Computers & Geosciences Vol. 19, No. 9, pp. 1239–1266, 1993
Printed in Great Britain.

0098-3004/93 $6.00 + 0.00

RICHARDS: COMPUTER PROGRAM FOR THE NUMERICAL SIMULATION OF ONE-DIMENSIONAL INFILTRATION INTO UNSATURATED SOIL

G. GOTTARDI[1] and M. VENUTELLI[2]

[1]Istituto di Scienze Minerarie and [2]Istituto di Costruzioni Idrauliche, Facoltà di Ingegneria, Università di Bologna, Viale del Risorgimento, 2, I-40136 Bologna, Italy

(Received 31 August 1992; accepted 8 April 1993)

Abstract—Flow into unsaturated soil is governed by the Richards partial differential equation expressing the mass conservation law for the water. Numerical approximations based on different forms of this one-dimensional (1-D) nonlinear partial differential equation can lead to significantly different results for unsaturated flow problems. The Richards equation may be written in three standard forms: the "h-based" form, the "θ-based" form, and the "mixed" form. Documentation is presented for a program which integrates the three standard forms of this equation using finite difference and finite-element techniques. Boundary conditions at prefixed pressure head and prefixed injection rate are allowed; moreover, mass balance errors are computed for controlling the accuracy of the results and for comparing the performances of the three standard forms. The listing of the computer code written in FORTRAN 77 is included along with the user manual and some illustrative infiltration tests in homogeneous and in layered soils.

Key Words: Numerical code, Unsaturated flow, Richards' equations, Finite difference method, Finite-element method, FORTRAN 77.

INTRODUCTION

In many branches of soil science and engineering, such as hydrology, agriculture, and resource management, modeling of water infiltration into unsaturated soil is important. For many practical purposes a 1-D solution of the infiltration problem may be sufficient.

The theory for vertical, transient, isothermal infiltration into unsaturated soil is assumed to obey the classical Richards equation (Bear, 1978; Hillel, 1980). This equation, obtained by applying the mass conservation law and the Darcy flow law, may be written in several forms with either pressure head h [L] or moisture content θ [L^3/L^3] as dependent variable. The three classical standard forms of the Richards equation for the 1-D geometry are:

$$c(h)\frac{\partial h}{\partial t}=\frac{\partial}{\partial z}\left(k(h)\frac{\partial h}{\partial z}-k(h)\right) \quad h\text{-form} \tag{1a}$$

$$\frac{\partial \theta}{\partial t}=\frac{\partial}{\partial z}\left(D(\theta)\frac{\partial \theta}{\partial z}-k(\theta)\right) \quad \theta\text{-form} \tag{1b}$$

$$\frac{\partial \theta}{\partial t}=\frac{\partial}{\partial z}\left(k(h)\frac{\partial h}{\partial z}-k(h)\right) \quad \text{mixed-form} \tag{1c}$$

where $k(h)$ [L/T] is the unsaturated hydraulic conductivity, $c(h)=\mathrm{d}\theta/\mathrm{d}h$ [1/L] is the moisture capacity, $D(\theta)$ [L^2/T] the unsaturated diffusivity and z denotes the vertical coordinate assumed positive downward. Many papers have been written on the numerical solution of these forms of the Richards equation (Haverkamp and others, 1977; Reeder and others, 1980; van Genuchten, 1982; Hills and others, 1989; Celia, Bouloutas, and Zarba, 1990; and Babajimopoulos, 1991).

Valid solutions of these equations are difficult to obtain because of the strong nonlinearity between k, c, D, and h or θ. By using different methods of discretization in the numerical resolution of these equations, a different accuracy can be obtained.

In absence of analytical solutions, the accuracy of the different numerical methods of solution may be evaluated based on the mass balance error. Obviously, this is only a necessary condition which is not sufficient to assure the validity of the solution. The validity of a solution may be assured on the basis of the mass-balance error and of the match of the solutions obtained by different algorithms.

To this purpose, the program presented in this article allows the user to check the validity of a solution by comparing the mass-balance error and the moisture fronts corresponding to different solutions.

