). In case of axisymmetric flow, *Width*(*n*) represents the area of the boundary strip [L^2] associated with node *KXB*(*n*), and along a horizontal boundary should be calculated as $Width(j)=\frac{\pi}{3}[(x_{j\text{-}1}+2x_j)(x_j\text{-}x_{j\text{-}1})+(x_{j+1}+2x_j)(x_{j+1}\text{-}x_j)]$ If a unit vertical hydraulic gradient or a deep drainage boundary condition is specified at node *n*, then *Width*(*n*) represents only the horizontal component of the boundary.

Table 10.10. (continued)

Record	Type	Variable	Description
8	Real	*Width*(2)	As above for node *KXB*(2).
.	.	.	.
.	.	.	.
8	Real	*Width*(*NumBP*)	As above for node *KXB*(*NumBP*).
9	-	-	Comment line.
10	Real	*rLen*	Width of soil surface associated with transpiration [L]; represents surface area [L^2] in case of axisymmetrical flow. Set *rLen* equal to zero for problems without transpiration.
11	-	-	Comment line.
12	Real	*GWL0L*	Reference position of groundwater table (usually the *z*-coordinate of the soil surface).
12	Real	*Aqh*	Value of the parameter A_{qh} [LT^{-1}] in the *q*(*GWL*)-relationship (equation (8.1)); set to zero if *qGWLF*=**.false.**
12	Real	*Bqh*	Value of the parameter B_{qh} [L^{-1}] in the *q*(*GWL*)-relationship (equation (8.1)); set to zero if *qGWLF* =**.false.**
			Records 11 and 12 are provided only when the logical variable *qGWLF* =**.true.**
13	-	-	Comment line.
14	Integer	*Node*(1)	Global node number of the first observation node for which values of the pressure head, the water content, temperature (for *lTemp*=.true.), and the solution and sorbed concentrations (for *lChem*=.true.) are printed at each time level. It does not have to be specified if *NObs* = 0.
14	Integer	*Node*(2)	Same as above for the second observation node. It does not have to be specified if *NObs* < 2.
.	.	.	.
.	.	.	.
14	Integer	*Node*(*NObs*)	Same as above for the last observation node.
			Records 13 and 14 are provided only when the logical variable *NObs* > 0.
15,16	-	-	Comment lines.
17	Integer	*NSeep*	Number of seepage faces expected to develop.
18	-	-	Comment line.
19	Integer	*NSP*(1)	Number of nodes on the first seepage face.
19	Integer	*NSP*(2)	Number of nodes on the second seepage face.
.	.	.	.
.	.	.	.
19	Integer	*NSP*(*NSeep*)	Number of nodes on the last seepage face.
20	-	-	Comment line.

Table 10.10. (continued)

Record	Type	Variable	Description
21	Integer	*NP*(1,1)	Sequential global number of the first node on the first seepage face.
21	Integer	*NP*(1,2)	Sequential global number of the second node on the first seepage face.
.	.	.	.
.	.	.	.
21	Integer	*NP*(1,*NSP*(1))	Sequential global number of the last node on the first seepage face.
			Record 21 information is provided for each seepage face. Records 15 through 21 information is not provided if the logical variable *SeepF* is set equal to **.false.** .
22,23	-	-	Comment lines.
24	Integer	*NDr*	Number of drains. See Section 5.3.7 for a discussion on how tile drains can be represented as boundary conditions in a regular finite element mesh.
24	Real	*DrCorr*	Additional reduction in the correction factor C_d (See Section 5.3.7).
25	-	-	Comment line.
26	Integer	*ND*(1)	Global number of the first drain.
26	Integer	*ND*(2)	Global number of the second drain.
.	.	.	.
.	.	.	.
26	Integer	*ND*(*NDr*)	Global number of the last drain.
27	-	-	Comment line.
28	Integer	*NElD*(1)	Number of elements surrounding the first drain.
28	Integer	*NElD*(2)	Number of elements surrounding the second drain.
.	.	.	.
.	.	.	.
28	Integer	*NElD*(*NDr*)	Number of elements surrounding the last drain.
29	-	-	Comment line.
30	Real	*EfDim*(1,1)	Effective diameter of the first drain, d_e [L] (see Section 5.3.7).
30	Real	*EfDim*(2,1)	Dimension of the square in finite element mesh representing the first drain, *D* [L] (see Section 5.3.7).
			Record 30 information is provided for each drain.
31	-	-	Comment line.
32	Integer	*KElDr*(1,1)	Global number of the first element surrounding the first drain.
32	Integer	*KElDr*(1,2)	Global number of the second element surrounding the first drain.
.	.	.	.
.	.	.	.
32	Integer	*KElDr*(1,*NElD*(1))	Global number of the last element surrounding the first drain.
			Record 32 information is provided for each drain. Records 22 through 32 are provided only if the logical variable *DrainF* is set equal to **.true.** .

Table 10.10. (continued)

Record	Type	Variable	Description
33,34	-	-	Comment lines.
35	Integer	*KodCB*(1)	Code specifying the type of boundary condition for solute transport applied to a particular node. Positive and negative signs indicate that first-, or second- or third- (depending upon the calculated water flux *Q*), type boundary conditions are implemented, respectively. In case of time-independent boundary conditions (*Kode*(1)= ±1, ±2, ±5, or ±6; See Block H), *KodCB*(1) also refers to the field *cBound* for the value of the solute transport BC. *cBound*(*i*,abs(*KodCB*(1))) is the value of the boundary condition for node *KXB*(1) (the first of a set of sequentially numbered boundary nodes for which *Kode*(*N*) is not equal to zero). Permissible values are ±1, ±2,..., ±5, ±6,-7. When *KodCB*(1) equals -7, a special type of boundary condition for volatile solutes described by equations (3.26) and (3.27) is applied.
35	Integer	*KodCB*(2)	Same as above for the second boundary node.
.	.	.	.
.	.	.	.
35	Integer	*KodCB*(*NumBP*)	Same as above for the last boundary node.
			Records 33 through 35 are provided only if the logical variable *lChem* (Block A) is set equal to **.true.** .
36,37	-	-	Comment line.
38	Integer	*KodTB*(1)	Code specifying the type of boundary condition for heat transport applied to a particular node. Positive and negative signs mean that first-, or second- or third- (depending upon the calculated water flux *Q*) type boundary conditions will be implemented, respectively. In case of time-independent boundary conditions (*Kode*(1)= ±1, ±2, ±5, or ±6; See Block H), *KodTB*(1) refers to the vector *TBound* for the value of the heat transport BC. *TBound*(abs(*KodTB*(1))) is the value of the boundary condition for node *KXB*(1) (the first of a set of sequentially numbered boundary nodes for which *Kode*(*N*) is not equal to zero). Permissible values are ±1,..., ±6.
38	Integer	*KodTB*(2)	Same as above for the second boundary node.
.	.	.	.
.	.	.	.
38	Integer	*KodTB*(*NumBP*)	Same as above for the last boundary node.
			Records 36 through 38 are provided only if the logical variable *lTemp* (Block A) is set equal to **.true.** .

Table 10.11. Block K - Atmospheric information.$^+$

Record	Type	Variable	Description
1	Char	*cVersion*	Set text equal to "Pcp_File_Version=".
1	Integer	*iVersion*	Version of the Atmosph.in file; set equal to 2 for current version.
2,3	-	-	Comment lines.
4	Integer	*MaxAl*	Number of atmospheric data records.
5	-	-	Comment line.
6	Real	*hCritS*	Maximum allowed pressure head at the soil surface [L].
7	-	-	Comment line.
8	Real	*tAtm*(*i*)	Time for which the *i*-th data record is provided [T].
8	Real	*Prec*(*i*)	Precipitation [LT^{-1}] (in absolute value).
8	Real	*rSoil*(*i*)	Potential evaporation rate [LT^{-1}] (in absolute value).
8	Real	*rRoot*(*i*)	Potential transpiration rate [LT^{-1}] (in absolute value).
8	Real	*hCritA*(*i*)	Absolute value of the minimum allowed pressure head at the soil surface [L].
8	Real	*rGWL*(*i*)	Drainage flux [LT^{-1}] across the bottom boundary, or other time-dependent prescribed flux boundary condition (positive when water leaves the flow region), for nodes where *Kode*(*n*)=-3; set to zero when no *Kode*(*n*)=-3 boundary condition is specified.
8	Real	*GWL*(*i*)	Groundwater level [L] (usually negative), or other time-dependent prescribed head boundary condition, for nodes where *Kode*(*n*)=+3; set equal to zero when no *Kode*(*n*)=+3 is specified. The prescribed value of the pressure head is *h*=*GWL*+*GWL0L*.
8	Real	*Temp1*(*i*)	First time-dependent temperature [K] prescribed for nodes where *Kode*(*n*)= ±3 or ±4 (must not be specified if *lTemp*=**.false.** and *lChem*=**.true.**; set equal to zero when no *Kode*(*n*)= ±3 or ±4 or when the flux is directed out of the flow domain).
8	Real	*Temp2*(*i*)	Second time-dependent temperature [K] prescribed for nodes where *Kode*(*n*)= ±3 or ±4 (must not be specified if *lTemp*=**.false.** and *lChem*=**.true.**; set equal to zero when no *Kode*(*n*)= ±3 or ±4 or when the flux is directed out of the flow domain).
8	Real	*Conc1*(1,*i*)	First time-dependent solute concentration [ML^{-3}] prescribed for nodes where *Kode*(*n*)= ±3 or ±4 (does not need to be specified if *lChem*=**.false.**).
8	Real	*Conc2*(1,*i*)	Second time-dependent solute concentration [ML^{-3}] prescribed for nodes where *Kode*(*n*)= ±3 or ±4 (does not need to be specified if *lChem*=**.false.**; set equal to zero when no *Kode*(*n*)= ±3 or ±4, or when the flux is directed out of the flow domain).
8	Real	*Conc3*(1,*i*)	Third time-dependent solute concentration [ML^{-3}] prescribed for nodes where *Kode*(*n*)= ±3 or ±4 (does not need to be specified if *lChem*=**.false.**; set equal to zero when no *Kode*(*n*)= ±3 or ±4, or when the flux is directed out of the flow domain).

Table 10.11. (continued)

Record	Type	Variable	Description
8	Real	*cPrec*(2,*i*)	Same as *cPrec*(1,*i*) for the second solute (does not need to be specified if *lChem*=**.false.** or NS < 2).
8	Real	*crt*(2,*i*)	Same as *crt*(1,*i*) for the second solute (does not need to be specified if *lChem*=**.false.** or NS < 2).
8	Real	*cht*(2,*i*)	Same as *cht*(1,*i*) for the second solute (does not need to be specified if *lChem*=**.false.** or NS < 2).
			The last three entries are entered for each solute from 1 to *NS*.
			The total number of atmospheric data records is *MaxAl* (i=1,2, ..,*MaxAl*).

[+] Block K is not read in if the logical variable *AtmInf* (Block A) is set equal to **.false.** .

Table 10.12. Block L - Dimension Information.

Record	Type	Variable	Description
1	Char	*cVersion*	Set text equal to "Pcp_File_Version=".
1	Integer	*iVersion*	Version of the Dimensio.in file; set equal to 2 for current version.
2	-	-	Comment lines.
3	Integer	*NumNPD*	Maximum number of nodes in finite element mesh
3	Integer	*NumElD*	Maximum number of elements in finite element mesh
3	Integer	*NumBPD*	Maximum number of boundary nodes for which *Kode*(*n*)≠0
3	Integer	*MBandD*	Maximum dimension of the bandwidth of matrix *A*
3	Integer	*NSeepD*	Maximum number of seepage faces
3	Integer	*NumSPD*	Maximum number of nodes along a seepage face
3	Integer	*NDrD*	Maximum number of drains
3	Integer	*NElDrD*	Maximum number of elements surrounding one drain
3	Integer	*NMatD*	Maximum number of materials
3	Integer	*NObsD*	Maximum number of observation nodes (maximum is 10)
3	Integer	*NSD*	Maximum number of solutes in a chain reaction.

Table 10.13. Block M - Inverse solution information.[+]

Record	Type	Variable	Description
1	Char	*cVersion*	Set text equal to "Pcp_File_Version=".
1	Integer	*iVersion*	Version of the Fit.in file; set equal to 2 for current version.
2	Integer	*NCase*	Number of cases being considered (only for the first data set).
3	Char	*Title1*	Descriptive title for simulation.
4	Char	*Title2*	Descriptive title for simulation.
5	-	-	Comment line.
6	Integer	*NOBB*	Number of observed data.
6	Integer	*MIT*	Maximum number of iterations for the inverse problem.
6	Integer	*iWeight*	Type of weighting used for the data set. = 0; no internal weighting. = 1; weighting by mean ratio. = 2; weighting by standard deviation.
7	-	-	Comment line.
8	Logical	*lWatF*	Set this logical variable equal to **.true.** when the soil hydraulic parameters are to be optimized.
8	Logical	*lChemF*	Set this logical variable equal to **.true.** when the solute transport parameters are to be optimized.
8	Integer	*NMat*	Number of soil materials. Materials are identified by the material number, *MatNum*, specified in Block H.
9	-	-	Comment line.
10	Integer	*iModel*	Soil hydraulic properties model: = 0; *van Genuchten*'s [1980] model containing six parameters. = 1; modified van Genuchten's model containing ten parameters, *Vogel and Císlerová* [1988]. = 2; *Brooks and Corey*'s [1964] model containing six parameters.
10	Integer	*iHyst*	Hysteresis in the soil hydraulic properties: = 0; no hysteresis = 1; hysteresis in the retention curve only = 2; hysteresis in both the retention and hydraulic conductivity functions
10	Logical	*lAniz*	=**.true.** is the coefficient of anisotropy is to be optimized.
11	-	-	Comment line.
12	Integer	*iQSame*	Parameter constraints = 0: $\theta_s^d > \theta_s^w$ = 1: $\theta_s^d = \theta_m^d$, $\theta_s^w = \theta_m^w$ = 2: $\theta_s^d = \theta_s^w = \theta_m$

Table. 10.13. (continued).

Record	Type	Variable	Description
12	Logical	*lAw2Ad*	**.true.** if parameter constraint $\alpha^w = 2\ \alpha^d$ is to be considered. **.false.** if no constraint on α^w and α^d is imposed.
12	Logical	*lKSame*	**.true.** if parameter constraint $K_s^w = K_s^d$ is to be considered. **.false.** if no constraint on K_s^w and K_s^d is imposed.
12	Integer	*iKappa*	= -1 if the initial condition is to be calculated from the main drying branch. = 1 if the initial condition is to be calculated from the main wetting branch.
			Records 11 and 12 are provided only when *iHyst* > 0. Records 9 through 12 are specified only when the logical variable *lWatF* is equal to **.true.**.
13	-	-	Comment line.
14	Integer	*NS*	Number of solutes (must be equal to 1). Records 13 and 14 are specified only when the logical variable *lChemF* is equal to **.true.**.
15	-	-	Comment line.
16	Real	*Par*(1,*M*)	Initial estimate of parameter θ_r for material *M* [-].
16	Real	*Par*(2,*M*)	Initial estimate of parameter θ_s for material *M* [-].
16	Real	*Par*(3,*M*)	Initial estimate of parameter α for material *M* [L^{-1}].
16	Real	*Par*(4,*M*)	Initial estimate of parameter *n* for material *M* [-].
16	Real	*Par*(5,*M*)	Initial estimate of parameter K_s for material *M* [LT^{-1}].
16	Real	*Par*(6,*M*)	Initial estimate of parameter *l* for material *M* [-].
			The following four parameters are specified only when *iModel*=1.
16	Real	*Par*(7,*M*)	Initial estimate of parameter θ_m for material *M* [-].
16	Real	*Par*(8,*M*)	Initial estimate of parameter θ_a for material *M* [-].
16	Real	*Par*(9,*M*)	Initial estimate of parameter θ_k for material *M* [-].
16	Real	*Par*(10,*M*)	Initial estimate of parameter K_k for material *M* [LT^{-1}].
			The following four parameters are specified only when *iModel*=0 and *iHyst*>1.
16	Real	*Par*(7,*M*)	Initial estimate of parameter θ_m for material *M* [-].
16	Real	*Par*(8,*M*)	Initial estimate of parameter θ_s^w for material *M* [-].
16	Real	*Par*(9,*M*)	Initial estimate of parameter α^w for material *M* [L^{-1}].
16	Real	*Par*(10,*M*)	Initial estimate of parameter K_s^w for material *M* [LT^{-1}].
17	Integer	*Index*(1,*M*)	Parameter estimation index for parameter θ_r, material *M* [-]. = 0; Coefficient is known and kept constant during optimization. = 1; Coefficient is unknown and estimated by curve fitting the data.
17	Integer	*Index*(2,*M*)	Parameter estimation index for parameter θ_s, material *M* [-].
17	Integer	*Index*(3,*M*)	Parameter estimation index for parameter α, material *M* [-].
17	Integer	*Index*(4,*M*)	Parameter estimation index for parameter *n*, material *M* [-].
17	Integer	*Index*(5,*M*)	Parameter estimation index for parameter K_s, material *M* [-].
17	Integer	*Index*(6,*M*)	Parameter estimation index for parameter *l*, material *M* [-].

Table 10.13. (continued).

Record	Type	Variable	Description
			The following four parameter estimation indices are specified only when *iModel*=1.
17	Integer	*Index*(7,*M*)	Parameter estimation index for parameter θ_m, material *M* [-].
17	Integer	*Index*(8,*M*)	Parameter estimation index for parameter θ_a, material *M* [-].
17	Integer	*Index*(9,*M*)	Parameter estimation index for parameter θ_k, material *M* [-].
17	Integer	*Index*(10,*M*)	Parameter estimation index for parameter K_k, material *M* [-].
			The following four parameter estimation indices are specified only when *iModel*=0 and *iHyst*>1.
17	Integer	*Index*(7,*M*)	Parameter estimation index for parameter θ_m, material *M* [-].
17	Integer	*Index*(8,*M*)	Parameter estimation index for parameter θ_s^w, material *M* [-].
17	Integer	*Index*(9,*M*)	Parameter estimation index for parameter α^w, material *M* [-].
17	Integer	*Index*(10,*M*)	Parameter estimation index for parameter K_s^w, material *M* [-].
18	Real	*BMn*(1,*M*)	Minimum constraint for parameter θ_r for material *M* [-] (dummy value if *Index*(1,*M*)=0).
18	Real	*BMn*(2,*M*)	Minimum constraint for parameter θ_s, material *M* [-].
18	Real	*BMn*(3,*M*)	Minimum constraint for parameter α, material *M* [L^{-1}].
18	Real	*BMn*(4,*M*)	Minimum constraint for parameter *n*, material *M* [-].
18	Real	*BMn*(5,*M*)	Minimum constraint for parameter K_s, material *M* [LT^{-1}].
18	Real	*BMn*(6,*M*)	Minimum constraint for parameter *l*, material *M* [-].
			The following four minimum parameter constraints are specified only when *iModel*=1.
18	Real	*BMn*(7,*M*)	Minimum constraint for parameter θ_m, material *M* [-].
18	Real	*BMn*(8,*M*)	Minimum constraint for parameter θ_a, material *M* [-].
18	Real	*BMn*(9,*M*)	Minimum constraint for parameter θ_k, material *M* [-].
18	Real	*BMn*(10,*M*)	Minimum constraint for parameter K_k, material *M* [LT^{-1}].
			The following four minimum parameter constraints are specified only when *iModel*=0 and *iHyst*>1.
18	Real	*BMn*(7,*M*)	Minimum constraint for parameter θ_m, material *M* [-].
18	Real	*BMn*(8,*M*)	Minimum constraint for parameter θ_s^w, material *M* [-].
18	Real	*BMn*(9,*M*)	Minimum constraint for parameter α^w, material *M* [L^{-1}].
18	Real	*BMn*(10,*M*)	Minimum constraint for parameter K_s^w, material *M* [LT^{-1}].
19	Real	*BMx*(1,*M*)	Maximum constraint for parameter θ_r, material *M* [-] (dummy value if *Index*(1,*M*)=0).
19	Real	*BMx*(2,*M*)	Maximum constraint for parameter θ_s, material *M* [-].
19	Real	*BMx*(3,*M*)	Maximum constraint for parameter α, material *M* [L^{-1}].
19	Real	*BMx*(4,*M*)	Maximum constraint for parameter *n*, material *M* [-].
19	Real	*BMx*(5,*M*)	Maximum constraint for parameter K_s, material *M* [LT^{-1}].
19	Real	*BMx*(6,*M*)	Maximum constraint for parameter *l*, material *M* [-].
			The following four maximum parameter constraints are specified only when *iModel*=1.
19	Real	*BMx*(7,*M*)	Maximum constraint for parameter θ_m, material *M* [-].
19	Real	*BMx*(8,*M*)	Maximum constraint for parameter θ_a, material *M* [-].

Table 10.13. (continued).

Record	Type	Variable	Description
19	Real	*BMx*(9,*M*)	Maximum constraint for parameter θ_k, material *M* [-].
19	Real	*BMx*(10,*M*)	Maximum constraint for parameter K_k, material *M* [LT^{-1}].
			The following four maximum parameter constraints are specified only when *iModel*=0 and *iHyst*>1.
19	Real	*BMx*(7,*M*)	Maximum constraint for parameter θ_m, material *M* [-].
19	Real	*BMx*(8,*M*)	Maximum constraint for parameter θ_s^w, material *M* [-].
19	Real	*BMx*(9,*M*)	Maximum constraint for parameter α^w, material *M* [L^{-1}].
19	Real	*BMx*(10,*M*)	Maximum constraint for parameter K_s^w, material *M* [LT^{-1}].
			Records 15 through 19 are specified only when logical variable *lWatF* is equal to **.true.** and then must be provided for each material *M* (from 1 to *NMat*). If *lWDep*=**.true.** (Block A) then the soil hydraulic parameters *Par*(*i,M*) must be specified at reference temperature T_{ref}=20°C.
20	-	-	Comment line.
21	Real	*ChPar*(1,*M*)	Initial estimate of bulk density of material *M*, ρ [ML^{-3}].
21	Real	*ChPar*(2,*M*)	Initial estimate of longitudinal dispersivity for material type *M*, D_L [L].
21	Real	*ChPar*(3,*M*)	Initial estimate of transverse dispersivity for material type *M*, D_L [L].
21	Real	*ChPar*(4,*M*)	Initial estimate of dimensionless fraction of the sorption sites classified as type-1, i.e., sites with instantaneous sorption when the **chemical nonequilibrium** option is considered. Set equal to 1 if equilibrium transport is to be considered. Dimensionless fraction of the sorption sites in contact with mobile water when the **physical nonequilibrium** option is considered. Set equal to 1 if all sorption sites are in contact with mobile water.
21	Real	*ChPar*(5,*M*)	Initial estimate of the immobile water content. Set equal to 0 when the **physical nonequilibrium** option is not considered.
21	Real	*ChPar*(6,*M*)	Initial estimate of the ionic or molecular diffusion coefficient in free water, D_w [L^2 T^{-1}].
21	Real	*ChPar*(7,*M*)	Initial estimate of the ionic or molecular diffusion coefficient in the gas phase, D_g [L^2 T^{-1}].
21	Real	*ChPar*(8,*M*)	Initial estimate of the adsorption isotherm coefficient, k_s, for material type *M* [L^3 M^{-1}]. Set equal to zero if no adsorption is to be considered.
21	Real	*ChPar*(9,*M*)	Initial estimate of the adsorption isotherm coefficient, η, for material type M [L^{3}M^{-1}]. Set equal to zero if a Langmuir adsorption isotherm is not to be considered.
21	Real	*ChPar*(10,*M*)	Initial estimate of the adsorption isotherm coefficient, β, for material type M [-]. Set equal to one if a Freundlich adsorption isotherm is not to be considered.
21	Real	*ChPar*(11,*M*)	Initial estimate of the equilibrium distribution constant between the liquid and gas phases, k_g, material type *M* [-].
21	Real	*ChPar*(12,*M*)	Initial estimate of the first-order rate constant for the dissolved phase, μ_w, material type *M* [T^{-1}].
21	Real	*ChPar*(13,*M*)	Initial estimate of the first-order rate constant for the solid phase, μ_s, material type *M* [T^{-1}].
21	Real	*ChPar*(14,*M*)	Initial estimate of the first-order rate constant for the gas phase, μ_g, material type *M* [T^{-1}].

Table 10.13. (continued).

Record	Type	Variable	Description
21	Real	*ChPar*(15,*M*)	Initial estimate of the rate constant, μ_w', representing first-order decay for the first solute and zero-order production for the second solute in the dissolved phase, material type *M* [T^{-1}].
21	Real	*ChPar*(16,*M*)	Initial estimate of the rate constant for the solid phase, μ_s', material type *M* [T^{-1}].
21	Real	*ChPar*(17,*M*)	Initial estimate of the rate constant for the gas phase, μ_g', material type *M* [T^{-1}].
21	Real	*ChPar*(18,*M*)	Initial estimate of the zero-order rate constant for the dissolved phase, γ_w, material type *M* [ML^{-3}T^{-1}].
21	Real	*ChPar*(19,*M*)	Initial estimate of the zero-order rate constant for the solid phase, γ_s, material type *M* [T^{-1}].
21	Real	*ChPar*(20,*M*)	Initial estimate of the zero-order rate constant for the gas phase, γ_g, material type *M* [ML^{-3}T^{-1}].
21	Real	*ChPar*(21,*M*)	Initial estimate of the first-order mass transfer coefficient for nonequilibrium adsorption, ω, material type M [T^{-1}].
22	Integer	*Index*(1,*M*)	Parameter estimation index for parameter *ChPar*(1,*M*). = 0; Coefficient is known and kept constant during optimization. = 1; Coefficient is unknown and estimated by curve fitting the data.
.	.	.	.
22	Integer	*Index*(21,*M*)	Parameter estimation index for parameter *ChPar*(21,*M*).
23	Real	*BMn*(1,*M*)	Minimum constraint for parameter *ChPar*(1,*M*) (dummy value if *Index*(1,*M*)=0).
.	.	.	.
.	.	.	.
23	Real	*BMn*(21,*M*)	Minimum constraint for parameter *ChPar*(21,*M*).
24	Real	*BMx*(1,*M*)	Maximum constraint for parameter *ChPar*(1,*M*) (dummy value if *Index*(1,*M*)=0).
.	.	.	.
.	.	.	.
24	Real	*BMx*(21,*M*)	Maximum constraint for parameter *ChPar*(21,*M*).
			Records 20 through 24 are specified only when the logical variable *lWatF* is equal to **.true.**, in which case they must be provided for each material *M* (from 1 to *NMat*).
25	-	-	Comment line.

26	Real	*HO*(*i*)	Observation data.
			Time *t* for *iType*(*i*)=0,1,2,3,4;
			Pressure head *h* for *iType*(*i*)=5,6;
			Dummy for *iType*(*i*)=7,8,9,10,11;

26	Real	*FO*(*i*)	Observation data (see description with variable *iType*). When *iType*(*i*)=2 and *iPos*(*i*)=0, then *FO*(*i*) represents the total volume of water in the entire flow domain. When *iType*(*i*)=4 and *iPos*(*i*)=0, then *FO*(*i*) represents the average concentration of the entire flow domain.

Table 10.13. (continued).

Record	Type	Variable	Description
26	Integer	*iType*(*i*)	Type of observed data: = 0: cumulative boundary water flux versus time = 1: $h(x,t)$ measurement = 2: $\theta(x,t)$ measurement = 3: boundary flux versus time = 4: $Conc(x,t)$ measurement = 5: $h(\theta)$ measurement = 6: $K(h)$ measurement = 7: prior knowledge of parameter α = 8: prior knowledge of parameter n = 9: prior knowledge of parameter θ_r = 10: prior knowledge of parameter θ_s = 11: prior knowledge of parameter K_s
26	Integer	*iPos*(*i*)	Position of the observation node for *iType*(*i*)=1,2,4; allowed values are 1, 2,...*NObs*. When *iType*(*i*)=0 or 3, then *iPos*(*i*) is equal to *Kode*(*n*) representing a particular type of boundary condition, e.g., 1 for a constant b.c., 2 for a seepage face, 3 for a varible b.c., 4 for an atmospheric b.c., 5 for a internal drains, and 6 for deep or free drainage. When *iType*(*i*)=5,6,7,8,9,10, or 11, then *iPos*(*i*) represents the material number *M*; allowed values are 1, 2,...*NMat*.
26	Real	*Weight*(*i*)	Weight associated with a particular data point.

[+] Block M is not needed if only the direct solution is calculated.

11. OUTPUT DATA

The program output consists of 14+3n_s output files organized into 3 groups:

T-level information
- H_MEAN.OUT
- V_MEAN.OUT
- CUM_Q.OUT
- RUN_INF.OUT
- SOLUTEx.OUT
- OBSNOD.OUT

P-level information
- H.OUT
- TH.OUT
- CONCx.OUT
- SORBCx.OUT
- TEMP.OUT
- V.OUT
- BOUNDARY.OUT
- BALANCE.OUT

A-level information
- A_LEVEL.OUT

In addition, some of the input data are printed to filea CHECK.OUT and FIT.OUT. Separate output files SOLUTEx.OUT, CONCx.OUT and SORBx.OUT are created for each solute. The various output files are described in detail below. The file CHECK.OUT summarizes the most important input variables, including the hydraulic and transport properties of each soil material. Results of the inverse solution are written to an output file FIT.OUT (Table 11.8). All output files are directed to the same directory as the input files, and must be created by the user prior to program execution (the directory is created automatically if the user interface is used).

T-level information - This group of output files contains information, which is printed at the end of each time step. Printing can be suppressed by setting the logical variable *ShortF* in input Block A equal to .true.; the information is then printed only at selected print times. Output files printed at the T-level are described in Tables 11.1 through 11.5. Output file OBSNOD.OUT brings the information about the time change of the pressure head, water content, temperature, and solution and sorbed concentrations, in specified observation nodes.

P-level information - P-level information is printed only at prescribed print times. The following output files are printed at the P-level:

H.OUT Nodal values of the pressure head

TH.OUT Nodal values of the water content

CONCx.OUT Nodal values of the solution concentrations

SORBx.OUT Nodal values of the nonequlibrium concentrations

TEMP.OUT Nodal values of the temperature

V.OUT Nodal values of the *x*- and *y*-components of the Darcian flux vector

BOUNDARY.OUT This file contains information about each boundary node, *n*, for which *Kode*(*n*) ≠ 0, including the discharge/recharge rate, *Q*(*n*), the boundary flux, *q*(*n*), the pressure head *h*(*n*), the water content $\theta(n)$, the temperature *Temp*(*n*), and the concentration *Conc*(n_s , *n*).

BALANCE.OUT This file gives the total amount of water, heat and solute inside each specified subregion, the inflow/outflow rates to/from that subregion, together with the mean pressure head (*hMean*), mean temperature (*TMean*) and the mean concentration (*cMean*) over each subregion (see Table 11.6). Absolute and relative errors in the water and solute mass balances are also printed to this file.

The output files H.OUT, TH.OUT, CONCx.OUT, SORBx.OUT, TEMP.OUT, and V.OUT provide binary output of the specific variables. The user interface can convert these binary files into the ASCII files H.TXT, TH.TXT, CONCx.TXT, SORBx.TXT, TEMP.TXT, and V.TXT.

A-level information - A-level information is printed each time a time-dependent boundary condition is specified. The information is directed to output file A_LEVEL.OUT (Table 11.7).

Table 11.1. H_MEAN.OUT - mean pressure heads.

hAtm Mean valuee of the pressure head calculated over a set of nodes for which *Kode*(*n*)=±4 (i.e., along part of a boundary controlled by atmospheric conditions) [L].

hRoot Mean value of the pressure head over a region for which *Beta*(*n*)>0 (i.e., within the root zone) [L].

*hKode*3 Mean value of the pressure head calculated over a set of nodes for which *Kode*(*n*)= ±3 (i.e., along part of a boundary where the groundwater level, the bottom flux, or other time-dependent pressure head and/or flux is imposed) [L].

*hKode*1 Mean value of the pressure head calculated over a set of nodes for which *Kode*(*n*)= ±1 (i.e., along part of a boundary where time-independent pressure heads and/or fluxes are imposed) [L].

hSeep Mean value of the pressure head calculated over a set of nodes for which *Kode*(*n*)= ±2 (i.e., along seepage faces) [L].

*hKode*5 Mean value of the pressure head calculated over a set of nodes for which *Kode*(*n*)= ±5 [L].

. .

. .

. .

hKodeN Mean value of the pressure head calculated over a set of nodes for which *Kode*(*n*)= ±*NumKD* [L].

Table 11.2. V_MEAN.OUT - mean and total water fluxes.*

rAtm	Potential surface flux per unit atmospheric boundary (*Kode(n)*=±4) $[LT^{-1}]$.
rRoot	Potential transpiration rate, T_p $[LT^{-1}]$.
vAtm	Mean value of actual surface flux per unit atmospheric boundary (*Kode(n)*=±4) $[LT^{-1}]$.
vRoot	Actual transpiration rate, T_a $[LT^{-1}]$.
*vKode*3	Total value of the bottom or other flux across part of a boundary where the groundwater level, the bottom flux, or other time-dependent pressure head and/or flux is imposed (*Kode*(*n*)=±3), $[L^2T^{-1}]$ or $[L^3T^{-1}]^+$.
*vKode*1	Total value of the boundary flux across part of a boundary where time-independent pressure heads and/or fluxes are imposed, including internal sinks/sources (*Kode*(*n*)=±1), $[L^2T^{-1}]$ or $[L^3T^{-1}]^+$.
vSeep	Total value of the boundary flux across a potential seepage face (*Kode*(*n*)=±2), $[L^2T^{-1}]$ or $[L^3T^{-1}]^+$.
*vKode*5	Total value of the flux across a boundary containing nodes for which *Kode*(*n*)=±5, $[L^2T^{-1}]$ or $[L^3T^{-1}]^+$.
.	.
.	.
.	.
vKodeN	Total value of the flux across a boundary containing nodes for which *Kode*(*n*)=±*NumKD*, $[L^2T^{-1}]$ or $[L^3T^{-1}]^+$.

+ For plane and axisymmetric flow, respectively

* Boundary fluxes are positive when water is removed from the system.

Table 11.3. CUM_Q.OUT - total cumulative water fluxes.[*]

CumQAP	Cumulative total potential surface flux across the atmospheric boundary (*Kode(n)*=±4), [L^2] or [L^3]$^+$.
CumQRP	Cumulative total potential transpiration rate, [L^2] or [L^3]$^+$.
CumQA	Cumulative total actual surface flux across the atmospheric boundary (*Kode(n)*=±4), [L^2] or [L^3]$^+$.
CumQR	Cumulative total actual transpiration rate, [L^2] or [L^3]$^+$.
*CumQ*3	Cumulative total value of the bottom or other boundary flux across part of a boundary where the groundwater level, the bottom flux, or other time-dependent pressure head and/or flux is imposed (*Kode*(*n*)=±3), [L^2] or [L^3]$^+$.
*CumQ*1	Cumulative total value of the flux across part of a boundary along which time-independent pressure heads and/or fluxes are imposed, including internal sinks/sources (*Kode*(*n*)=±1), [L^2] or [L^3]$^+$.
CumQS	Cumulative total value of the flux across a potential seepage faces (*Kode*(*n*)=±2), [L^2] or [L^3]$^+$.
*CumQ*5	Cumulative total value of the flux across a boundary containing nodes for which *Kode*(*n*)=±5, [L^2] or [L^3]$^+$.
.	.
.	.
.	.
CumQN	Cumulative total value of the flux across a boundary containing nodes for which *Kode*(*n*)=±*NumKD*, [L^2] or [L^3]$^+$.

[+] For plane and axisymmetric flow, respectively

[*] Boundary fluxes are positive when water is removed from the system.

Table 11.4. RUN_INF.OUT - time and iteration information.

TLevel	Time-level (current time-step number) [-].
Time	Time, *t*, at current time-level [T].
dt	Time step, Δt [T].
IterW	Number of iterations for water flow [-].
IterC	Number of iterations for solute transport [-].
ItCum	Cumulative number of iterations [-].
Peclet	Maximum local Peclet number [-].
Courant	Maximum local Courant number [-].

Table 11.5. SOLUTEx.OUT - actual and cumulative concentration fluxes.[*]

*CumCh*0	Cumulative amount of solute removed from the flow region by zero-order reactions (positive when removed from the system), [ML^{-1}] or [M][+].
*CumCh*1	Cumulative amount of solute removed from the flow region by first-order reactions, [ML^{-1}] or [M][+].
CumChR	Cumulative amount of solute removed from the flow region by root water uptake *S*, [ML^{-1}] or [M][+].
CumChN	Cumulative amount of solute being transferred to either kinetic adsorption sites (type-2 sorption sites), or to the immobile liquid region, [ML^{-1}] or [M][+].
*ChemS*1	Cumulative solute flux across part of a boundary along which time-independent pressure heads and/or fluxes are imposed, including internal sink/sources (*Kode*(*n*)= ±1), [ML^{-1}] or [M][+].
*ChemS*2	Cumulative solute flux across a potential seepage faces (*Kode*(*n*)= ±2), [ML^{-1}] or [M][+].
*ChemS*3	Cumulative solute flux across part of a boundary along which the groundwater level, the bottom flux, or other time-dependent pressure head and/or flux is imposed (*Kode*(*n*)= ±3), [ML^{-1}] or [M][+].
*ChemS*4	Cumulative total solute flux across the atmospheric boundary (*Kode*(*n*)= ±4), [ML^{-1}] or [M][+].
*ChemS*5	Cumulative total solute flux across an internal or external boundary containing nodes for which *Kode*(*n*)= ±5, [ML^{-1}] or [M][+].
.	.
.	.
.	.
ChemSN	Cumulative total solute flux across an internal or external boundary containing nodes for which *Kode*(*n*)= ±*NumKD*, [ML^{-1}] or [M][+].
*qc*1	Total solute flux across part of a boundary along which time-independent pressure heads and/or fluxes are imposed (*Kode*(*n*)= ±1), [ML^{-1}T^{-1}] or [MT^{-1}][+].
*qc*2	Total solute flux across a potential seepage face (*Kode*(*n*)= ±2), [ML^{-1}T^{-1}] or [MT^{-1}][+].
*qc*3	Total solute flux calculated across a boundary containing nodes for which *Kode*(*n*)= ±3 (i.e., along part of a boundary where the groundwater level, the bottom flux, or other time-dependent pressure head and/or flux is specified), [ML^{-1}T^{-1}] or [MT^{-1}][+].
*qc*4	Total solute flux across the atmospheric boundary (*Kode*(*n*)= ±4), [ML^{-1}T^{-1}] or [MT^{-1}][+].
*qc*5	Total solute flux across an internal or external boundary containing nodes for which *Kode*(*n*)= ±5, [ML^{-1}T^{-1}] or [MT^{-1}][+].
.	.
.	.
.	.
qcN	Total solute flux across an internal or external boundary containing nodes for which *Kode*(*n*)= ±*NumKD*, [ML^{-1}T^{-1}] or [MT^{-1}][+].

[+] For plane and axisymmetric flow, respectively

[*] The same output file is created for each solute from 1 to NS. Values of the solute flux and the cumulative solute flux are positive when solute is removed from the system.

Table 11.6. BALANCE.OUT - mass balance variables.