NUMERICAL APPROACH

In the following section, the numerical approximations used in the program for integrating the three forms of the Richards equation are given. The numerical discretization of these three partial differential equations by finite difference (FD)

(von Rosenberg, 1969; Lapidus and Pinder, 1982) and finite element (FE) (Connor and Brebbia, 1976) methods leads to a tridiagonal nonlinear set of equations of the form:

$$A_i^m \delta_{i-1}^{m+1} + B_i^m \delta_i^{m+1} + C_i^m \delta_{i+1}^{m+1} = R_i^m \qquad (i = 1, 2, \ldots, N) \quad (2)$$

where the coefficients A_i^m, B_i^m, C_i^m, and R_i^m are nonlinear functions of the variables h or θ and the dependent variables $\delta_j^{m+1} = h_j^{m+1} - h_j^m$ or $\delta_j^{m+1} = \theta_j^{m+1} - \theta_j^m$, $(j = i-1, i, i+1)$ are the increments of the variables h or θ for passing from the interation level m to the iteration level $m+1$. Hereafter the superscripts n, m, and $m+1$ refer to time. The quantities with superscripts m and $m+1$ are evaluated at two subsequent iterations at time $t = (n+1)\Delta t$, whereas quantities with superscript n are calculated at time $t = n\Delta t$, being $\Delta t = t^{n+1} - t^n$. In this program the set of nonlinear algebraic Equations (2) is solved by using a standard Picard iterative technique. The initial values of the dependent variables at time $t = (n+1)\Delta t$ and $m = 1$ are computed by linear extrapolation of the known values of these variables at times $t = (n-1)\Delta t$ and $t = n\Delta t$.

Boundary conditions of Dirichlet type (constant h or θ) and of Neumann type (constant flux) are allowed. The Neumann boundary condition for the first block is given by imposing:

$$-\left(k\frac{\partial h}{\partial z} + k\right) = q \quad \text{or} \quad -\left(D\frac{\partial \theta}{\partial z} + D\right) = q$$

q being the imposed flux. So doing, the coefficients in Equation (2) for the first block are obtained from those of the generic internal ith-block as follows:

$A_1 = 0$

B_1 from B_i by putting: $k_{1-1/2} = 0$ or $D_{1-1/2} = 0$

C_1 from C_i

R_1 from R_i by putting:

$$\frac{k_{1-1/2}}{(\Delta z)^2}(h_0^m - h_1^m) + \frac{k_{1-1/2}}{(\Delta z)} = 0$$

or

$$\frac{D_{1-1/2}}{(\Delta z)^2}(\theta_0^m - \theta_1^m) + \frac{D_{1-1/2}}{(\Delta z)} = 0.$$

As regards the FD methods, different weighting formulae for estimating interblock quantities ($f = k$, $f = c$ or $f = D$) from the available grid point hydraulic conductivities or unsaturated diffusivities are proposed (Haverkamp and Vauclin, 1979):

(1) the arithmetic mean:

$$f_{i\pm 1/2} = 0.5(f_i + f_{i\pm 1})$$

(2) the harmonic mean:

$$f_{i\pm 1/2} = 2\frac{f_i \times f_{i\pm 1}}{f_i \pm f_{i\pm 1}}$$

(3) the geometric mean:

$$f_{i\pm 1/2} = (f_i \times f_{i\pm 1})^{1/2}$$

(4) the upstream method:

$$\begin{cases} f_{i\mp 1/2}^{up} = f_{i\mp 1} & \text{if} \quad h_{i\mp 1} > h_i \quad \text{or} \quad \theta_{i\mp 1} > \theta_i \\ k_{i\mp 1/2}^{up} = k_i & \text{if} \quad h_i > h_{i\mp 1} \quad \text{or} \quad \theta_i > \theta_{i\mp 1}. \end{cases}$$

Five schemes for integrating the three forms of Richards equation are given in the following.

h-based FD approximation

The standard FD approximation of the h-based form (HFD) of the Richards equation using a backward Euler method is:

$$c_i^m \frac{h_i^{m+1} - h_i^n}{\Delta t} = \frac{k_{i-1/2}^m}{(\Delta z)^2}(h_{i-1}^{m+1} - h_i^{m+1}) + \frac{k_{i+1/2}^m}{(\Delta z)^2}(h_{i+1}^{m+1} - h_i^{m+1}) - \frac{k_{i+1/2}^m - k_{i-1/2}^m}{\Delta z} \quad (3)$$

where $k_{i\pm 1/2}^m$ indicate the interblock permeabilities. By indicating with $\delta_i^{m+1} = h_i^{m+1} - h_i^m$ the increment of pressure head at two subsequent iteration levels we obtain:

$$h_i^{m+1} = h_i^m + \delta_i^{m+1} \quad (4)$$

and analogously for the block $i+1, i-1$. Using Equation (4) it is possible to put Equation (3) in the form of Equation (2) where:

$$A_i^m = -\frac{k_{i-1/2}^m}{(\Delta z)^2}; \qquad B_i^m = \frac{c_i^m}{\Delta t} + \frac{k_{i-1/2}^m}{(\Delta z)^2} + \frac{k_{i+1/2}^m}{(\Delta z)^2};$$

$$C_i^m = -\frac{k_{i+1/2}^m}{(\Delta z)^2} \quad (5a, b, c)$$

$$R_i^m = \frac{k_{i-1/2}^m}{(\Delta z)^2}(h_{i-1}^m - h_i^m) + \frac{k_{i+1/2}^m}{(\Delta z)^2}(h_{i+1}^m - h_i^m) - \frac{k_{i+1/2}^m - k_{i-1/2}^m}{\Delta z} - c_i^m \frac{h_i^m - h_i^n}{\Delta t}. \quad (5d)$$

h-based FE approximation

The Galerkin linear FE approximation of the h-based form (HFE) of the Richards equation was obtained by considering that inside each finite element the quantities h, k, and c are expressed in the form:

$$f(x, t) \simeq \hat{f} = \sum_{i=1}^{2} f_i(t)\alpha_i(x) \quad (6)$$

where $f_i(t) = f(x_i, t)$ and $\alpha_i(x)$ is the linear basis function associated with node i. In this way we obtain, in accordance with Celia, Bouloutas, and

Zarba (1990), that the coefficients in Equation (2) are expressed as:

$$A_i^m = \frac{\tilde{c}_{i-1}^m}{\Delta t} - \frac{k_{i-1/2}^m}{(\Delta z)^2}; \qquad B_i^m = \frac{\tilde{c}_i^m}{\Delta t} + \frac{k_{i-1/2}^m}{(\Delta z)^2} + \frac{k_{i+1/2}^m}{(\Delta z)^2};$$

$$C_i^m = \frac{\tilde{c}_{i+1}^m}{\Delta t} - \frac{k_{i+1/2}^m}{(\Delta z)^2} \qquad (7a, b, c)$$

$$R_i^m = \frac{k_{i-1/2}^m}{(\Delta z)^2}(h_{i-1}^m - h_i^m) + \frac{k_{i+1/2}^m}{(\Delta z)^2}(h_{i+1}^m - h_i^m) - \frac{k_{i+1/2}^m - k_{i-1/2}^m}{\Delta z} - \tilde{c}_{i-1}^m \frac{h_{i-1}^m - h_{i-1}^n}{\Delta t} - \tilde{c}_i^m \frac{h_i^m - h_i^n}{\Delta t} - \tilde{c}_{i+1}^m \frac{h_{i+1}^m - h_{i+1}^n}{\Delta t} \qquad (7d)$$

where

$$\tilde{c}_{i-1}^m = \tfrac{1}{12}(c_{i-1}^m + c_i^m);$$
$$\tilde{c}_i^m = \tfrac{1}{12}(c_{i-1}^m + 6c_i^m + c_{i+1}^m);$$
$$\tilde{c}_{i+1}^m = \tfrac{1}{12}(c_i^m + c_{i+1}^m)$$
$$k_{i-1/2}^m = \tfrac{1}{2}(k_{i-1}^m + k_i^m); \quad k_{i+1/2}^m = \tfrac{1}{2}(k_i^m + k_{i+1}^m).$$

Comparison between Equations (7) and (5) indicates that the FE method spatially distributes the time approximation to the three z_{i-1}, z_i, and z_{i+1}, whereas the FD method evaluates the time derivatives at z_i only. If in the FD method the interblock conductivities are taken as the arithmetic mean between k_i and $k_{i\mp1}$, the spatial derivatives in the HFD and HFE methods are identical, moreover, using the mass lumping procedure (van Genuchten, 1980), Equations (7) become identical to Equations (5) in the HFD approximation.