Area	Area of the entire flow domain or a specified subregion, [L^2] or [L^3]$^+$.
Volume	Volume of water in the entire flow domain or in a specified subregion, [L^2] or [L^3]$^+$.
InFlow	Inflow/Outflow to/from the entire flow domain or to/from a specified subregion, [L^{2}T^{-1}] or [L^{3}T^{-1}]$^+$.
hMean	Mean pressure head in the entire flow domain or in a specified subregion [L].
TVol	Amount of heat in the entire flow domain or in a specified subregion, [MLT^{-2}] or [ML^{2}T^{-2}]$^+$.
TMean	Mean temperature in the entire flow domain or in a specified subregion [K].
ConcVol	Amount of solute in the entire flow domain or in a specified subregion. This variable is given for all solutes from 1 to *NS*, [ML^{-1}] or [M]$^+$.
cMean	Mean concentration in the entire flow domain or in a specified subregion. This variable is given for all solutes from 1 to *NS* [ML^{-3}].
WatBalT	Absolute error in the water mass balance of the entire flow domain, [L^2] or [L^3]$^+$.
WatBalR	Relative error in the water mass balance of the entire flow domain [%].
CncBalT	Absolute error in the solute mass balance of the entire flow domain. This variable is given for all solutes from 1 to *NS*, [ML^{-1}] or [M]$^+$.
CncBalR	Relative error in the solute mass balance of the entire flow domain. This variable is given for all solutes from 1 to *NS* [%].

$^+$ For plane and axisymmetric flow, respectively

Table 11.7. A_LEVEL.OUT - mean pressure heads and total cumulative fluxes.[*]

CumQAP	Cumulative total potential flux across the atmospheric boundary (*Kode(n)*=±4), [L^2] or [L^3][+].
CumQRP	Cumulative total potential transpiration rate, [L^2] or [L^3][+].
CumQA	Cumulative total actual flux across the atmospheric boundary (*Kode*(*n*)=±4), [L^2] or [L^3][+].
CumQR	Cumulative total actual transpiration rate, [L^2] or [L^3][+].
*CumQ*3	Cumulative total bottom or other flux across a boundary along which the groundwater level, the bottom flux, or other time-dependent pressure head and/or flux is imposed (*Kode*(*n*)=±3), [L^2] or [L^3][+].
hAtm	Mean value of the pressure head calculated over a set of nodes for which *Kode*(*n*)=±4 [L].
hRoot	Mean value of the pressure head over a region for which *Beta*(*n*)>0 (i.e., the root zone) [L].
*hKode*3	Mean value of the pressure head over a set of nodes for which *Kode*(*n*)=±3 [L].

[+] For plane and axisymmetric flow, respectively

[*] Boundary fluxes are positive when water is removed from the system.

Table 11.8. FIT.OUT - information related to parameter estimation.

SSQ	Value of the objective function Φ being minimized during the parameter optimization process.
S.E.Coeff	Standard error.
RSQUARE	r^2 value for regression of observed versus fitted values.
Quantity	Measured data, e.g., the pressure head, water content, cumulative flux.
Type	Type of measured data (see Table 10.13).
Position	Position of the measurement (see Table 10.13).
Weight	Weight associated with a particular data point.
Residual	Residual between measured and fitted quantity.

12. PROGRAM ORGANIZATION

The program consists of a main program and 71 subprograms (77 subprograms when the inverse solution is used). The subprograms are organized by means of 10(11) source files, which are stored and compiled separately and then linked together with the main program to form an executable program. Below are a list and brief descriptions of the source files and the associated subprograms.

HYDRUS2.FOR	(Main program unit for the direct solution), ErrorOut
INVERSE2.FOR	(Main program unit for the inverse solution), QRSolv, MatInv, Weight, SortData, WContrib, CloseFiles
INPUT2.FOR	BasInf, MatIn, GenMat, TmIn, SeepIn, DrainIn, NodInf, ElemIn, GeomIn, AtmIn, SinkIn, ChemIn, TempIn
WATFLOW2.FOR	WatFlow, Reset, Dirich, Solve, Shift, SetMat, Hyster, Veloc
TIME2.FOR	TmCont, SetAtm, Fgh
MATERIA2.FOR	FK, FC, FQ, FH, FS
SINK2.FOR	SetSnk, FAlfa, FSAlfa
OUTPUT2.FORTLInf,	ALInf, hOut, thOut, FlxOut, SubReg, BouOut, cOut, SolInf, TOut, ObsNod
SOLUTE2.FOR Solute,	Coef, SolMat, c_Bound, Disper, SolveT, WeFact, PeCour, SorbConc, MassTran
TEMPER2.FORTemper,	T_Bound, DispT
ORTHOFEM.FOR	IADMake, Insert, Find, ILU, DU, ORTHOMIN, LUSolv, MatM2, SDot, SDotK, SNRM, SAXPYK, SCopy, SCopyK

Main program unit HYDRUS2.FOR

This is the main program unit of HYDRUS2. This unit controls execution of the program and determines which optional subroutines are necessary for a particular application.

Subroutine **ErrorOut** sends error messages to the screen and to the output file.

Main program unit INVERSE2.FOR

This is the main program unit of HYDRUS2 for the inverse solution. This unit controls execution of the program for inverse solutions, and calls the HYDRUS2 program to perform direct solutions.

Subroutine **QRSolv** solves a system of linear equations AX=B by QR-decomposition.

Subroutine **MatInv** inverts matrix A.

Subroutine **Weight** calculates internally generated weights associated with particular data points.

Subroutine **SortData** sorts time-related data in increasing numerical order.

Subroutine **WContrib** calculates the contribution of a particular measurement set to the final value of the objective function.

Table 12.1. Input subroutines/files.

Subroutine	Input Block	Input File
BasInf	A. Basic Information	
MatIn	B. Material Information	
TmIn	C. Time Information	
SinkIn	D. Sink Information	SELECTOR.IN
ChemIn	E. Solute Transport Information	
TempIn	F. Heat Transport Information	
NodInf	G. Finite Element Mesh Information	MESHTRIA.TXT
NodInf	H. Nodal Information	DOMAIN.DAT
ElemIn	I. Element Information	
GeomIn	J. Boundary Information	
SeepIn	J. Seepage Information	BOUNDARY.IN
DrainIn	J. Drainage Information	
AtmIn	K. Atmospheric Information	ATMOSPH.IN
Main	L. Dimension Information	DIMENSIO.IN
Main (Inverse)	M. Inverse Solution Information	FIT.IN

Source file INPUT.FOR

Subroutines included in this source file are designed to read data from different input blocks. The following table summarizes from which input file and input block (described in Section 10) a particular subroutine reads.
Subroutine **GenMat** generates for each soil type in the flow domain a table of water contents, hydraulic conductivities, and specific water capacities from set of hydraulic parameters.

Source file WATFLOW2.FOR

Subroutine **WatFlow** is the main subroutine for simulating water flow; this subroutine controls the entire iterative procedure of solving the Richards equation.

Subroutine **Reset** constructs the global matrix equation for water flow, including the right-hand side vector.

Subroutine **Dirich** modifies the global matrix equation by incorporating prescribed pressure head nodes.

Subroutine **Solve** solves the banded symmetric matrix equation for water flow by Gaussian elimination.

Subroutine **Shift** changes atmospheric or seepage face boundary conditions from Dirichlet type to Neumann type conditions, or vice versa, as needed. Also updates boundary conditions for the variable boundary fluxes (free and deep drainage).

Subroutine **Veloc** calculates nodal water fluxes.

Subroutine **SetMat** determines the nodal values of the hydraulic properties $K(h)$, $C(h)$ and $\theta(h)$ by interpolation between intermediate values in the hydraulic property tables.

Subroutine **Hyster** calculates scaling factors for the drying and wetting scanning curves, hysteresis reversal points, and other information connected with hysteresis in the soil hydraulic properties.

Source file TIME2.FOR

Subroutine **TmCont** adjusts the current value of the time increment Δt.

Subroutine **SetAtm** updates time variable boundary conditions.

Function **Fqh** describes the groundwater level - discharge relationship, $q(h)$, defined by equation (8.1). This function is called only from subroutine SetAtm.

Source file MATERIA2.FOR

This file includes the functions **FK, FC, FQ**, **FH** and **FS** which define the unsaturated hydraulic properties $K(h)$, $C(h)$, $\theta(h)$, $h(\theta)$, and $S_e(\theta)$ for each soil material.

Source file SINK2.FOR

This file includes subroutine **SetSnk** and functions **FAlfa** and **FSAlfa**. These subroutines calculate the actual root water extraction rate as a function of water and salinity stress in the soil root zone.

Source file OUTPUT2.FOR

The subroutines included in this file are designed to print data to different output files. Table 11.2 summarizes which output files are generated by a particular subroutine.

Source file SOLUTE2.FOR

Subroutine **Solute** is the main subroutine for simulating solute transport; this subroutine controls the entire iterative procedure of solving the nonlinear nonequilibrium convection-dispersion equation.

Table 11.2. Output subroutines/files.

Subroutine	Output File
TLInf	H_MEAN.OUT V_MEAN.OUT CUM_Q.OUT RUN_INF.OUT
SolInf	SOLUTEx.OUT
hOut	H.OUT
thOut	TH.OUT
cOut	CONCx.OUT
TOut	TEMP.OUT
FlxOut	V.OUT
BouOut	BOUNDARY.OUT
SubReg	BALANCE.OUT
ALInf	A_LEVEL.OUT
ObsNod	OBSNOD.OUT

Subroutine **Coeff** calculates solute transport parameters.

Subroutine **SolMat** constructs the global matrix equation for transport, including the right-hand side vector.

Subroutine **c_Bound** determines the values of the solute transport boundary codes, *cKod*(*n*), and incorporates prescribed boundary conditions in the global matrix equation for solute transport.

Subroutine **Disper** calculates nodal values of the dispersion coefficients for solute transport.

Subroutine **SolveT** solves the final asymmetric banded matrix equation for solute transport using Gaussian elimination.

Subroutine **WeFact** computes the optimum weighting factors for all sides of all elements.

Subroutine **PeCour** computes the maximum local Peclet and Courant numbers and the maximum permissible time step.

Subroutine **SorbConc** calculates nonequilibrium state variables, i.e, sorption on type-2 sorption sites, or concentrations in the immobile liquid region.

Subroutine **MassTran** calculates mass transfer fluxes for nonequilibrium processes.

Source file TEMPER2.FOR

Subroutine **Temper** is the main subroutine for simulating heat transport; this subroutine constructs the global matrix equation for heat transport, including the right-hand side vector.

Subroutine **T_Bound** determines the values of the heat transport boundary, and incorporates prescribed boundary conditions in the global matrix equation for heat transport.

Subroutine **DispT** calculates nodal values of the apparent dispersion coefficients for heat transport.

Source file ORTHOFEM.FOR

The subroutines included in this file solve large sparse systems of linear algebraic equations by the preconditioned conjugate gradient method for symmetric matrices, and by the ORTHOMIN method for asymmetric matrices. The subroutines were adopted from *Mendoza et al.* [1991]. See *Mendoza et al.* [1991] for a detailed description of both methods.

Subroutine **IADMake** generates the adjacency matrix which determines nodal connections from the finite element incidence matrix.

Subroutine **Insert** adds node *j* to the adjacency list for node *i*.

Subroutine **Find** retrieves from the adjacency matrix the appropriate position of two global points in the coefficient matrix.

Subroutine **ILU** performs incomplete lower-upper decomposition of matrix [*A*].

Function **DU** searches the *i*th row of the upper diagonal matrix for an adjacency of node *j*.

Subroutine **ORTHOMIN** governs the ORTHOMIN (conjugate gradient) acceleration.

Subroutine **LUSolv** performs lower diagonal matrix inversion by forward substitution, and upper diagonal matrix inversion by backward substitution.

Subroutine **MatM2** multiplies a matrix by a vector.

Function **SDot** calculates the dot product of two vectors.

Function **SDotK** calculates the dot product of a column in matrix by a vector.

Function **SNRM** computes the maximum norm of a vector.

Subroutine **SAXPYK** multiplies a column in a matrix by a scalar, and adds the resulting value to another vector.

Subroutine **SCopy** copies a vector into another vector.

Subroutine **SCopyK** copies a column in a matrix into a vector.

13. REFERENCES

Alexander, M., and K. M. Scow. 1989. Kinetics of biodegradation in soil, In *Reactions and Movement of Organic Chemicals in Soils*, edited by B. L. Sawhney and K. Brown, Spec. Publ. No. 22, 243-270, Soil Science Society of America, Madison, WI.

Bear, J. 1972. *Dynamics of Fluid in Porous Media*. Elsevier, New York, NY.

Behie, A., and P. Forsyth, Jr. 1983. Comparison of fast iterative methods for symmetric systems. *IMA J. of Numerical Analysis, 3,* 41-63.

Belmans, C., J. G. Wesseling and R. A. Feddes. 1983. Simulation model of the water balance of a cropped soil: SWATRE, *J. Hydrol., 63*, 271- 286.

Bromilow, R. H., and M. Leistra. 1980. Measured and simulated behavior of aldicarb and its oxidation products in fallow soils, *Pestic. Sci., 11*(4), 389-395.

Brooks, R. H., and A. T. Corey. 1966. Properties of porous media affecting fluid flow. *J. Irrig. Drainage Div.*, ASCE Proc. 72(IR2), 61-88.

Castro, C. L., and D. E. Rolston. 1977. Organic phosphate transport and hydrolysis in soil: theoretical and experimental evaluation, *Soil Sci. Soc. Am. J., 41*(6), 1085-1092.

Celia, M. A., and E. T. Bououtas, R. L. Zarba. 1990. A general mass-conservative numerical solution for the unsaturated flow equation, *Water Resour. Res., 26*, 1483-1496.

Chiou, C. T. 1989. Theoretical consideration of the partition uptake of nonionic organic compounds by soil organic matter, In *Reactions and Movement of Organic Chemicals in Soils*, edited by B. L. Sawhney and K. Brown, Spec. Publ. Number 22, 1-30, Soil Science Society of America, Madison, WI.

Chung S.-O., and R. Horton. 1987. Soil heat and water flow with a partial surface mulch, *Water Resour. Res., 23*(12), 2175-2186.

Cho, C. M. 1971. Convective transport of ammonium with nitrification in soil, *Can. Jour. Soil Sci., 51*(3), 339-350.

Christie, L. D., D. F. Griffiths, A. R. Mitchell, and O. C. Zienkiewicz. 1976. Finite element methods for second order differential equations with significant first derivatives, *Int. J. Num. Methods in Engineering, 10*, 1389-1396.

Císlerová, M. 1987. Comparison of simulated water balance for ordinary and scaled soil hydraulic characteristics, *Publ. No. 82*, Dept. of Hydraulics and Catchment Hydrology, Agricultural Univ., Wageningen, The Netherlands.

Cleary, R. W., and M. J. Ungs. 1978. Groundwater pollution and hydrology, Mathematical models and computer programs, *Research Rep. No. 78-WR-15*, Water Resour. Program, Princeton Univ. Princeton, NJ.

Clausnitzer, V., and J. W. Hopmans. 1995. Non-linear parameter estimation: LM_OPT. General-purpose optimization code based on the Levenberg-Marquardt algorithm, *Land, Air and Water Resources Paper No. 100032*, University of California, Davis, CA.

Constantz, J. 1982. Temperature dependence of unsaturated hydraulic conductivity of two soils. *Soil Sci. Soc. Am. J., 46*(3), 466-470.

Daniel, C., and F. S. Wood. 1971. *Fitting Equations to Data*. Wiley-Interscience, New York.

Davis, L. A., and S. P. Neuman. 1983. Documentation and user's guide: UNSAT2 - Variably saturated flow model, *Final Rep., WWL/TM-1791-1,* Water, Waste & Land, Inc., Ft. Collins, CO.

de Marsily, G. 1986. *Quantitative Hydrogeology*, Academic Press, London.

de Vries, D. A. 1963. The thermal properties of soils, In *Physics of Plant Environment*, edited by R. W. van Wijk, pp. 210-235, North Holland, Amsterdam.

Feddes, R. A., E. Bresler, and S. P. Neuman. 1974. Field test of a modified numerical model for water uptake by root systems, *Water Resour. Res., 10*(6), 1199-1206.

Feddes, R. A., P. J. Kowalik, and H. Zaradny. 1978. *Simulation of Field Water Use and Crop Yield*, John Wiley & Sons, New York, NY.

Fipps, G., R. W. Skaggs, and J. L. Nieber. 1986. Drains as a boundary condition in finite elements, *Water Resour. Res., 22*(11), 1613-1621.

Glotfelty, D. E., and C. J. Schomburg. 1989. Volatilization of pesticides from soil, In *Reactions and Movement of Organic Chemicals in Soils*, edited by B. L. Sawhney and K. Brown, Spec. Publ. Number 22, 181-208, Soil Science Society of America, Madison, WI.

Gribb, M. M. 1996. Parameter estimation for determining hydraulic properties of a fine sand from transient flow measurements, *Water Resour. Res., 32*(7), 1965-1974.

Gribb, M. M., J. Šimůnek, and M. F. Leonard. 1998. Development of a cone penetrometer method to determine soil hydraulic properties, *ASCE J. of Geotech. and Geoenviron. Eng., 124*(9), 820-829.

Gureghian, A. B. 1981. A two-dimensional finite-element solution for the simultaneous transport of water and multi-solutes through a nonhomogeneous aquifer under transient saturated-unsaturated flow conditions, *Sci. Total Environ., 21*, 329-337.

Gureghian, A. B., and G. Jansen. 1983. LAYFLO: A one-dimensional semianalytical model for the migration of a three-member decay chain in a multilayered geologic medium, *Tech. Rep. ONWI-466*, Office of Nuclear Waste Isolation, Battelle Memorial Institute, Columbus, OH.

Harada, M., P. L. Chambre, M. Foglia, K. Higashi, F. Iwamoto, D. Leung, T. H. Pigford, and D. Ting. 1980. Migration of radionuclides through sorbing media, analytical solutions - I, *Rep. no. LBL-10500 (UC-11)*, Lawrence Berkeley Laboratory, Univ. of California, Berkeley, CA.

Higashi, K., and T. H. Pigford. 1980. Analytical models for migration of radionuclides in geologic sorbing media, *J. Nucl. Sci. and Techn., 7*(9), 700-709.

Hopmans, J. W., and J. N. M. Stricker. 1989. Stochastic analysis of soil water regime in a watershed, *J. Hydrol., 105*, 57-84.

Huyakorn, P. S., and G. F. Pinder. 1983. *Computational Methods in Subsurface Flow*, Academic Press, London, United Kingdom.

Inoue, M., J. Šimůnek, J. W. Hopmans, and V. Clausnitzer. 1998. In-situ estimation of soil hydraulic functions using a multi-step soil water extraction technique, *Water Resour. Rer., 34*(5), 1035-1050.

Javandel, I., Ch. Doughty, Chin-Fu Tsang. 1984. *Groundwater Transport: Handbook of Mathematical Models*, Water Resour. Monograph No. 10, Am. Geophys. Union, Washington, D.C.

Jury, W. A., W. F. Spencer, and W. J. Farmer. 1983. Behavior assessment model for trace organics in soil, I. Model description, *J. Environ. Qual., 12*, 558-564.

Kirkham, D. 1949. Flow of ponded water into drain tubes in soil overlying an impervious layer, *Trans. Amer. Geoph. Union, 30*(3), 369-385.

Kirkham, D., and W. L. Powers. 1972. *Advanced Soil Physics*, John Wiley & Sons, New York, NY.

Kodešová, R., M. M. Gribb, and J. Šimůnek. 1998. Estimating soil hydraulic properties from transient cone permeameter data, *Soil Science, 163*(6), 436-453.

Kool, J. B., J. C. Parker, and M. Th. van Genuchten. 1985. Determining soil hydraulic properties from one-step outflow experiments by parameter estimation: I. Theory and numerical studies, *Soil Sci. Soc. Am. J., 49*, 1348-1354.

Kool, J. B., and J. C. Parker. 1987. Development and evaluation of closed-form expressions for hysteretic soil hydraulic properties. *Water Resour. Res., 23*(1), 105-114.

Kool, J. B., J. C. Parker, and M. Th. van Genuchten. 1987. Parameter estimation for unsaturated flow and transport models - A review, *J. Hydrol., 91*, 255-293.

Kool, J. B., M. Th. van Genuchten. 1991. HYDRUS - One-dimensional variably saturated flow and transport model, including hysteresis and root water uptake, Version 3.3, Research Report No. 124, U. S. Salinity Laboratory, USDA, ARS, Riverside, CA.

Leij, F. J., T. H. Skaggs, and M. Th. van Genuchten. 1991. Analytical solutions for solute transport in three-dimensional semi-infinite porous media, *Water Resour. Res., 27*(10), 2719-2733.

Leij, F. J., and S. A. Bradford. 1994. 3DADE: A computer program for evaluating three-dimensional equilibrium solute transport in porous media, *Research Report No. 134*, U. S. Salinity Laboratory, USDA, ARS, Riverside, CA.

Lester, D. H., G. Jansen, and H. C. Burkholder. 1975. Migration of radionuclide chains through an adsorbing medium, In: *Adsorption and Ion Exchange,* Am. Inst. Chem. Eng., Symp. Series no. 152, 71, 202-213.

Letniowski, F. W. 1989. An overview of preconditioned iterative methods for sparse matrix equations. *Research Report CS-89-26,* Faculty of Mathematics, Univ. of Waterloo, Waterloo, Ontario, Canada.

Luckner, L., M. Th. van Genuchten, and D. R. Nielsen. 1989. A consistent set of parametric models for the two-phase flow of immiscible fluids in the subsurface. *Water Resour. Res., 25*(10), 2187-2193.

Lynch, D. 1984. Mass conservation in finite element groundwater models. *Adv. Water Resour., 7*, 67-75.

Marquardt, D. W. 1963. An algorithm for least-squares estimation of nonlinear parameters, *SIAM J. Appl. Math., 11*, 431-441.

McCord, J. T. 1991. Application of second-type boundaries in unsaturated flow modeling. *Water Resour. Res., 27*(12), 3257-3260.

Meijerink, J. A. , and H. A. van der Vorst. 1977. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. of Comp., 31*(137), 148-162.

Mendoza, C. A., R. Therrien, and E. A. Sudicky. 1991. *ORTHOFEM User`s Guide, Version 1.02.*

Waterloo Centre for Groundwater Research, Univ. of Waterloo, Waterloo, Ontario, Canada.

Miller, E. E., and R. D. Miller. 1956. Physical theory for capillary flow phenomena, *J. Appl. Phys., 27*, 324-332.

Millington, R. J., and J. M. Quirk. 1961. Permeability of porous solids, *Trans. Faraday Soc., 57*, 1200-1207.

Misra, C., D. R. Nielsen, J. W. Biggar. 1974. Nitrogen transformations in soil during leaching: I. Theoretical considerations, *Soil Sci. Soc. Am. Proc., 38*(2), 289-293.

Mls, J. 1982. Formulation and solution of fundamental problems of vertical infiltration, *Vodohosp. Čas., 30*, 304-313 (in Czech).

Mohammad, F. S., and R. W. Skaggs. 1983. Drain tube opening effects on drain inflow, *J. Irrig. Drain. Div., Am. Soc. Civ. Eng., 109*(4), 393-404.

Mualem, Y. 1976. A new model for predicting the hydraulic conductivity of unsaturated porous media, *Water Resour. Res., 12*(3), 513-522.

Neuman, S. P. 1972. Finite element computer programs for flow in saturated-unsaturated porous media, *Second Annual Report, Project No. A10-SWC-77*, Hydraulic Engineering Lab., Technion, Haifa, Israel.

Neuman, S. P. 1973. Saturated-unsaturated seepage by finite elements, *J. Hydraul. Div.*, ASCE, 99 (HY12), 2233-2250.

Neuman, S. P., R. A. Feddes, and E. Bresler. 1974. Finite element simulation of flow in saturated-unsaturated soils considering water uptake by plants, *Third Annual Report*, *Project No. A10-SWC-77*, Hydraulic Engineering Lab., Technion, Haifa, Israel.

Neuman, S. P. 1975. Galerkin approach to saturated-unsaturated flow in porous media, Chapter 10 in *Finite Elements in Fluids, Vol. I, Viscous Flow and Hydrodynamics*, edited by R. H. Gallagher, J. T. Oden, C. Taylor, and O.C. Zienkiewicz, John Wiley & Sons, London, pp. 201-217.

Nielsen, D. R., and L. M. Luckner. 1992. Theoretical aspects to estimate reasonable initial parameters and range limits in indentification procedures for soil hydraulic properties. In, Proc. Intl. Workshop on *Indirect Methods for Estimating the Hydraulic Properties of Unsaturated Soils,* edited by M. Th van Genuchten, F. J. Leij, and L. J. Lund, University of California, Riverside, pp. 147-160.

Ou, L. T., P. S. C. Rao, K. S. V. Edvardson, R. E. Jessup, A. G. Hornsby, and R. L. Jones. 1988.

Aldicarb degradation in sandy soils from different depths, *Pesticide Sci., 23*, 1-12.

Perrochet, P., and D. Berod. 1993. Stability of the standard Crank-Nicolson-Galerkin scheme applied to the diffusion-convection equation: some new insights, *Water Resour. Res., 29*(9), 3291-3297.

Philip, J. R., and D. A. de Vries. 1957. Moisture movement in porous media under temperature gradients, *Eos Trans. AGU, 38*(2), 222-232.

Pignatello, J. J. 1989. Sorption dynamics of organic compounds in soils and sediments, In *Reactions and Movement of Organic Chemicals in Soils*, edited by B. L. Sawhney and K. Brown, Spec. Publ. Number 22, 45-80, Soil Science Society of America, Madison, WI.

Pinder, G. F., W. G. Gray. 1977. *Finite Element Simulation in Surface and Subsurface Hydrology*, Academic Press, New York, N.Y.

Rogers, V. C. 1978. Migration of radionuclide chains in groundwater, *Nucl. Techn.,* 40(3), 315-320.

Rogers, J. S., J. L. Fouss. 1989. Hydraulic conductivity determination from vertical and horizontal drains in layered soil profiles, *Transaction of the ASAE, 32*(2), 589-595.

Scott, P. S., G. J. Farquhar, and N. Kouwen. 1983. Hysteresis effects on net infiltration, Advances in Infiltration, *Publ. 11-83*, pp.163-170, Am. Soc. Agri. Eng., St. Joseph, Mich.

Selim, H. M., R. Schulin, H. Flühler. 1987. Transport and ion exchange of calcium and magnesium in an aggregated soil, *Soil Sci. Soc. Am. J., 51*(4), 876-884.

Simmons, C. S., D. R. Nielsen, J. W. Biggar. 1980. Scaling of field-measured soil water properties, *Hilgardia, 47*, 101-122.

Šimůnek, J., T. Vogel and M. Th. van Genuchten. 1992. The SWMS_2D code for simulating water flow and solute transport in two-dimensional variably saturated media, Version 1.1, *Research Report No. 126*, U. S. Salinity Laboratory, USDA, ARS, Riverside, CA.

Šimůnek, J., and D. L. Suarez.1993a. Modeling of carbon dioxide transport and production in soil: 1. Model development, *Water Resour. Res.*, *29*(2), 487-497.

Šimůnek, J., and D. L. Suarez. 1993b. UNSATCHEM-2D code for simulating two-dimensional variably saturated water flow, heat transport, carbon dioxide production and transport, and multicomponent solute transport with major ion equilibrium and kinetic chemistry, Version 1.1, *Research Report No. 128*, U. S. Salinity Laboratory, USDA, ARS, Riverside, CA.

Šimůnek, J., and M. Th. van Genuchten. 1994. The CHAIN_2D code for the two-dimensional movement of water, heat, and multiple solutes in variably saturated media, Version 1.1, *Research Report No. 136*, U. S. Salinity Laboratory, USDA, ARS, Riverside, CA.

Šimůnek, J., T. Vogel and M. Th. van Genuchten. 1994. The SWMS_2D code for simulating water flow and solute transport in two-dimensional variably saturated media, Version 1.1, *Research Report No. 132*, U. S. Salinity Laboratory, USDA, ARS, Riverside, CA.

Šimůnek, J., and M. Th. van Genuchten. 1995. Numerical model for simulating multiple solute transport in variably-saturated soils, *Proc. "Water Pollution III: Modelling, Measurement, and Prediction*, Ed. L. C. Wrobel and P. Latinopoulos, Computation Mechanics Publication, Ashurst Lodge, Ashurst, Southampton, UK, pp. 21-30.

Šimůnek, J., and M. Th. van Genuchten. 1996. Estimating unsaturated soil hydraulic properties from tension disc infiltrometer data by numerical inversion, *Water Resour. Res., 32*(9), 2683-2696.

Šimůnek, J., and M. Th. van Genuchten. 1997. Estimating unsaturated soil hydraulic properties from multiple tension disc infiltrometer data, *Soil Science, 162*(6), 383-398.

Šimůnek, J., R. Angulo-Jaramillo, M. Schaap, J.-P. Vandervaere, and M. Th. van Genuchten. 1998. Using an inverse method to estimate the hydraulic properties of crusted soils from tension disc infiltrometer data, *Geoderma, 86*(1-2), 61-81.

Šimůnek, J., M. Th. van Genuchten, M. M. Gribb, and J. W. Hopmans. 1998. Parameter estimation of unsaturated soil hydraulic properties from transient flow processes, *Soil & Tillage Research, 47*, 27-36.

Šimůnek, J., R. Kodešová, M. M. Gribb, and M. Th. van Genuchten. 1999. Estimating hysteresis in the soil water retention function from modified cone penetrometer test, *Water Resour. Res.*, in press.

Šír, M., T. Vogel, and M. Císlerová. 1985. Analytical expression of the retention curve and hydraulic conductivity for porous material, *Vodohosp. Čas., 33*(1), 74-85 (in Czech).

Sisson, J. B. 1987. Drainage from layered field soils: Fixed gradient models, *Water Resour. Res., 23*(11), 2071-2075.

Skaggs, R. W., E. J. Monke, and L. F. Huggins. 1970. An approximate method for determining the hydraulic conductivity function of an unsaturated soil, *Techn. Rep. No. 11*, Water Resour. Res. Center, Purdue University, Lafayette, IN.

Sophocleous, M. 1979. Analysis of water and heat flow in unsaturated-saturated porous media, *Water Resour. Res., 15*(5), 1195-1206.

Spencer, W. F. 1991. Volatilization of pesticides from soil: processes and measurement, *Pesticide Res. J., 3*(1), 1-14.

Sposito, G. 1981. *The Thermodynamics of Soil Solutions*. Oxford University Press, New York, NY.

Stumm, W., and J. J. Morgan. 1981. *Aquatic Chemistry: An Introduction Emphasizing Chemical Equilibria in Natural Waters*, John Wiley & Sons, New York, NY.

Sudicky, E. A., and P. S. Huyakorn. 1991. Contaminant migration in imperfectly known heterogeneous groundwater systems, *Review of Geophysics*, Supplement, U. S. National Rep. to Inter. Union of Geodesy and Geophysics 1987-1990, 240-253.

Tillotson, W. R., C. W. Robbins, R. J. Wagenet, R. J. Hanks. 1980. Soil water, solute and plant growth simulation, *Bulletin 502,* Utah Agricultural Experiment Station, 53 p.

Toride, N., F. J. Leij, and M. Th. van Genuchten. 1993. A comprehensive set of analytical solutions for nonequilibrium solute transport with first-order decay and zero-order production, *Water Resour. Res., 29*(7), 2167-2182.

Toride, N., F. J. Leij, and M. Th. van Genuchten. 1995. The CXTFIT code for estimating transport parameters from laboratory or field tracer experiments, Version 2.0, *Research Report No 137*, U.S. Salinity laboratory, USDA, ARS, Riverside, California.

van Dam, J. C., J. N. M. Stricker, and P. Droogers. 1992. Inverse method for determining soil hydraulic functions from one-step outflow experiment, *Soil Sci. Soc. Am. J., 56*, 1042-1050.

van Dam, J. C., J. N. M. Stricker, and P. Droogers. 1994. Inverse method to determine soil hydraulic functions from multistep outflow experiment, *Soil Sci. Soc. Am. J., 58*, 647-652.

van Genuchten, M. Th. 1976. On the accuracy and efficiency of several numerical schemes for solving the convective-dispersive equation, in *Finite Elements in Water Resources*, edited by W. G. Gray et al., Pentech Press, London, pp. 1.71-1.90.

van Genuchten, M. Th. 1978. Mass transport in saturated-unsaturated media: one-dimensional solutions, *Research Rep. No. 78-WR-11*, Water Resources Program, Princeton Univ., Princeton, NJ.

van Genuchten, M. Th. 1980. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils, *Soil Sci. Soc. Am. J., 44*, 892-898.

van Genuchten, M. Th. 1981. Non-equilibrium transport parameters from miscible displacement experiments. *Research Report No. 119*, U.S. Salinity Laboratory, Riverside, CA.

van Genuchten, M. Th. 1985. Convective-dispersive transport of solutes involved in sequential first-order decay reactions, *Computers & Geosciences, 11*(2), 129-147.

van Genuchten, M. Th., 1987. A numerical model for water and solute movement in and below the root zone. *Research Report No 121*, U.S. Salinity laboratory, USDA, ARS, Riverside, California.

van Genuchten, M. Th., and J. Parker. 1984. Boundary conditions for displacement experiment through short laboratory soil columns, *Soil Sci. Soc. Am. J., 48*, 703-708.

van Genuchten, M. Th., and R. J. Wagenet. 1989. Two-site/two-region models for pesticide transport and degradation: Theoretical development and analytical solutions, *Soil Sci. Soc. Am. J., 53*, 1303-1310.

Vimoke, B. S., and G. S. Taylor. 1962. Simulating water flow in soil with an electric resistance network, *Report No. 41-65*, 51 p., Soil and Water Conserv. Res. Div., U. S. Agric. Res. Serv., Columbus, OH.

Vimoke, B. S., T. D. Tura, T. J. Thiel, and G. S. Taylor. 1963. Improvements in construction and use of resistance networks for studying drainage problems, *Soil Sci. Soc. Am. Proc., 26*(2), 203-207.

Vogel, T. 1987. SWMII - Numerical model of two-dimensional flow in a variably saturated porous medium, *Research Rep. No. 87*, Dept. of Hydraulics and Catchment Hydrology, Agricultural Univ., Wageningen, The Netherlands.

Vogel, T., M. Císlerová. 1988. On the reliability of unsaturated hydraulic conductivity calculated from the moisture retention curve, *Transport in Porous Media, 3*, 1-15.

Vogel, T., M. Císlerová, and J. W. Hopmans. 1991. Porous media with linearly variable hydraulic properties, *Water Resour. Res., 27*(10), 2735-2741.

Vogel, T., K. Huang, R. Zhang, and M. Th. van Genuchten. 1996. The HYDRUS code for simulating one-dimensional water flow, solute transport, and heat movement in variably-saturated media, Version 5.0, *Research Report No 140*, U.S. Salinity laboratory, USDA, ARS, Riverside, CA.

Wagenet, R. J., J. W. Biggar, and D. R. Nielsen. 1976. Analytical solutions of miscible displacement equations describing the sequential microbiological transformations of urea, ammonium and nitrate, *Research Rep. no 6001,* Dept. of Water Science and Engineering, Univ. California, Davis, CA.

Wagenet R. J., and J. L. Hutson. 1987. LEACHM: Leaching Estimation And CHemistry Model, A process-based model of water and solute movement, transformations, plant uptake and chemical reactions in the unsaturated zone, *Continuum 2*, Dept. of Agronomy, Cornell University, Ithaca, New York, NY.

Wesseling, J. G., and T. Brandyk. 1985. Introduction of the occurrence of high groundwater levels and surface water storage in computer program SWATRE, *Nota 1636*, Institute for Land and Water Management Research (ICW), Wageningen, The Netherlands.

Yeh, G. T., and V. S. Tripathi. 1990. HYDROGEOCHEM: A coupled model of HYDROlogic transport and GEOCHEMical equilibria in reactive multicomponent systems, *Environs Sci. Div., Publ. No. 3170.*, Oak Ridge National Lab., Oak Ridge, TN.

Zienkiewicz, O.C. 1977. *The Finite Element Method*, 3rd ed., McGraw-Hill, London, United Kingdom.

PART B

Interactive Graphics-Based User Interface

HYDRUS2D

Version 2.0

by

J. Šimůnek, M. Šejna, and M. Th. van Genuchten

B.1. *Introduction*

Even with well-documented multi-dimensional numerical computer models available, one major problem often preventing the use of such codes is the extensive work required for data preparation, finite element grid design, and graphical presentation of the output results. Hence, the more widespread use of multi-dimensional models requires techniques which make it easier to create, manipulate and display large data files, and which facilitate interactive data management. Introducing such techniques will free users from cumbersome manual data processing, and should enhance the efficiency in which programs are being implemented for a particular example. To avoid or simplify the preparation and management of relatively complex input data files for two-dimensional (or three-dimensional axi-symmetric) applications, and to graphically display the final simulation results, we developed an interactive graphics-based user-friendly interface (HYDRUS-2D) for the MS Windows 95, 98, and NT environments. The interface is connected directly to the HYDRUS2 code (which is an update of the CHAIN_2D code). The current HYDRUS2D version 2.0 graphical interface represents a complete rewrite of version 1.0 into a fully 32-bit environment.

As was discussed in the introduction, the program is distributed in two options. The complete interactive graphics-based user interface HYDRUS-2D (i.e., option B) includes seven modules (Table B.1), including the MESHGEN2D module which may be used to define virtually any two-dimensional geometric transport domain and subsequently to design a finite element discretization for that domain. MESHGEN2D is not included with Option A. Within option A, users can either employ the internal mesh generator for a simple rectangular structured transport domain, or are themselves responsible for creating the input files describing the domain geometry and associated finite element mesh. The user interface itself automatically handles all data management and users are not involved in writing the input files.

In addition to the information given in this chapter, extensive context-sensitive on-line help is made part of every module of the interface. By pushing the F1 button or clicking on the Help button while working in any window, the user obtains information about the window content. In addition, context-sensitive help is available in every module using the "SHIFT+F1" help button. In this mode, the mouse cursor changes to a help cursor (a combination arrow + question mark), and the user proceeds to click on the object for which he needs help (i.e, a menu item, toolbar button, or other features). At that point, a help file will be displayed giving information about the item on which the user clicked. Except for the FORTRAN application itself, all modules are written in C++.

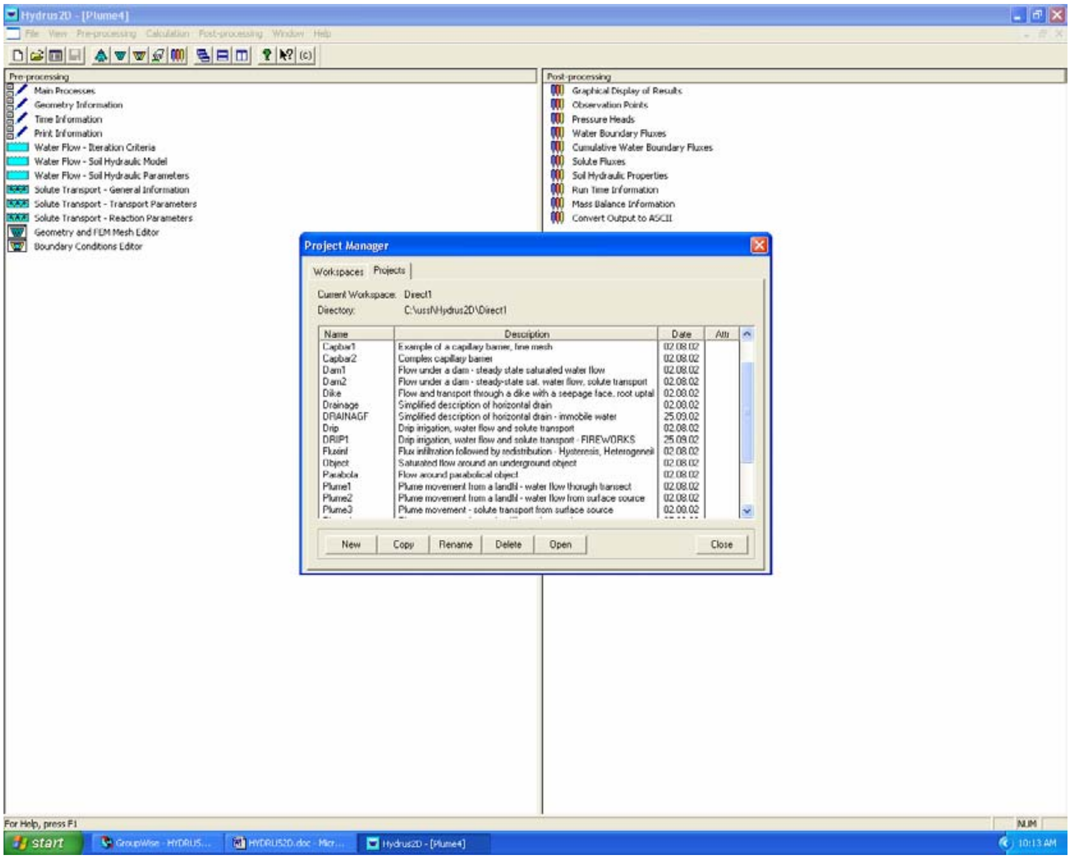
Table B.1. Main modules of the HYDRUS-2D software package

HYDRUS2D	main program unit
PROJECT MANAGER	project manager
MESHGEN2D	transport domain geometry and finite element mesh generator
H2D_BDRC (Boundary)	boundary and initial conditions
H2D_CALC (HYDRUS2)	FORTRAN application for direct solution
H2D_CLCI (HYDRUS2)	FORTRAN application for inverse solution
H2D_GRAF (Graphics)	output results presentation

B.2. *Brief Description of Selected Modules*

B.2.1. *Module **HYDRUS2D***

HYDRUS2D (Fig. B.1) is the main program unit defining the overall computational environment of the system. This module controls execution of the program and determines which other optional modules are necessary for a particular application. The module contains a project manager and both the pre-processing and post-processing units. The pre-processing unit includes specification of all necessary parameters to successfully run the **HYDRUS2** FORTRAN code (modules H2D_CALC and/or H2D_CLCI), a grid generator for relatively simple rectangular transport domains, and a small catalog of soil hydraulic properties. The post-processing unit consists of simple x-y graphics for graphical presentation of soil hydraulic properties, as well as such output as distribution versus time of a particular variable at selected observation points, and actual or cumulative water and solute fluxes across boundaries of a particular type.