θ-based FD approximation

The standard FD approximation for the θ-based form (TFD) of Richards Equation (1b) obtained by a backward Euler method is:

$$\frac{\theta_i^{m+1} - \theta_i^n}{\Delta t} = \frac{D_{i-1/2}^m}{(\Delta z)^2}(\theta_{i-1}^m - \theta_i^n) + \frac{D_{i+1/2}^m}{(\Delta z)^2}(\theta_{i+1}^{m+1} - \theta_i^{m+1}) - \frac{k_{i+1/2}^m - k_{i-1/2}^m}{\Delta z} \qquad (8)$$

by indicating with $\delta_i^{m+1} = \theta_i^{m+1} - \theta_i^m$ the increment of the moisture content between two subsequent iteration levels we obtain for the coefficients in Equation (2):

$$A_i^m = -\frac{D_{i-1/2}^m}{(\Delta z)^2}; \qquad B_i^m = \frac{1}{\Delta t} + \frac{D_{i-1/2}^m}{(\Delta z)^2} + \frac{D_{i+1/2}^m}{(\Delta z)^2};$$

$$C_i^m = -\frac{D_{i+1/2}^m}{(\Delta z)^2} \qquad (9a, b, c)$$

$$R_i^m = \frac{D_{i-1/2}^m}{(\Delta z)^2}(\theta_{i-1}^m - \theta_i^m) + \frac{D_{i+1/2}^m}{(\Delta z)^2}(\theta_{i+1}^m - \theta_i^m) - \frac{k_{i+1/2}^m - k_{i-1/2}^m}{\Delta z} - \frac{\theta_i^m - \theta_i^n}{\Delta t}. \qquad (9d)$$

θ-based FE approximation

The Galerkin linear FE approximation of the θ-based form (TFE) of Richards Equation (1b) was obtained by considering that inside each finite element the quantities h and D are expressed as a linear combination of the nodal values with the shape functions as in Equation (6). In this situation the coefficients in Equation (2) are:

$$A_i^m = \frac{1}{6\Delta t} - \frac{D_{i-1/2}^m}{(\Delta z)^2};$$

$$B_i^m = \frac{2}{3\Delta t} + \frac{D_{i-1/2}^m}{(\Delta z)^2} + \frac{D_{i+1/2}^m}{(\Delta z)^2};$$

$$C_i^m = \frac{1}{6\Delta t} - \frac{D_{i+1/2}^m}{(\Delta z)^2} \qquad (10a, b, c)$$

$$R_i^m = \frac{D_{i-1/2}^m}{(\Delta z)^2}(\theta_{i-1}^m - \theta_i^m) + \frac{D_{i+1/2}^m}{(\Delta z)^2}(\theta_{i+1}^m - \theta_i^m) - \frac{k_{i+1/2}^m - k_{i-1/2}^m}{\Delta z} - \frac{\theta_{i-1}^m - \theta_{i-1}^n}{6\Delta t} - \frac{2}{3}\frac{\theta_i^m - \theta_i^n}{\Delta t} - \frac{\theta_{i+1}^m - \theta_{i+1}^n}{6\Delta t}. \qquad (10d)$$

Comparison between Equations (10) and (9) indicates that the FE method spatially distributes the time approximation to the three z_{i-1}, z_i, and z_{i+1}, whereas the FD method evaluates the time derivative at z_i only. If in the FD method the interblock conductivities are taken as the arithmetic mean between k_i and $k_{i\mp1}$, the spatial derivatives in the TFD and TFE methods are identical, moreover, by using the mass lumping procedure (van Genuchten, 1980), Equations (10) become identical to Equations (9) in the HFD approximation.

Mixed-form FD approximation

The standard FD approximation for the mixed-form (MFD) of Richards Equation (1c) obtained by a backward method is:

$$\frac{\theta_i^{m+1} - \theta_i^n}{\Delta t} = \frac{k_{i-1/2}^m}{(\Delta z)^2}(h_{i-1}^{m+1} - h_i^{m+1}) + \frac{k_{i+1/2}^m}{(\Delta z)^2}(h_{i+1}^{m+1} - h_i^{m+1}) - \frac{k_{i+1/2}^m - k_{i-1/2}^m}{\Delta z} \qquad (11)$$

by putting with $\delta_i^{m+1} = h_i^{m+1} - h_i^m$ and expressing θ^{m+1} as a truncated Taylor series with respect to h, we obtain:

$$\theta_i^{m+1} = \theta_i^m + \left.\frac{d\theta}{dh}\right|_i^m (h_i^{m+1} - h_i^m) = \theta_i^m + c_i^m \delta_i^{m+1}. \qquad (12)$$