Hydrus2D - [Plume4]
File View Pre-processing Calculation Post-processing Window Help
Pre-processing
Main Processes
Geometry Information
Time Information
Print Information
Water Flow - Iteration Criteria
Water Flow - Soil Hydraulic Model
Water Flow - Soil Hydraulic Parameters
Solute Transport - General Information
Solute Transport - Transport Parameters
Solute Transport - Reaction Parameters
Geometry and FEM Mesh Editor
Boundary Conditions Editor
Post-processing
Graphical Display of Results
Observation Points
Pressure Heads
Water Boundary Fluxes
Cumulative Water Boundary Fluxes
Solute Fluxes
Soil Hydraulic Properties
Run Time Information
Mass Balance Information
Convert Output to ASCII
Project Manager
Workspaces
Projects
New
Copy
Rename
Delete
Open
Close
For Help, press F1
NUM
start
GroupWise - HYDRUS...
HYDRUS2D.doc - Micr...
Hydrus2D - [Plume4]
10:13 AM

Current Workspace: Direct1
Directory: C:\ussl\Hydrus2D\Direct1

Name	Description	Date	Attr
Capbar1	Example of a capillary barrier, fine mesh	02.08.02	
Capbar2	Complex capillary barrier	02.08.02	
Dam1	Flow under a dam - steady state saturated water flow	02.08.02	
Dam2	Flow under a dam - steady-state sat. water flow, solute transport	02.08.02	
Dike	Flow and transport through a dike with a seepage face, root uptal	02.08.02	
Drainage	Simplified description of horizontal drain	02.08.02	
DRAINAGF	Simplified description of horizontal drain - immobile water	25.09.02	
Drip	Drip irrigation, water flow and solute transport	02.08.02	
DRIP1	Drip irrigation, water flow and solute transport - FIREWORKS	25.09.02	
Fluxinf	Flux infiltration followed by redistribution - Hysteresis, Heterogeneil	02.08.02	
Object	Saturated flow around an underground object	02.08.02	
Parabola	Flow around parabolical object	02.08.02	
Plume1	Plume movement from a landfill - water flow thorugh transect	02.08.02	
Plume2	Plume movement from a landfill - water flow from surface source	02.08.02	
Plume3	Plume movement - solute transport from surface source	02.08.02	

Fig. B.1. The main window of the HYDRUS-2D module, including the project manager.

Work for a new project should begin by opening the Project Manager (see Chapter B.2.2), and giving a name and brief description to this new project. Then select the Main Processes command from the Pre-processing Menu. From this point on, the program will navigate the user through the entire process of entering input files. The user may either select particular commands from a menu, or allow the interface to lead him or her through the process of entering input data by selecting the Next buttons. Alternatively, clicking the Previous button will return users to the previous window.

Table B.2 lists all commands accessibe through the menu, whereas Table B.3 gives a brief discussion of the action taken with the particular commands. More detailed descriptions are available through the on-line help. Table B.4 gives an overview of the different graph options made available through the main module HYDRUS2D.

Table B.2. Menu commands in the main module HYDRUS2D.

Group	Menu	Submenu	Sub-Submenu
A	File	New	
		Open	
		Project Manager	
		Save	
		Save As	
		Recent Files	
		Exit	
B	View	Toolbar	
		Status Bar	
		List Boxes for Inverse Data	
C	Pre-processing	Main Processes	
		Inverse Solution	
		Geometry Information	
		Time Information	
		Print Information	
		Water Flow	Iteration Criteria
			Hydraulic Properties Model
			Soil Hydraulic Properties
		Solute Transport	General Solute Transport Information
			Solute Transport Parameters
			Solute Reaction Parameters
			Temperature Dependence
		Heat Transport Parameters	
		Root Water Uptake	Root Water Uptake Models
			Pressure Head Reduction
			Osmotic Head Reduction
		Variable Boundary Conditions	
		Data for Inverse Solution	
		Geometry	
		FEM Mesh	
		Boundary Conditions	
D	Calculation	Execute HYDRUS2	
E	Results	Graphical Display of Results	
		Observation Points	
		Boundary Information	Pressure Heads
			Boundary Fluxes
			Cumulative Fluxes
			Solute Fluxes
		Soil Hydraulic Properties	
		Run Time Information	
		Mass Balance Information	
		Convert Output to ASCII	
		Inverse Solution	

Table B.2. (continued).

Group	Menu	Submenu	Sub-Submenu
F	Window	Cascade Tile Horizontally Tile Vertically Arrange Icons	
G	Help	Index Using Help Help On About HYDRUS-2D	

Table B.3. Description of all menu commands in the main module HYDRUS2D.

Group	Command	Brief description of the command
A	New	Creates a new project.
	Open	Opens an existing project (represented by *project_name*.h2d file)
	Project Manager	Calls the project manager to manage data of existing projects; helps to locate, open, copy, delete or rename the desired projects and their data.
	Save	Saves the input data of an actual project specified in the main program module if the data were either newly created or changed during an application run. This command deletes at the same time all existing output files of a selected project since the output data are no longer consistent with the changed input data. A warning is issued to the user before doing so.
	Save as	Saves data of a particular project under a new project name.
	Recent File	Lists the projects most recently opened in this program. To quickly reopen one of these projects, click it.
	Exit	Closes the project and leaves the program. This command informs the user before exiting the application whether or not the input data of an actual project were changed during the application run. If changes did occur, the user is given a possibility to save data before exiting the application.
B	Toolbar	Shows or hides the toolbar.
	Status Bar	Shows or hides the status bar.
	List Boxes for Inverse Data	Shows a text information in the inverse data list.
C	Main Processes	Selects the title, which is printed into output files, and specifies the processes to be simulated, i.e., water flow, multiple solute transport, heat transport, root growth, and/or root water uptake.
	Inverse Solution	Selects type of the weighting of measured data, and whether the soil hydraulic parameters, solute transport parameters, or both are to be fitted.

Table B.3. (continued).

Group	Command	Brief description of the command
	Main Geometry Information	Selects the length unit, specifies the depth and inclination of the soil profile to be analyzed, and determines the number of materials to be used.
	Main Time Information	Selects time units, and gives the time discretization information.
	Print Information	Specifies print options.
	Iteration Criteria	Specifies iteration criteria for the solution precision, and parameters for the time step control.
	Hydraulic Properties Model	Selects the type of model used for the soil hydraulic properties, and decides whether the hysteresis is to be considered.
	Soil Hydraulic Properties	Specifies parameters in the soil hydraulic model.
	General Solute Transport Information	Selects the time and spatial weighting schemes for numerical solution of the solute transport equation; specifies the number of solutes to be considered.
	Solute Transport Parameters	Specifies solute transport parameters.
	Solute Reaction Parameters	Specifies solute reaction parameters.
	Temperature Dependence	Specifies parameters, which define the temperature dependence of reaction and transport parameters.
	Heat Transport Parameters	Specifies heat transport parameters.
	Root Water Uptake Models	Selects the root water uptake stress response models for both salinity and water stress.
	Pressure Head Reduction	Specifies parameters in the root water uptake water stress response model.
	Osmotic Head Reduction	Specifies parameters in the root water uptake salinity stress response model.
	Variable Boundary Condition	Specifies time-dependent boundary conditions for all transport processes.
	Data for Inverse Solution	Specifies data for the inverse solution, their type, location and associated weight.
	Geometry	Calls external module MESHGEN2D, for users to design and discretize the transport domain.
	Boundary Conditions	Specification of boundary conditions, initial conditions, parameters which describe the properties of the transport domain, and/or observation nodes.
D	Execute HYDRUS2	Executes a HYDRUS2 version 2.0 FORTRAN application.
E	Graphical Display of Results	Graphical presentation of output results (pressure heads, water contents, temperatures, and/or concentrations) in terms of contour and spectral maps, velocity vectors, animation of both contour and spectral maps.
	Observation Points	Graphical presentation of changes in water content, pressure head, temperature, and/or solute and sorbed concentration at specified observation nodes.

Table B.3. (continued).

Group	Command	Brief description of the command
E	Boundary Information	Graphical presentation of actual and cumulative boundary water and solute fluxes, and surface, root zone, and bottom pressure heads and concentrations.
	Soil Hydraulic Properties	Graphical presentation of the soil hydraulic properties.
	Run Time Information	Graphical presentation of information about the number of iterations, time step, and Peclet and Courant numbers.
	Mass Balance Information	Displays mass balance information and mean profile properties
	Convert to ASCII	Convert binary input and output files into ASCII files.
	Inverse Solution	Displays information about the inverse solution.
G	Index	Offers an index of topics for which help is available.
	Using Help	Provides general instructions on using help.
	About HYDRUS-2D	Displays the version and authors of the HYDRUS-2D application.

Table B.4. Graph options in the HYDRUS-2D interface.

Command	Horizontal Axis	Vertical Axis
Observation Points	Time	Pressure Head Water Content Temperature Concentration – x^+ Sorbed Concentration - x^+
Pressure Heads	Time	Atmospheric Boundary Head Root Zone Head Variable Boundary Head Constant Boundary Head Seepage Face Head Drainage Boundary Head Free and Deep Drainage Boundary Head
Water Boundary Fluxes	Time	Potential Atmospheric Flux Potential Root Water Uptake Rate Actual Atmospheric Flux Actual Root Water Uptake Rate Constant Boundary Flux Variable Boundary Flux Seepage Face Flux Drainage Boundary Flux Free and Deep Drainage Boundary Flux

Table B.4. (continued).

Command	Horizontal Axis	Vertical Axis
Cumulative Water Boundary Fluxes	Time	Potential Atmospheric Flux Potential Root Water Uptake Rate Actual Atmospheric Flux Actual Root Water Uptake Rate Constant Boundary Flux Variable Boundary Flux Seepage Face Flux Drain Boundary Flux Free and Deep Drainage Boundary Flux
Solute Fluxes[+]	Time	Cumulative Zero-Order Reaction Cumulative First-Order Reaction Cumulative Root Solute Uptake Cumulative Non-Equil. Mass Transfer Cumulative Constant Boundary Solute Flux Cumulative Seepage Face Solute Flux Cumulative Variable Boundary Solute Flux Cumulative Atmospheric Solute Flux Cumulative Drain Boundary Solute Flux Cum. Free/Deep Drainage Bound. Solute Flux Constant Boundary Flux Seepage Face Flux Variable Boundary Flux Actual Atmospheric Flux Drain Boundary Flux Free and Deep Drainage Boundary Flux
Soil Hydraulic Properties	Pressure Head Log Pressure Head Water Content	Water Content Soil Water Capacity Hydraulic Conductivity Log Hydraulic Conductivity Effective Water Content
Run-Time Information	Time Level Time	Time Step Number of Iterations Cumulative Number of Iterations Peclet Number Courant Number Number of Solute Iterations

[+] This graph is given for each solute

B.2.2. *Project Manager*

A **Project Manager** (Fig. B.1) is used to manage the data of existing projects, and helps to locate, open, copy, delete and/or rename desired projects or their input or output data. A "project" represents any particular problem to be solved by HYDRUS-2D. The project name (8 letters maximum), as well as a brief description of the project, helps to locate a particular problem. Input and output data for each project are placed in a subdirectory with the same name as the project. Projects are represented by a file *project_name.h2d* and the *project_name* subdirectory.

As compared to previous versions, the current version of the **Project Manager** gives users much more freedom in organizing his/her projects. The projects are grouped into **Workspaces** which can be placed anywhere in accessible memory, i.e., on local and/or network hard drives. The **Workspace** can be any existing accessible subdirectory (folder). HYDRUS-2D is installed together with two default workspaces, *Direct* and *Inverse*, which are located in the HYDRUS2D subdirectory. The *Direct* and *Inverse* workspaces contain test examples for direct and inverse problems. We suggest that the user creates his/her own workspaces, e.g., the *My Direct* and *My Inverse* workspaces, and keeps the provided examples intact for further reference. Projects can be copied with the **Project Manager** only within a particular workspace. The user can copy projects between workspaces using standard file managing software, e.g., Windows Explorer. In that case one must copy both the subdirectory of a particular project and the *project_name.h2d* file.

The Project Manager can also be used to connect the current version 2.0 of HYDRUS-2D with projects created with previous versions of HYDRUS-2D (e.g., version 1.0). This is done by creating a new **Workspace**, browsing for the location of projects created with a previous version of HYDRUS-2D, and checking the box "Connect a directory with HYDRUS-2D version 1.0 projects (16 bit)". The results of the older projects then can be viewed through the new version of HYDRUS-2D, input data can be converted into a new format, and/or projects can be recalculated.

Table B.5. Commands in the PROJECT MANAGER.

Group	Command	Description
Workspace	New	Registers a new workspace in the Project Manager. This command can also be used to connect the current version of HYDRUS-2D with projects created with a previous version of HYDRUS-2D.
	Change	Renames selected workspace, changes its description and/or location.
	Remove	Removes registration of a selected workspace from the Project Manager.
	Set As Current	Sets a selected workspace as active workspace.
	Close	Closes the Project Manager.
Project	New	Creates a new project in an actual workspace.
	Copy	Copies a selected project within an actual workspace.
	Rename	Renames a selected project.
	Delete	Deletes a selected project.
	Open	Opens a selected project.
	Close	Closes the Project Manager.

B.2.3. *Module **MESHGEN2D***

The **MESHGEN2D** module (Fig. B.2) can be used to design boundary curves of virtually any two-dimensional computational domain in a MS Windows graphical environment. Boundary curves can consist of any combination of polylines, arcs, circles or cubical splines. The module permits one to specify internal boundaries (e.g., drains, wells, impermeable objects), as well as internal curves. This module also checks the consistency of the invoked geometry. Boundary curves can be entered from a keyboard or with a mouse, or by reading an ASCII file.

The computational transport *domain* (see definition in Table B.5) is formed by an arbitrary number of mutually nonintersecting *curves* (Fig. B.3). Each curve can be formed by connecting an arbitrary number of *objects*. Objects are given in terms of nodes, the positions of which are defined either by a graphical cursor (with a possibility to use alignment grid), or by defining Cartesian coordinates X and Y. It is possible to read the objects defined with a large number of nodes (spline, polyline) from a file. The order of inputting particular objects is arbitrary; the code automatically forms the desired curves. In order to have a reasonable domain for the purpose of transport calculations, only one closed *outer curve* can exist (for multicomponent domains such a curve must exist for each subregion of the domain). The domain can have an arbitrary number of *holes* or *internal curves*. The consistency of the geometry can be verified at any time using the command

Check Consistency. Any change in geometry can be undone by using the command (button) undo (up to ten levels backward in time), or redone by using the command redo (again up to 10 levels back).

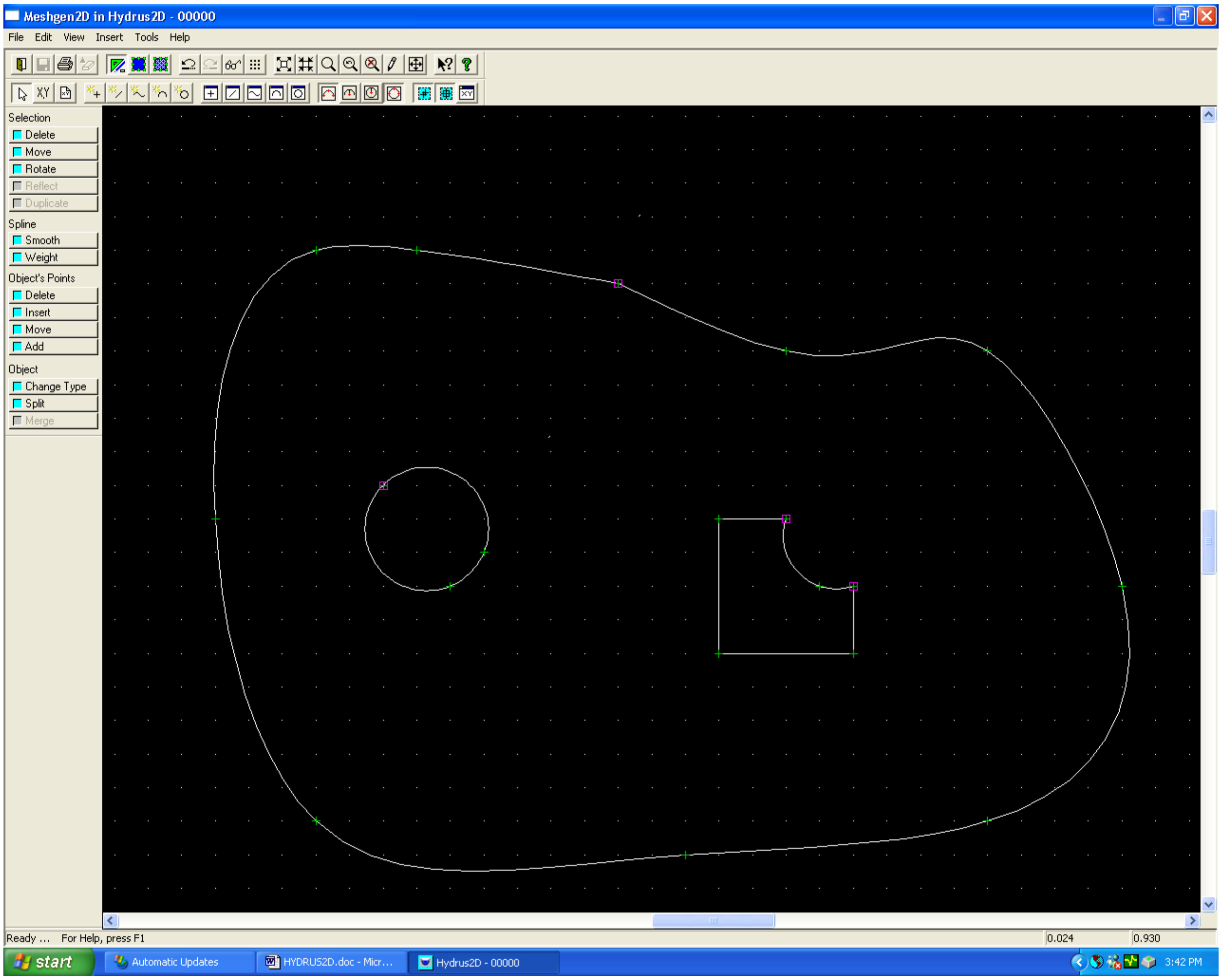


Fig. B.2. The main window of the MESHGEN2D module, geometry design.

New points defining a particular object should be entered with the mouse or from the keyboard using the arrow buttons. The manner in which nodes are entered depends on the selected input style. When entering data graphically, a grid with appropriately set parameters can be used to facilitate the input; while doing this, the coordinates X and Y of the cursor are continuously being displayed in the bottom right corner of the window.

Existing objects can be edited in several ways. Almost all edit actions start with the command select object, followed by the desired operation. The current selection, displayed in yellow, may be modified (edited) using the following operations: delete, copy, move, rotate and reflect. Note that, in addition to objects, particular nodes of an object can belong to a selected set as well, in which case the edit operations are carried out also for these points. Editing of selected objects (e.g., moving objects) depends on the currently selected input style. The objects are moved with the cursor when the graphical mode is selected, whereas a vector of translation (X, Y) must be specified in the numerical mode. It is possible to directly edit nodes of objects with the commands insert point, delete point, and/or move point.

Before saving the data, an option is always displayed whether or not the consistency of the geometry is to be verified. We strongly recommend to always perform this check in order to prevent errors in subsequent calculations, e.g., during mesh generation.

The **MESHGEN2D** (Fig. B.4) module may also be used to discretize a two-dimensional flow region into an unstructured triangular mesh. The algorithm used for this purpose is general and can be applied to virtually any two-dimensional computational domain. The first step of the mesh generation process is discretization of the boundary curves, while the second step is generation of the unstructured triangular mesh.

Table B.6. Definition of terms in the MESHGEN2D module (related to geometry design).

Computational Domain	The term domain is defined by a finite number of continuous disjunctively bounded two-dimensional subdomains. This means that the domain can be multicomponent, but that each of its components must have only one outer boundary curve. The domain can contain any finite number of internal holes or internal curves.
Curve	A curve is a set of a finite number of objects connected by boundary nodes (i.e., from each mutually shared boundary node, only two objects can start). Except at this node, the objects cannot intersect each other nor themselves. A curve can be open or closed. When a curve is closed, the code automatically lines up objects and their nodes counter-clockwise, i.e., the curve is positively oriented.
Objects	The following types of objects can be used: polyline, arc, circle, spline, single points

Table B.6. (continued)

Outer Boundary Curve	An outer boundary curve is a boundary curve with the following properties: the curve is closed, positively oriented (i.e., in a counter-clockwise direction), does not intersect any other curve or itself, and has the computational domain on its left side (in the sense of positive orientation), while the right side is not part of the computational domain.
Internal Hole	An internal hole is a boundary curve which has the following properties: the curve is closed, positively oriented (in a counter-clockwise direction), does not intersect any other curve or itself, and has the computational domain located on its right-hand side (in the sense of positive orientation), while the left side is not part of the computational domain.
Internal Curve	An internal curve has a computational domain on both sides. An internal curve can be open or closed, but cannot intersect any other curves or itself.
Single Points	A single point is located inside of a domain and is not located on any curve.
Line	Lines and polylines are the most commonly used objects for describing boundaries of the domain or internal curves. Lines are defined by two points defining their beginning and end. Several lines can be connected to form "polylines" by associating the beginning and end of two neighboring lines. The lines can not intersect each other.
Spline	A spline is a set of more than two points connected by cubic arcs. Cubic splines can either interpolate exactly, or approximately, between given points (i.e., they pass exactly through points, or deviate from them). This choice depends on the value of the standard deviation and the weight of a spline point. A spline cannot intersect itself, and its two boundary points must have a relatively large enough weight to be located exactly on the spline (otherwise problems may arise when connecting a spline to other objects).
Arc	An arc is defined either by three points on its circumference, or by a center, a radius, two angles (starting and final angle), and its orientation.
Circle	A circle is defined either by a center and a radius, or by three different points on its circumference.

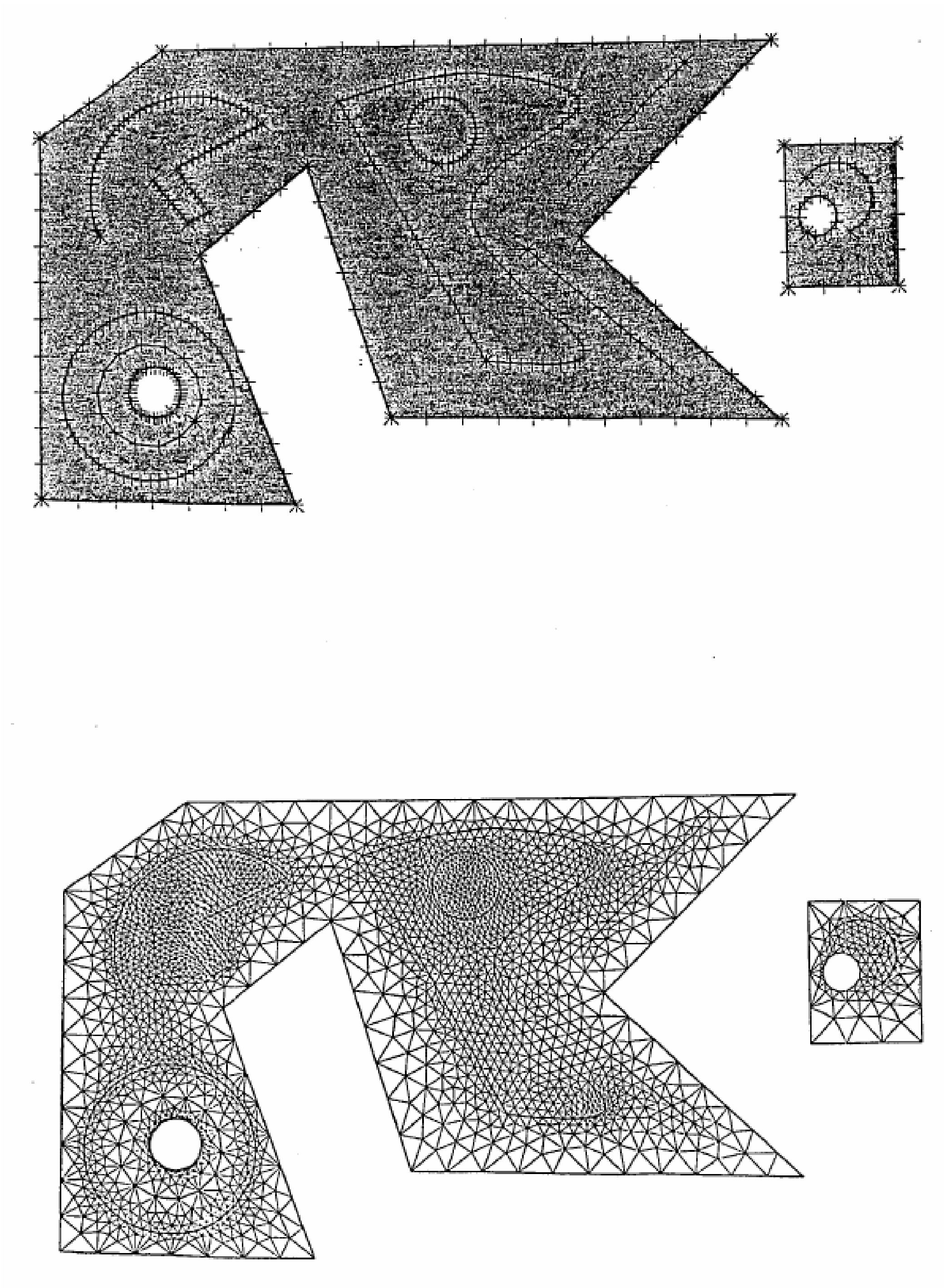


Fig. B.3. Example of the mesh for a multicomponent domain with internal curves and holes.

Generation of Boundary Points

The first step of the mesh generation process involves discretization of the boundary curves, i.e., boundary nodes are generated on all boundaries and internal curves by dividing them in abscissas, i.e., short boundary edges (Fig. B.3). If no previous boundary nodes exist, the program automatically generates a default equidistant point distribution. Boundary nodes can be edited by the user to optimize the lengths of the boundary edges. The local density of the mesh can thereby be determined in any part of the domain (also taking into account the use of internal curves). There are two ways of specifying appropriate distributions of the boundary nodes, i.e., by (1) editing the number of points on a curve, and (2) specifying fixed points and nodal densities. The nodal density determines the relative length of boundary edges, and can be specified only at fixed points. Fixed points can be inserted or deleted anywhere on a boundary. The user can edit the nodal density at a fixed point in order to locally refine the nodal distribution around this point.

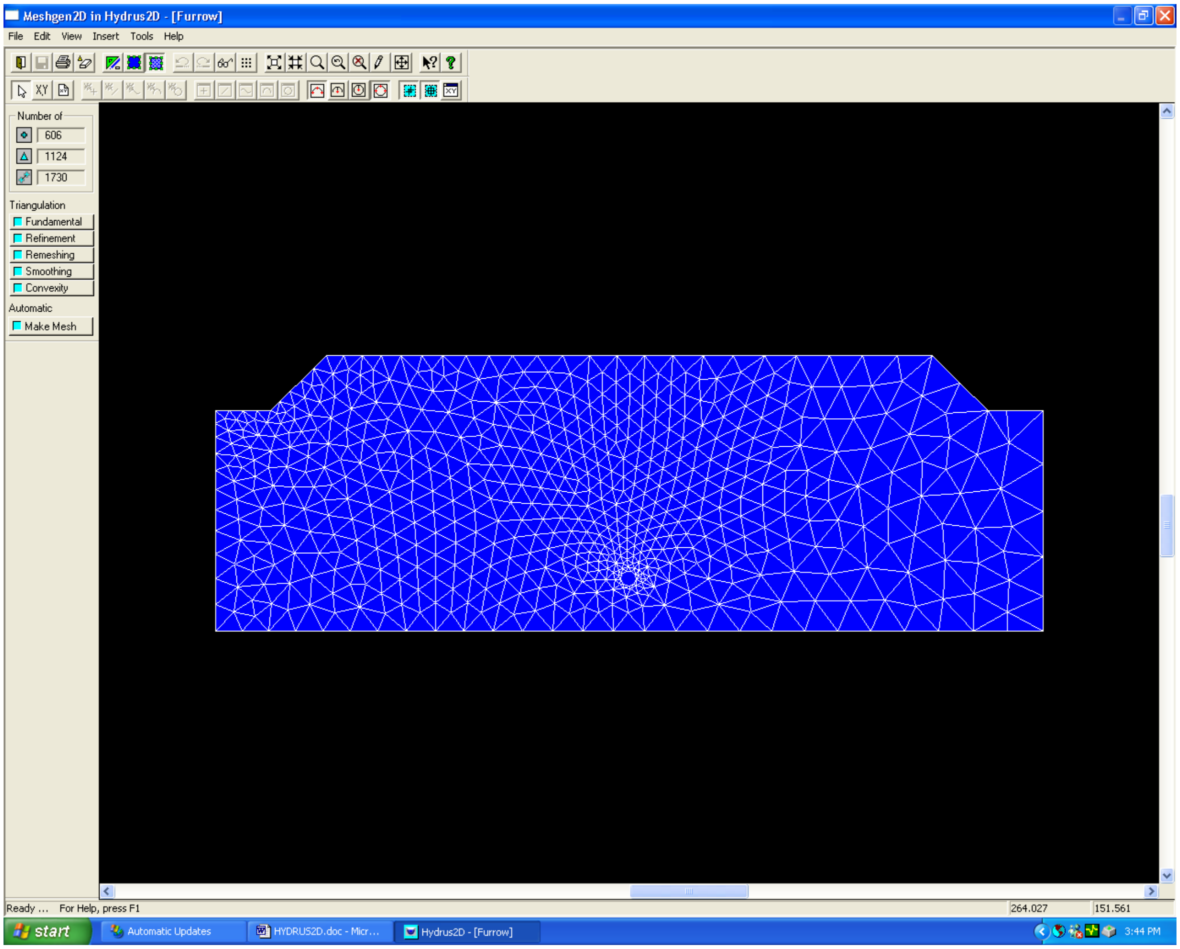
The boundary nodal distribution determines in a very substantial manner the ultimate quality and size of the unstructured finite element mesh. Optimally distributing nodes along the boundaries of relatively complicated domains (e.g., a strongly irregular, anisotropic domain) can be a very difficult problem, and may require some experience.

Table B.7. Definition of terms in the MESHGEN2D module (related to the boundary discretization).

Boundary Nodes	Boundary nodes are points which discretize boundary curves and which are marked by green crosses. These nodes are generated along every boundary curve and are ordered in a counter-clockwise direction (on closed curves). Boundary nodes determine the local densities of the triangular mesh that is being generated for a given boundary nodal distribution, and are part of the triangular mesh.
Boundary Edges	Boundary edges are abscissas discretizing boundary curves. They connect generated boundary nodes, are oriented in counter-clockwise direction, and are located on the edge of the mesh.
Fixed Points	Fixed points are points on boundary curves marked by purple stars. These points may be used to adjust the local density of boundary nodes. By default, fixed points are placed on all nodes of polylines and on all object boundary points describing the boundary, but they can be also inserted or deleted at any other point on boundary curves.

Table B.7. (continued)

Nodal Density	The nodal density is a real number in the range <0.01, 100.> characterizing the local density of boundary points. The density can be specified only at fixed points. The program distinguishes between left and right density. The left (right) density at a fixed boundary point specifies the relative length of the boundary edge, which ends (or begins) at this point (boundary edges are oriented in a counter-clockwise direction). If the left and right densities are equal then the nodal density is continuous, i.e., both edges have the same length. If the left and right density values are different then the lengths will be different as well. For example: if DL = 3., DR = 0.1 then LL/LR = 30, where DL and DR are the left and right densities at a fixed point, respectively, and LL and LR are the lengths of boundary edges which end and begin at that fixed point, respectively.



Number of
606
1124
1730

Ready ... For Help, press F1 264.027 151.561

Fig. B.4. The main window of the MESHGEN2D module, mesh generator.

Generation of the Unstructured Triangular Mesh

Generation of the unstructured triangular mesh consists of five operations: (1) discretization of the flow domain into triangles with vertices at given boundary nodes (*fundamental triangulation*), (2) inserting new points in all triangles which do not fulfill a certain smoothness criterion (*mesh refinement*), (3) implementation of Delaunay's retriangulation for the purpose of eliminating all nodes surrounded by more than six triangles, as well as for avoiding extreme angles (*remeshing*), (4) smoothing of the mesh by solving a set of coupled elliptic equations in a recursive algorithm (*smoothing*), and (5) correction of possible errors which may appear during smoothing of the finite element mesh (*convexity check*). Operations 2 through 5 are repeated until a prescribed smoothness of the mesh has been achieved.

Generation of the unstructured triangular mesh for a given boundary nodal distribution is possible in two different ways: a step-by-step approach, or by using automatic mesh generation. The step-by-step approach should be used only for special cases, and then only by an experienced user. Automatic generation (recommended option) is a much faster and easier approach. The mesh generation parameters can be specified before the mesh generation process is started. By modifying the mesh generation parameters, the user can influence the smoothness of the mesh, its anisotropy (Fig. B.5), computational time, and the possible display of intermediate results, among other features. The most important mesh generation parameter is the "smoothing factor", which can significantly affect the final number of elements. The smoothing factor is defined as the ratio of the minimum and maximum dimensions of a triangle. When a very smooth finite element mesh is required, the smoothing factor can be decreased to about 1.1; when a coarse mesh can be tolerated, the smoothing factor can be increased. An example of an automatically generated finite element mesh is presented in Figures B.6 and B.7.

The code saves mesh data, by default, in a binary format (the file MESHGEN2.PMG). However, the user has the option to also save mesh data into a file MESHTRIA.TXT in ASCII format, and use the data later for other calculations.

When the module MESHGEN2D is not present (Option A), the user can either employ the internal mesh generator for a simple rectangular transport domain, or is herself responsible for creating the input file MESHTRIA.TXT giving positions of nodal points and definitions of the finite elements. The structure of the MESHTRIA.TXT file is described in Table 10.7.

Table B.8 lists all commands accessible through the menu in the MESHGEN2D module, whereas Table B.9 gives a brief discussion of actions taken with particular commands. More detailed descriptions are available through the on-line help features.

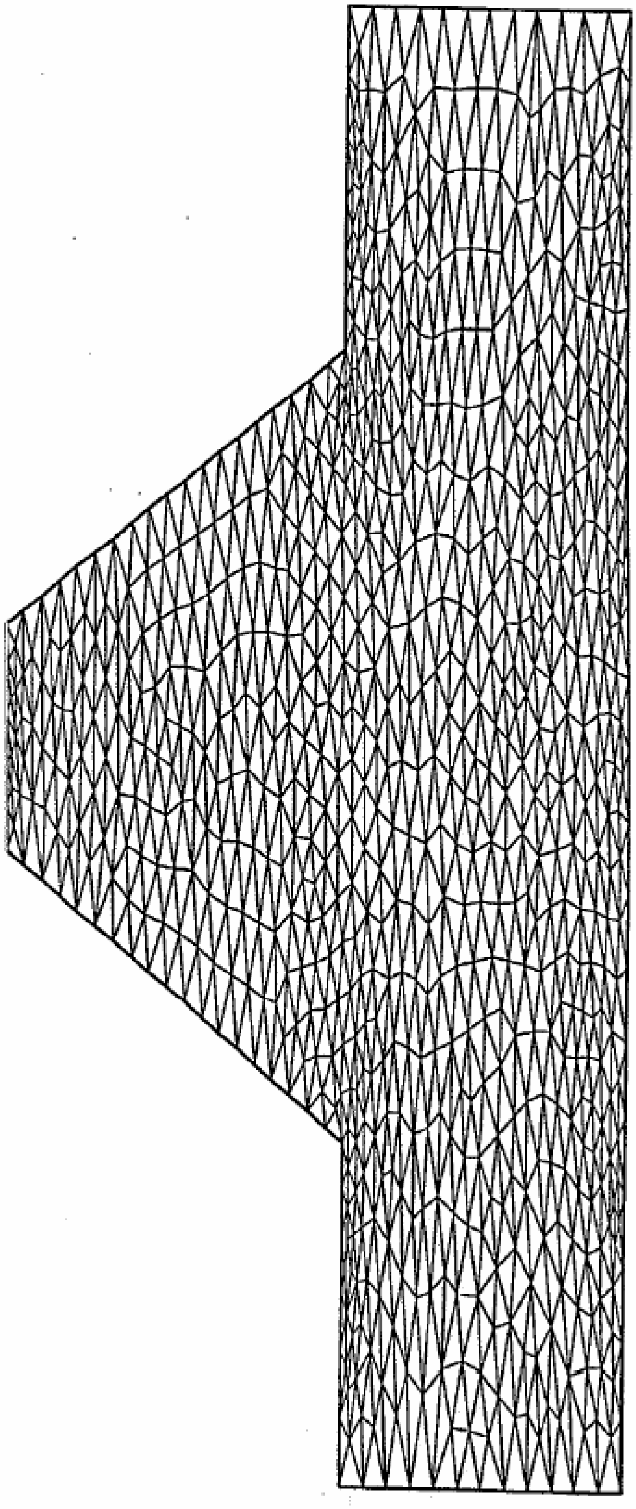


Fig. B.5. Example of mesh stretching. Stretching factor Sf=3 (in the *y*-direction).