Substituting Equation (12) into Equation (11), the coefficients of the tridiagonal system of equations become:

$$A_i^m = -\frac{k_{i-1/2}^m}{(\Delta z)^2}; \qquad B_i^m = \frac{c_i^m}{\Delta t} + \frac{k_{i-1/2}^m}{(\Delta z)^2} + \frac{k_{i+1/2}^m}{(\Delta z)^2};$$

$$C_i^m = -\frac{k_{i+1/2}^m}{(\Delta z)^2} \qquad (13a, b, c)$$

$$R_i^m = \frac{k_{i-1/2}^m}{(\Delta z)^2}(h_{i-1}^m - h_i^m) + \frac{k_{i+1/2}^m}{(\Delta z)^2}(h_{i+1}^m - h_i^m) - \frac{k_{i+1/2}^m - k_{i-1/2}^m}{\Delta z} - \frac{\theta_i^m - \theta_i^n}{\Delta t}. \qquad (13d)$$

In this situation θ_i^m is computed as a function of h_i^m.

DESCRIPTION OF THE PROGRAM

The computer program is written in FORTRAN 77 and consists of a main section and eleven subroutines. The program can handle layered profiles by assigning to each block a material code. Three types of constitutive relationships $k = k(h)$ and $\theta = \theta(h)$ can be used to characterize the different soils:

Type 1

$$k = k_s \frac{A}{A + |h|^\beta}; \qquad \theta = \frac{\alpha(\theta_s - \theta_r)}{\alpha + |h|^\gamma} + \theta_r \qquad (14a, b)$$

Type 2

$$k = k_s \frac{A}{A + |h|^\beta}; \qquad \theta = \frac{\alpha(\theta_s - \theta_r)}{\alpha + (\ln|h|)^\gamma} + \theta_r \qquad (15a, b)$$

Type 3

$$k = k_s S_e^{1/2}[1 - (1 - S_e^{1/m})]^2; \qquad \theta = S_e(\theta_s - \theta_r) + \theta_r \qquad (16a, b)$$

where

$$S_e = \left[\frac{1}{1 + (\alpha h)^n}\right]^m \qquad m = 1 - n. \qquad (16c, d)$$

In Equations (14)–(16), A, α, β, γ, m, and n are dimensionless parameters, θ_s is the moisture content at saturated conditions, θ_r is the residual moisture content, k_s is the saturated hydraulic conductivity, h the pressure head, and θ the volumetric moisture content. The relationships in Equations (14) and (15) were used by Haverkamp and others (1977) for a sand and for the Yolo light clay, whereas Equations (16) have been used by van Genuchten (1980) to model a Berino loamy fine sand and the Glendale clay loam. The numerical values of the parameters for the four types of the above mentioned soils are given in Table 1.

Constitutive relationships $k = k(h)$ and $\theta = \theta(h)$ for soils different from the previously mentioned ones can be obtained from Equations (14)–(16) by suitably changing the values of the parameters.

For checking the program, an input file of data is given in Table 2, and the corresponding output is given in Table 3. Appendix 1 describes how to use the program, and the listing of the program is given in Appendix 2.

The task performed by each subroutines of the RICHARD program is described in the following.

MAIN: the main program reads input data, drives the other subroutines, computes the adaptive time step and prints out the results.

ASSFD: this subroutine assembles the tridiagonal system of Equations (2) for the HFD approximation (KODE = 2) and for the MFD approximation (KODE = 1) by using Equations (7) and (13) respectively.

ASSFE: this subroutine assembles the tridiagonal system of Equations (2) for the HFE approximation (KODE = 3) using Equations (7).

ASST: this subroutine assembles the tridiagonal system of Equations (2) for the TFD approximation (KODE = 4) and for the TFE approximation (KODE = 5) by using Equations (9) and (10) respectively.

F-KT(θ, i): this subprogram computes the link $k = k(\theta)$ for the ith type of soil.

FUN-K(h, i): this subprogram computes the link $k = k(h)$ for the ith type of soil.

FUN-C(h, i): this subprogram computes the link $c = c(h)$ for the ith type of soil.

FUN-H(θ, i): this subprogram computes the link $h = h(\theta)$ for the ith type of soil.

FUN-T(h, i): this subprogram computes the link $\theta = \theta(h)$ for the ith type of soil. All the given five function subprograms use the formulae in Equations (14)–(16).

SKIP: this subroutine allows to write comment lines in the input file. All the lines with a colon in column one are skipped.

TIME: is a system subroutine to compute execution CPU time.

TRIDAG: this subroutine (Press and others, 1986) solves a tridiagonal algebraic system of equations using the Thomas algorithm.

Table 1. Parameter values for four types of soil used

Soil	Type	k_s [cm/s]	A	α	β	γ/n	θ_s	θ_r
Sand	1	$9.440\ 10^{-3}$	$1.175\ 10^{6}$	$1.611\ 10^{6}$	4.474	3.9600	0.2870	0.0750
Yolo	2	$1.230\ 10^{-5}$	$1.246\ 10^{2}$	$7.390\ 10^{2}$	1.770	4.0000	0.4950	0.1240
Berino	3	$6.261\ 10^{-3}$	-	$2.800\ 10^{-2}$	-	2.2390	0.3658	0.0286
Glendale	3	$1.516\ 10^{-4}$	-	$1.040\ 10^{-2}$	-	1.3954	0.4686	0.1060

Table 2. Input data file for infiltration test in homogeneous soil with MFD method

```
;Title of the problem
;--------------------
  INFILTRATION IN HOMOGENEOUS COLUMN OF SAND
;-----------------------------------------
; KODE: 1 mixed form FD (MFD); 2: h form FD (HFD); 3: h form FE (HFE)
; 4:teta form FD (TFD); 5:teta form FE (TFE)
; KBLOCK: code to choose the way for calculating interblock conductivity:
;         1: arithmetic mean; 2: harmonic mean; 3: geometric mean;
;         4: upstream
; KBOUN: code for boundary conditions 0: constan h o theta; 1: constant rate
;-------------------------------------------------
;NNODE NMAT   NLAY   KODE   KREST KBOUN KBLOCK
;-------------------------------------------------
 60     4      1      1     0      0     1
;-------------------------------------------------------------------------------
;IM   KS(IM)  PA(IM)   ALFA(IM)  BETA(IM) GAMMA(IM)  TETAS(IM)   TETAR(IM)
;-------------------------------------------------------------------------------
;  Sand             (type 1)
  1  9.44E-3  1.175E+6  1.611E+6    4.74    3.96       0.287         0.075
;  Yolo light clay (type 2)
 10  1.23E-5  124.6       739.      1.77    4.0        0.495         0.124
;  Berino           (type 3)
 20  6.26E-3   0.0      2.80E-2     2.239   0.553      0.3658        0.0286
;  Glendale         (type 3)
 21  1.516E-4  0.0      1.04E-2     1.3954  0.283      0.4686        0.1060
;
;----------------- Control data--------------------------------
;DTMIN   DTPRINT  TPRINT  ITERMX  NSTEPS  TINIZ   TFINAL   EPS
;-------------------------------------------------------------
1.D-5      200.     200.    30    100000.   0.    4000.   1.D-8
;--------------------------------------
;ZMIN   DZ   DT    DTMAX DMUL DDIV  NLIM
;--------------------------------------
 0.0    2. 1.D-6  10.D0  1.1   0.5   10
;------------
; Initial data
;-------------
; N        HO
;-------------
 1        -20.
 2        -100.
 60       -100.
;--------------
; Material codes
;--------------
;   N       IMAT(N)
    1         1
   60         1
; Restart data
;-------------------
; VWI    CVWI    CVWO
;-------------------
;  0      0.      0.
;-------
;  RATE
;-------
  2.E-3
```

TEST CASES

Infiltration simulations in homogeneous and in layered soils are presented in the following test cases. The simulations were carried out using the code "RICHARDS" where all five numerical approximations methods were used. The tests allow the comparison of the performance of the different approximation methods on the basis of mass-balance errors and of water content profiles.