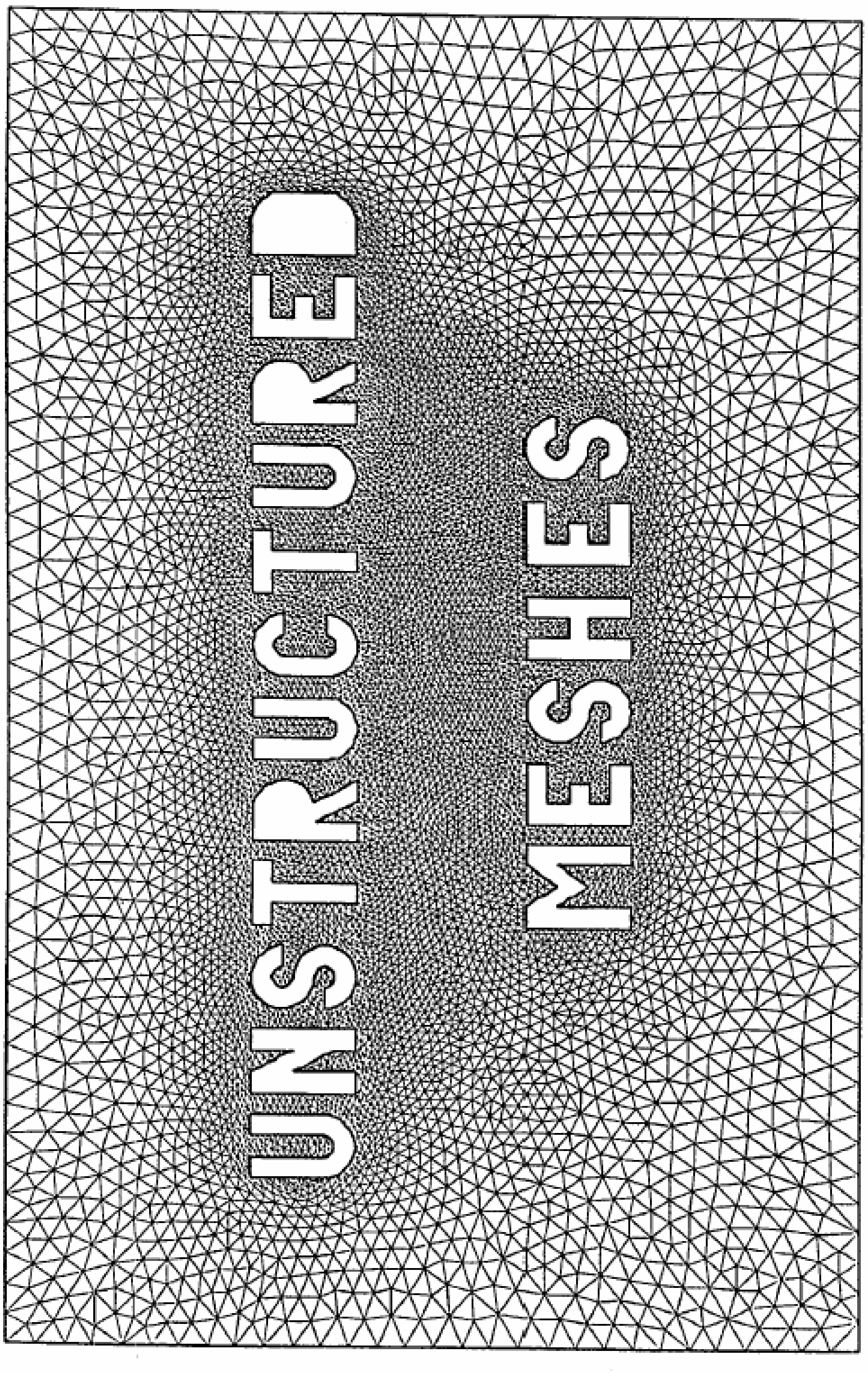


Fig. B.6. Example of the unstructured mesh for a complex geometry.

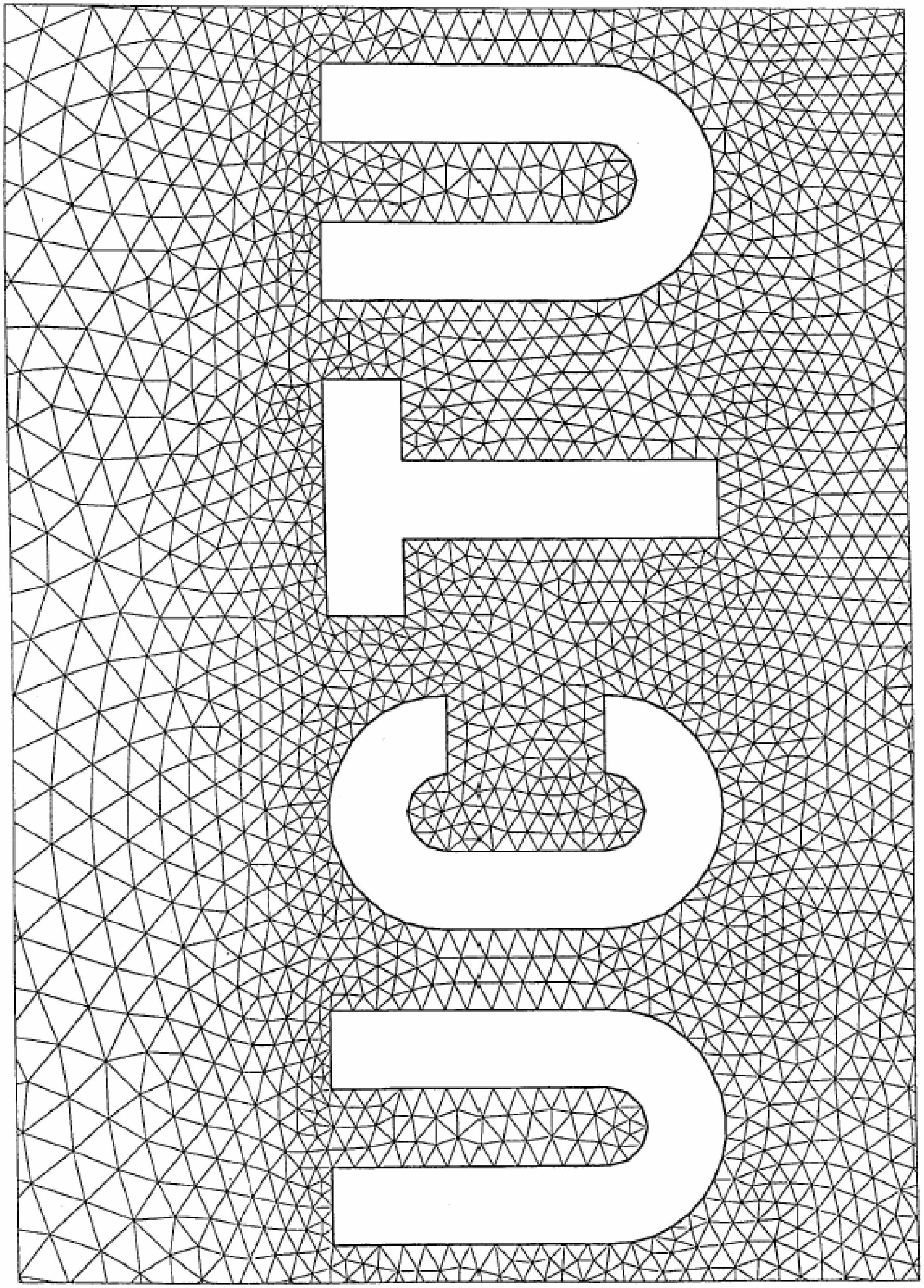


Fig. B.7. Detail of the mesh shown in Figure B.6.

Table B.8. Menu commands in the MESHGEN2D module.

Group	Menu	Submenu	Sub-Submenu	Sub-Sub-Submenu
A	File	Save		
		Print		
		Print Preview		
		Print Setup		
		Copy to Clipboard		
		Save as Text File		
		Close and Return		
B	Edit	Undo		
		Redo		
		Delete All		
		Delete	Selected Objects	
			Fixed Point	
			Periodical Conditions	
			Mesh	
		Select	Select By Rhomboid	
			Always Select/Deselect	
		Geometry	Split Object	
			Merge Two Objects	
			Change Object Type	
			Smooth Spline	
			Object's Points	Add
				Insert
				Delete
				Move
				Coordinates
				Spline Point Weight
			Manipulate Selection	Copy
				Move
				Reflect
				Rotate
		Boundary Points	Default Discretization	
			Change Boundary Type	
			Number of Points	
			Density at Fixed Points	
		Mesh	Generator Parameters	
			Automatic Generation	
			Fundamental Triangulation	
			Mesh Refinement	
			Delaunay Retriangulation	
			Smoothing	
			Convexity Check	
C	View	Geometry		
		Boundary Points		
		FEM Mesh		
		Toolbars		

Table B.8. (continued).

Group	Menu	Submenu	Sub-Submenu
		Status Bar	Display Options
		View All	
		Enlarge View	
		Reduce View	
		Zoom	
		Previous Zoom	
		Viewport	
		Stretching	
		Redraw	
D	Insert	Single Points	Graphically
			Numerically
		Line	Graphically
			Numerically
		Spline	Graphically
			Numerically
		Arc	Graphically
			Numerically
		Circle	Graphically
			Numerically
		Object From File	
		Fixed Point	
		Periodical Condition	
E	Tools	Manipulation Mode	Graphical
			Numerical
		Circle Input Mode	Center and Radius
			Three Points
		Arc Input Mode	Center, Radius and Two Angles
			Three Points
		Point Picking	
		Overlapped Points	
		Grid Setting	
		Check Geometry	
		Find	Node
			Edge
			Triangle
F	Help	Index	
		Using Help	
		Help On	
		About MESHGEN2D	

Table B.9. Description of all menu commands in the MESHGEN2D module.

Group	Command	Brief description of the command
A	Save	Saves the geometry and finite element mesh information in a binary file.
	Print	Prints a figure displayed in the graphical window.
	Print Preview	Displays a figure in the view window on the screen as it would appear printed.
	Print Setup	Selects a printer and printer connection.
	Copy to Clipboard	Copies the content of the graphical window to a Windows metafile in the clipboard.
	Save as Text File	Saves the geometry and finite element mesh information in an ASCII file.
	Close and Return	Exits the MESHGEN2D module.
B	Undo	Reverses the last edit actions.
	Redo	Repeats the last edit actions.
	Delete All	Deletes all data realted to the geometry of the transport domain.
	Delete->Selected Objects	Deletes selected objects.
	Delete->Fixed Point	Deletes fixed points.
	Delete->Periodical Condition	Deletes periodical condition.
	Delete->Mesh	Deletes the finite element mesh.
	Select->Select By Rhomboid	Selects objects using rhomboid.
	Select->Always Select/Deselect	
	Geometry->Split Objects	Splits an object into two parts.
	Geometry->Merge Two Objects	Merges two objects into one.
	Geometry->Change Object Type	Changes the type of an object.
	Geometry->Smooth Spline	Smoothes spline.
	Geometry->Object's Point->Add	Adds new points to existing objects (lines, splines).
	Geometry->Object's Point->Insert	Inserts new points in existing objects (lines, splines).
	Geometry->Object's Point->Delete	Deletes points from existing objects (lines, splines).
	Geometry->Object's Point->Move	Moves points on existing objects (lines, splines).
	Geometry->Object's Point->Coordinates	Edits coordinates of selected nodes on existing objects (lines, splines).
	Geometry->Object's Point->Spline Point	Weight Edits weight of the node on an existing spline.
	Geometry->Manipulate Selection->Copy	Copies selected objects.
	Geometry->Manipulate Selection->Move	Moves selected objects.
	Geometry->Manipulate Selection->Reflect	Reflects selected objects.
	Geometry->Manipulate Selection->Rotate	Rotates selected objects.
	Boundary Points->Default Discretization	Sets the default boundary distribution.
	Boundary Points->Change Boundary Type	Specifies if an internal line is an internal hole or not.
	Boundary Points->Number of Points	Specifies the number of nodes on a selected boundary curve.
	Boundary Points->Density at Fixed Point	Selects nodal density at a fixed point.
	Mesh->Generator Parameters	Edits parameters of mesh generation process.
	Mesh->Automatic Generation	Start automatic generation of finite element mesh.
	Mesh->Fundamental Triangulation	Performs triangulation of boundary nodes based on the Delaunay criterion.
	Mesh->Mesh Refinement	Inserts a new point in the center of all triangles that do not fulfill the smoothness criterion.

Table B.9. (continued).

Group	Command	Brief description of the command
	Mesh->Delaunay Retriangulation	Retriangulates mesh according to Delaunay criterion.
	Mesh->Smoothing	Smoothes the mesh by solving a set of coupled elliptic equations in a recursive algorithm.
	Mesh->Convexity Check	Corrects possible errors which may appear while smoothing and retriangulating.
C	View->Geometry	Shows main window for transport domain design.
	View->Boundary Points	Shows main window for boundary discretization.
	View->FEM Mesh	Shows main window for finite element discretization.
	View->Toolbar	Shows or hides the toolbar.
	View->Status Bar	Shows or hides the status bar.
	View All	Shows a default view of the view window.
	Enlarge View	Enlarges the content of the view window by 25%.
	Reduce View	Reduces the content of the view window by 25%.
	Zoom	Zooms in on a certain part of the view window.
	Previous Zoom	Shows the previous zoom on a certain part of the view window.
	Viewport	Specifies dimensions of the view window.
	Stretching	Specifies stretching factors for graphical display of the transport region.
	Redraw	Redraws the content of the view window.
D	Inserts->Single Points	Inserts single points either graphically or numerically.
	Inserts->Line	Inserts line either graphically or numerically.
	Inserts->Spline	Inserts spline either graphically or numerically.
	Inserts->Arc	Inserts arc either graphically or numerically.
	Inserts->Circle	Inserts circle either graphically or numerically.
	Inserts->Object From File	Inserts object defined in a file.
	Inserts->Fixed Point	Inserts fixed point either graphically or numerically.
	Inserts->Periodic Condition	Inserts a periodic condition.
E	Manipulation Mode->Graphical	Specifies whether or not object defining information is entered graphically.
	Manipulation Mode->Numerical	Specifies whether or not object defining information is entered numerically.
	Circle Input Mode->Center and Radius	Inputs circle by defining a center and a radius.
	Circle Input Mode->Three Points	Inputs circle by defining three points.
	Arc Input M.->Center,Radius,Two Angles	Inputs arc by defining a center, a radius, and two angles.
	Arc Input Mode->Three Points	Inputs arc by defining three points.
	Point Picking	Determines whether or not existing nodes influence the moving of a cursor.
	Overlapped Points	Defines editing of multiple nodes.
	Grid Setting	Displays the regular rectangular grid and selects its parameters such as *x* and *y* increments.
	Check Geometry	Controls the consistency of information defining the transport domain geometry.
	Find Node	Finds a node having a specified number.
	Find Edge	Finds an edge having a specified number.
	Find Triangle	Finds a triangle having a specified number.

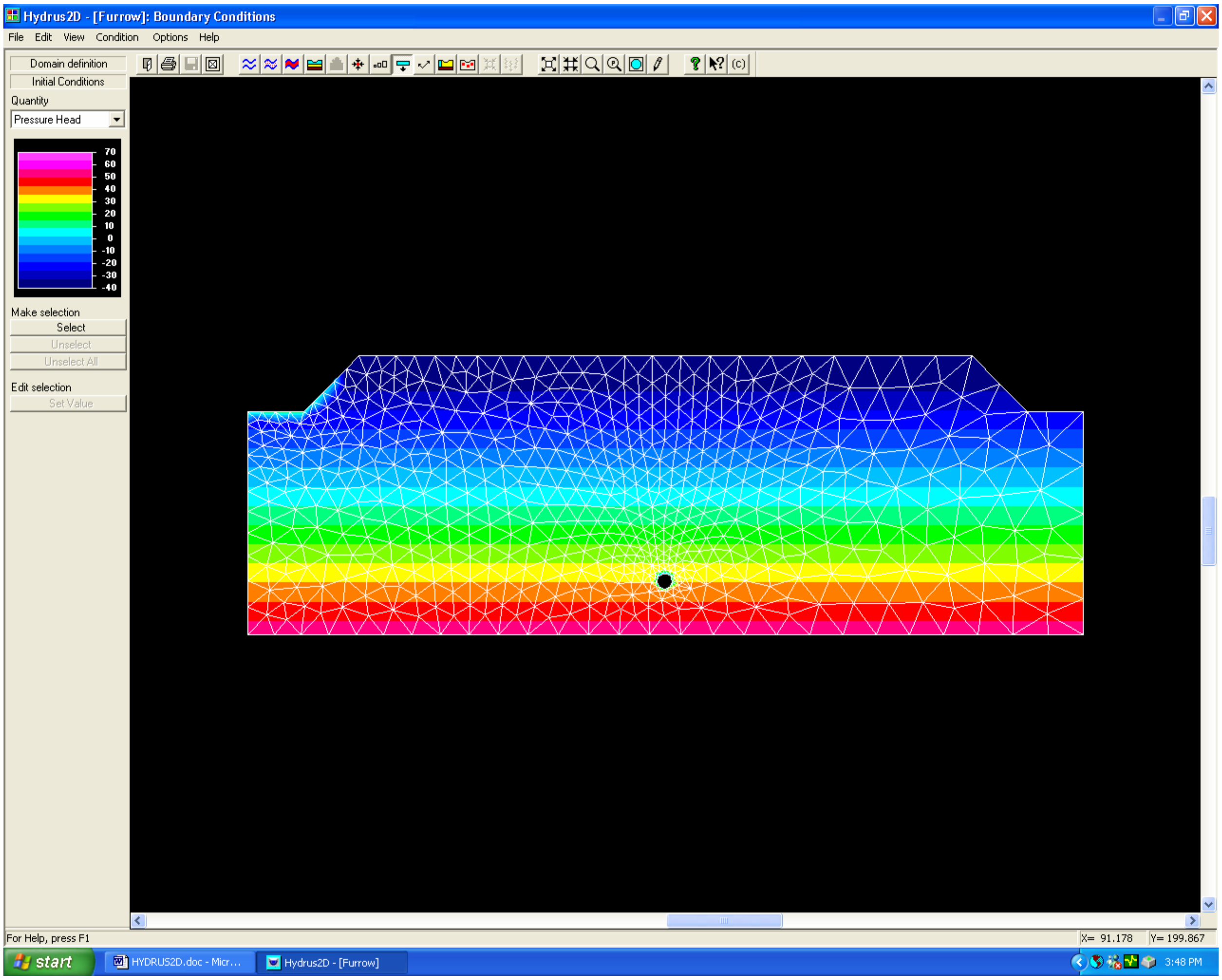
B.2.4. *Module **BOUNDARY** (H2D_BDRC)*

The **BOUNDARY** module (Fig. B.8) helps a user to (1) specify boundary and initial conditions for both water flow and solute transport, and (2) define the spatial distribution of other parameters characterizing the flow domain (e.g., spatial distribution of soil materials, hydraulic scaling factors, root-water uptake parameters, and possible hydraulic anisotropy) and/or observation nodes. All parameters in this module are specified in a graphical environment with the help of a mouse. The program controls the logical correspondence between the water flow and solute transport boundary conditions.

Specification of the boundary conditions is relatively straightforward. The user must first click on the selected boundary condition (e.g., constant head), then click on the Set condition range button, and subsequently move the mouse to the selected position and click the left button. Implementation of the boundary condition terminates with a repeated click of the left button. The boundary nodes will have the color of the corresponding type of boundary condition. See the rules for specifying boundary condition as described in Chapter 8.

Other parameters characterizing the flow domain (initial condition, material distribution) are defined in a similar way. The user must first select that part of the transport domain to which he/she wants to assign a particular value of the selected variable. It is possible to select the entire transport domain, part of it, or only individual nodes or elements. A particular part of the transport domain can be selected as followings: the user must first click the Select button, and then move the mouse to a selected position. The beginning and end of the selection operation is framed by clicking the left button. The selected area is the rectangle defined by the two mouse positions when the left button was clicked. When the selection is completed, the user must click the Set value button and specify the value of the particular variable. The given value will then be assigned to the selected area. When the initial condition is specified, then the user has the possibility of assigning either a constant value to a selected domain, or a unit-gradient distribution for the pressure head and associated water content. All variables are assigned to nodal points, except for those defining anisotropy angles, first and second components of anisotropy, and subregion numbers, which are all assigned to elements.

Table B.10 lists all commands accessible through the menu in the BOUNDARY module, whereas Table B.11 gives a brief discussion of the action taken with the particular commands. More detailed descriptions are available through on-line help.



Quantity: Pressure Head

For Help, press F1

X= 91.178 Y= 199.867

Fig. B.8. The main window of the BOUNDARY module.

Table B.10. Menu commands in the BOUNDARY module.

Group	Menu	Submenu	Sub-Submenu
A	File	Save Data Transform to ASCII Transform to BIN Print Print Preview Print Setup Copy to Clipboard Close	
B	Edit	Select Select by Rhomboid Unselect Unselect All Condition Range Condition Value Condition Code Insert an Obs. Node/Drain Delete an Obs. Node/Drain Delete All Obs. Nodes/Drains	
C	View	Toolbar Status Bar View All Enlarge View Reduce View Zoom Previous Zoom Viewport Stretching Redraw	
D	Condition	Boundary Conditions - Water Flow	No Flux Constant Pressure Head Constant Flux Seepage Face Variable Pressure Head Variable Flux Free Drainage Deep Drainage Atmospheric Boundary
		Boundary Conditions - Solute Transport	First-Type Third-Type Volatile Type
		Boundary Conditions - Heat Transport	First-Type Third-Type
		Material Distribution Root Distribution	

Nodal Recharge

Table B.10. (continued).

Group	Menu	Submenu	Sub-Submenu
		Scaling Factors	Hydraulic Conductivity
			Pressure Head
			Water Content
		Initial Conditions	Pressure Head/Water Content
			Temperature
			Concentration
			Adsorbed/Immobile Concentration
			Import Pressure Head
			Import Temperature
			Import Concentration
			Import Adsorbed/Immobile Concentration
		Local Anisotropy	Angle
			First Component
			Second Component
		Subregions	
		Observation Nodes	
		Drains	
E	Options	Display Options	
		Color Palette	
		Grid Settings	
		Stochastic Distribution of S.F.	
		Subregions=Material Distribution	
		Find	Fine Node
			Fine Edge
			Fine Triangle
F	Help	Index	
		Using Help	
		Help On	
		About BOUNDARY	

Table B.11. Description of all menu commands in the BOUNDARY module.

Group	Command	Brief description of the command
A	Save Data	Saves the active document to its current name and directory.
	Transform to ASCII	Saves the active document to the ASCII file DOMAIN.DAT.
	Transform to BIN	Saves the active document to the binary file DOMAIN.IN.
	Print	Prints a figure displayed in the view window.
	Print Preview	Displays a figure in the view window on the screen as it would appear printed.
	Print Setup	Selects a printer and printer connection.
	Copy to Clipboard	Copies the content of the view window to the Windows metafile in the clipboard.
	Close	Closes the BOUNDARY module.
B	Select	Selects the specified part of a boundary or transport region.
	Select by Rhomboid	Selects the specified part of a transport region using rhomboid.
	Unselect	Unselects the specified part of a boundary or transport region.
	Unselect All	Unselect all selected parts of the boundary or transport region.
	Condition Range	Selects part of a boundary for specifying a boundary condition.
	Condition Value	Specifies the value of a particular variable along a certain part of the boundary.
	Condition Code	Specifies the boundary code of a particular solute transport boundary condition as applied to a certain part of a boundary.
	Insert an Obs. Node/Drain	Inserts an observation node or drain.
	Delete an Obs. Node/Drain	Deletes an observation node or drain.
	Delete All Obs. Nodes/Drains	Deletes all observation nodes or drains.
	Drain Parameters	Specifies values of the drain parameters.
C	Toolbar	Shows or hides the toolbar.
	Status Bar	Shows or hides the status bar.
	View All	Shows a default view of the view window.
	Enlarge View	Enlarges the content of the view window by 25%.
	Reduce View	Reduces the content of the view window by 25%.
	Zoom	Zooms in on a certain part of the view window.
	Previous Zoom	Shows the previous zoom on a certain part of the view window.
	Viewport	Specifies dimensions of the view window.
	Stretching	Specifies stretching factors for graphical display of the transport region.
	Redraw	Redraws the content of the view window.
D	Water Flow BC->No Flux	Specifies a no-flux boundary condition along a selected part of the boundary.
	Water Flow BC->Constant Pressure Head	Specifies a constant pressure head boundary condition along a selected part of the boundary.
	Water Flow BC->Constant Flux	Specifies a constant flux boundary condition along a selected part of the boundary.
	Water Flow BC->Seepage Face	Specifies a seepage face boundary condition along a selected part of the boundary.
	Water Flow BC->Variable Pressure Head	Specifies a variable pressure head boundary condition along a selected part of the boundary.

Table B.11. (continued).

Group	Command	Brief description of the command
	Water Flow BC->Variable Flux	Specifies a variable flux boundary condition along a selected part of the boundary.
	Water Flow BC->Free Drainage	Specifies a free drainage boundary condition along a selected part of the boundary.
	Water Flow BC->Deep Drainage	Specifies a deep drainage boundary condition along a selected part of the boundary.
	Water Flow BC->Atmospheric Boundary	Specifies an atmospheric boundary condition along a selected part of the boundary.
	Solute Transport BC->First-Type	Specifies a first-type boundary condition for solute transport along a selected part of the boundary.
	Solute Transport BC->Third-Type	Specifies a third-type boundary condition for solute transport along a selected part of the boundary.
	Solute Transport BC->Volatile Type	Specifies a volatile-type boundary condition for solute transport along a selected part of the boundary.
	Heat Transport BC->First-Type	Specifies a first-type boundary condition for heat transport along a selected part of the boundary.
	Heat Transport BC->Third-Type	Specifies a third-type boundary condition for heat transport along a selected part of the boundary.
	Material Distribution	Specifies the spatial distribution of soil materials.
	Root Distribution	Specifies the spatial distribution of root water uptake.
	Nodal Recharge	Specifies the spatial distribution of nodal recharge.
	Scaling Fact.->Hydraulic Conductivity	Specifies the spatial distribution of the hydraulic conductivity scaling factors.
	Scaling Fact.->Pressure Head	Specifies the spatial distribution of the pressure head scaling factors.
	Scaling Fact.->Water Content	Specifies the spatial distribution of the water content scaling factors.
	IC->Pressure Head/Water Content	Specifies the initial condition for water flow.
	IC->Temperature	Specifies the initial condition for heat transport.
	IC->Concentration	Specifies the initial condition for solute transport.
	IC->Adsorbed/Immobile Concentration	Specifies the initial condition for nonequilibrium solute transport.
	IC->Import Pressure Head	Imports the initial condition for water flow.
	IC->Import Temperature	Imports the initial condition for heat transport.
	IC->Import Concentration	Imports the initial condition for solute transport.
	IC->Import Adsorbed/Immobile Concentr.	Imports the initial condition for nonequilibrium solute transport.
	Local Anisotropy->Angle	Specifies the spatial distribution of the angle of local anisotropy.
	Local Anisotropy->First Component	Specifies the spatial distribution of the first component of local anisotropy.
	Local Anisotropy->Second Component	Specifies the spatial distribution of the second component of local anisotropy.
	Subregions	Specifies the spatial distribution of subregions for mass balance calculation.
	Observation Nodes	Specifies observation nodes for output of the pressure head, water content, temperature, and concentration at each time step.

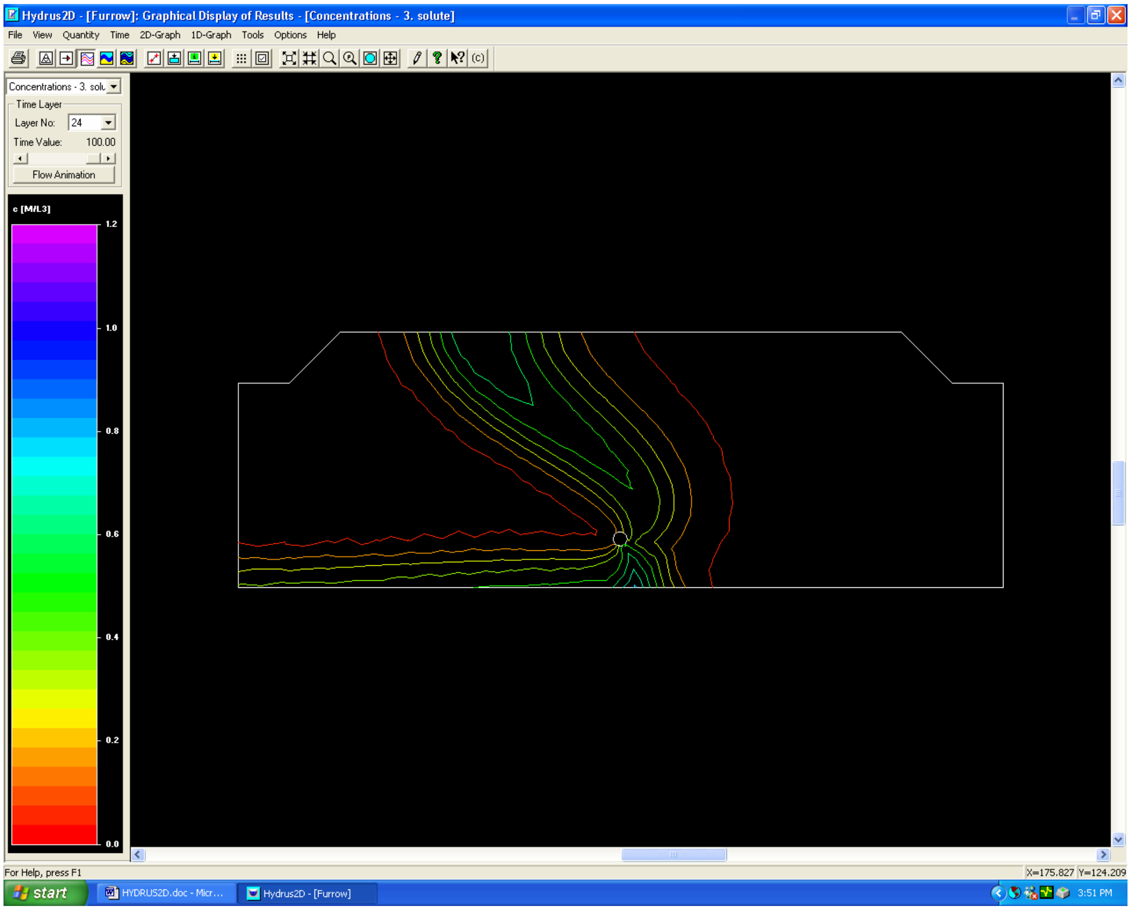
Table B.11. (continued).

Group	Command	Brief description of the command
	Drains	Specifies nodal points representing tile drains.
E	Display Options	Selects options on how to display the transport domain.
	Color Palette	Selects a color palette for display of the initial condition.
	Grid Settings	Specifies grid nodes for better orientation in space, and for easier identification of the *x* and *y* coordinates.
	Stochastic Distribution of S.F.	Generates a stochastic distribution of scaling factors.
	Subregions=Material Distribution	Makes the subregions for mass balance calculations similar to those for the soil materials.
	Fine Node	Finds a particular node.
	Fine Edge	Finds a particular edge.
	Fine Triangle	Finds a particular triangle.
F	Index	Offers an index of topics for which help is available.
	Using Help	Provides general instructions on using help.
	Help On	Provides context sensitive help on parts of the BOUNDARY module.
	About BOUNDARY	Displays the version and authors of the BOUNDARY module.

B.5. *Module* ***GRAPHICS*** *(H2D_GRAF)*

An external **GRAPHICS** module (Fig. B.9) can be used to present results of a simulation by means of contour maps, isolines (Fig. B.10), spectral maps, and velocity vectors (Fig. 12.11), and/or by animation using both contour and spectral maps. The spectral maps can use either the entire color spectrum, or can be made to change monochromatically from white to blue (i.e., using different shades of blue). The number of colors in the color spectrum, as well as the numerical increment between isolines, can be selected by the user. Contour and spectral maps may be drawn for the pressure head, water content, temperature, solute concentration, and/or velocity. Animation of these four variables is also possible. Graphs of all variables at the boundaries, as well as those along any selected cross-section, can be readily obtained.

The entire finite element mesh, boundary nodes, and the numbering of nodes, elements, and/or edges, can be displayed together with isolines and spectral graphs. The user may zoom into a



Layer No: 24

Time Value: 100.00

X=175.827 Y=124.209

Fig. B.9. The main window of the GRAPHICS module.

certain part of the transport domain, and can enlarge or reduce the transport domain, among other features.

Flow animation is an alternative to the display at selected times. During flow animation, distributions at consecutive times are displayed continuously, thereby visualizing the flow and transport process. Note, however, that display times are defined by the print time intervals specified in the input data file. This means that the print times must be at constant intervals so that the time scale of the flow animation will not be distorted. In other words, undistorted flow animation requires that the print time intervals be constant. The speed of the flow animation is dependent on the hardware being used, i.e., the speed of the microprocessor and graphical card.

Table B.12 lists all commands accessible through the menu in the GRAPHICS module, whereas Table B.13 gives a brief discussion of actions taken with particular commands. More detailed descriptions are available through on-line help.

Table B.12. Menu commands in the GRAPHICS module.

Group	Menu	Submenu	Sub-Submenu
A	File	Print Print Options Print Preview Print Setup Copy to Clipboard Close	 View Spectrum
B	View	Toolbar Status Bar View All Enlarge View Reduce View Zoom Previous Zoom Viewport Stretching Redraw	
C	Quantity	Pressure Head Water Content Temperature Concentration Adsorbed/Immobile Concentration Velocity	

Table B.12. (continued).

Group	Menu	Submenu	Sub-Submenu
D	Time	First Time Level	
		Last Time Level	
		Next Time Level	
		Previous Time Level	
		Flow Animation	
E	2D-Graph	Mesh	
		Isolines	
		Velocity Vectors	
		Spectrum	
		Spectrum + Isolines	
F	1D-Graph	Cross-sections	
		Boundary - Whole Curve	
		Boundary - Indexes	
		Boundary - Selection	
G	Tools	Find Node	
		Find Edge	
		Find Triangle	
H	Options	Isolines Parameters	
		Velocity Vectors Parameters	
		Display Options	
		Grid Settings	
		Color Palette	
		Reverse Palette Colors	
		Color Spectrum for All Time Levels	
I	Help	Index	
		Using Help	
		Help On	
		About GRAPHICS	

Table B.13. Description of all menu commands in the GRAPHICS module.

Group	Command	Brief description of the command
A	Print	Sends a graph to the printer.
	Print Options	Specifies the title printed with the figure and whether or not the spectrum scale is to be printed.
	Print Preview	Displays a figure in the view window on the screen as it would appear printed.
	Print Setup	Selects a printer and printer connection.
	Copy to Clipboard: View	Copies the content of the current view to a Windows metafile in the clipboard.
	Copy to Clipboard: Spectrum	Copies the spectrum scale to a Windows metafile in the clipboard.
	Close	Closes the GRAPHICS module.
B	Toolbar	Shows or hides the toolbar.
	Status Bar	Shows or hides the status bar.
	View All	Shows a default view of the view window.
	Enlarge View	Enlarges the content of the view window by 25%.
	Reduce View	Reduces the content of the view window by 25%.
	Zoom	Zooms in on a certain part of the view window.
	Previous Zoom	Shows the previous zoom on a certain part of the view window.
	Viewport	Specifies dimensions of the view window.
	Stretching	Specifies stretching factors for graphical display of the transport region.
	Redraw	Redraws the content of the view window.
C	Quantity->Pressure Head	Displays the spatial distribution of the pressure head by means of isolines or spectrum maps.
	Quantity->Water Content	Displays the spatial distribution of the water content by means of isolines or spectrum maps.
	Quantity->Temperature	Displays the spatial distribution of the temperature by means of isolines or spectrum maps.
	Quantity->Concentration	Displays the spatial distribution of the solution concentration by means of isolines or spectrum maps.
	Quantity->Adsorbed/Immobile Concentr.	Displays the spatial distribution of the adsorbed/immobile concentration by means of isolines or spectrum maps.
	Quantity->Velocity	Displays the spatial distribution of velocities by means of vectors, isolines or spectrum maps.
D	First Time Level	Displays a particular variable at the first time level.
	Last Time Level	Displays a particular variable at the last time level.
	Next Time Level	Displays a particular variable at the next time level.
	Previous Time Level	Displays a particular variable at the previous time level.
	Flow Animation	Displays time levels of a particular variable consecutively and continuously.
E	2D-Graph->Mesh	Displays a finite element mesh.
	2D-Graph->Isolines	Displays the spatial distribution of a particular variable by means of isolines.
	2D-Graph->Velocity Vectors	Displays Darcy velocity vectors.

Table B.13. (continued).

Group	Command	Brief description of the command
	2D-Graph->Spectrum	Displays the spatial distribution of a particular variable by means of a color spectrum.
	2D-Graph->Spectrum + Isolines	Displays the spatial distribution of a particular variable by means of both the color spectrum and isolines.
F	1D-Graph->Cross-sections	Displays values of a particular variable along an arbitrary cross-section.
	1D-Graph->Boundary - Whole Curve	Displays values of a particular variable along a certain boundary curve.
	1D-Graph->Boundary - Indexes	Displays values of a particular variable along a certain boundary between two selected points.
	1D-Graph->Boundary - Selection	Displays values of a particular variable along a certain part of a boundary.
G	Find Node	Finds a specified node.
	Find Edge	Finds a specified edge.
	Find Triangle	Finds a specified triangle.
H	Isolines Parameters	Selects the increments between isolines for display of a particular variable.
	Velocity Vectors Parameters	Selects the multiplication factor for velocity vectors, so as to enlarge or reduce their size.
	Display Options	Selects features to be displayed together with the main graph.
	Grid Settings	Displays a regular rectangular grid and selects its parameters such as *x* and *y* increments.
	Color Palette	Selects a color palette for display of a particular variable.
	Reverse Palette Colors	Reverses colors in a color palette.
	Color Spectrum for All Time Levels	Selects the local or global scale for the spectrum and contour maps.
I	Index	Offers an index of topics for which help is available.
	Using Help	Provides general instructions on using help.
	Help On	Provides context sensitive help on parts of the GRAPHICS module.
	About GRAPHICS	Displays the version and authors of the GRAPHICS module.

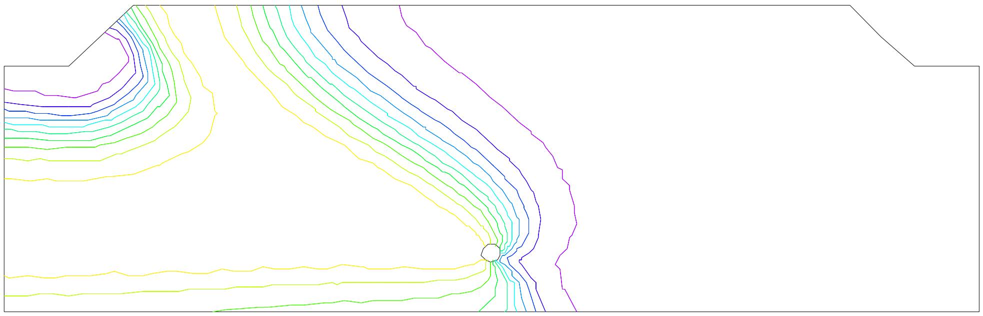


Fig. B.10. Example of an isoline map.

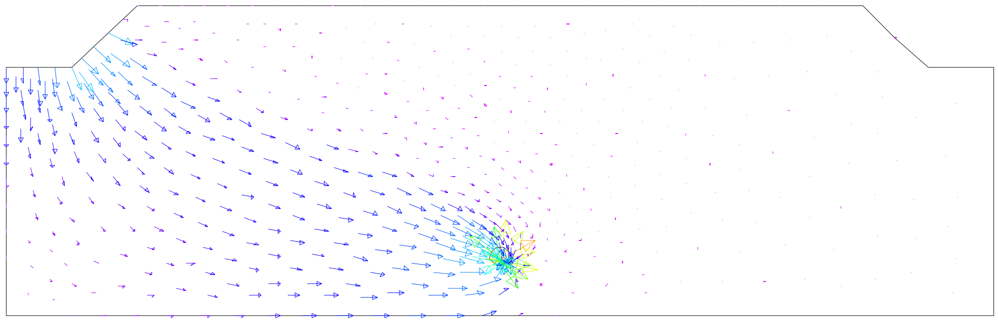


Fig. B.11. Example of a velocity field.