Table 3. Output report for infiltration test in homogeneous soil with MFD method

REPORT AT TIME: 1.200E+03 CPU_TIME: 88

****

WATER ENTERED IN STEP:	4.2514E-03	CUM. WATER ENTERED:	6.2952E+00
WATER OUT IN THE STEP:	3.6715E-06	CUM. WATER OUT:	4.4065E-03
INITIAL WATER VOLUME:	9.8650E+00	ACTUAL WATER VOLUME:	1.6156E+01
WATER BALANCE ERROR %:	-1.7777E-04	NUMBER OF TIME STEPS :	345

NODE	IMAT	Z	H	T
1	1	.50	-2.000000E+01	2.698348E-01
2	1	2.50	-2.003055E+01	2.697392E-01
3	1	4.50	-2.007028E+01	2.696144E-01
4	1	6.50	-2.012175E+01	2.694518E-01
5	1	8.50	-2.018831E+01	2.692402E-01
6	1	10.50	-2.027431E+01	2.689643E-01
7	1	12.50	-2.038552E+01	2.686036E-01
8	1	14.50	-2.052968E+01	2.681294E-01
9	1	16.50	-2.071734E+01	2.675007E-01
10	1	18.50	-2.096316E+01	2.666577E-01
11	1	20.50	-2.128810E+01	2.655097E-01
12	1	22.50	-2.172313E+01	2.639125E-01
13	1	24.50	-2.231598E+01	2.616257E-01
14	1	26.50	-2.314439E+01	2.582205E-01
15	1	28.50	-2.434405E+01	2.528709E-01
16	1	30.50	-2.617314E+01	2.438394E-01
17	1	32.50	-2.917080E+01	2.272185E-01
18	1	34.50	-3.450020E+01	1.952312E-01
19	1	36.50	-4.398802E+01	1.457232E-01
20	1	38.50	-5.837406E+01	1.047527E-01
21	1	40.50	-7.581801E+01	8.661608E-02
22	1	42.50	-9.036412E+01	8.096075E-02
23	1	44.50	-9.723886E+01	7.949043E-02
24	1	46.50	-9.931607E+01	7.913691E-02
25	1	48.50	-9.983834E+01	7.905349E-02
26	1	50.50	-9.996247E+01	7.903398E-02
27	1	52.50	-9.999138E+01	7.902945E-02
28	1	54.50	-9.999804E+01	7.902841E-02
29	1	56.50	-9.999956E+01	7.902817E-02
30	1	58.50	-9.999990E+01	7.902811E-02
31	1	60.50	-9.999998E+01	7.902810E-02
32	1	62.50	-1.000000E+02	7.902810E-02
33	1	64.50	-1.000000E+02	7.902810E-02
34	1	66.50	-1.000000E+02	7.902810E-02
35	1	68.50	-1.000000E+02	7.902810E-02
36	1	70.50	-1.000000E+02	7.902810E-02
37	1	72.50	-1.000000E+02	7.902810E-02
38	1	74.50	-1.000000E+02	7.902810E-02
39	1	76.50	-1.000000E+02	7.902810E-02
40	1	78.50	-1.000000E+02	7.902810E-02
41	1	80.50	-1.000000E+02	7.902810E-02
42	1	82.50	-1.000000E+02	7.902810E-02
43	1	84.50	-1.000000E+02	7.902810E-02
44	1	86.50	-1.000000E+02	7.902810E-02
45	1	88.50	-1.000000E+02	7.902810E-02
46	1	90.50	-1.000000E+02	7.902810E-02
47	1	92.50	-1.000000E+02	7.902810E-02
48	1	94.50	-1.000000E+02	7.902810E-02
49	1	96.50	-1.000000E+02	7.902810E-02
50	1	98.50	-1.000000E+02	7.902810E-02
51	1	100.50	-1.000000E+02	7.902810E-02
52	1	102.50	-1.000000E+02	7.902810E-02
53	1	104.50	-1.000000E+02	7.902810E-02
54	1	106.50	-1.000000E+02	7.902810E-02
55	1	108.50	-1.000000E+02	7.902810E-02
56	1	110.50	-1.000000E+02	7.902810E-02
57	1	112.50	-1.000000E+02	7.902810E-02
58	1	114.50	-1.000000E+02	7.902810E-02
59	1	116.50	-1.000000E+02	7.902810E-02
60	1	118.50	-1.000000E+02	7.902810E-02

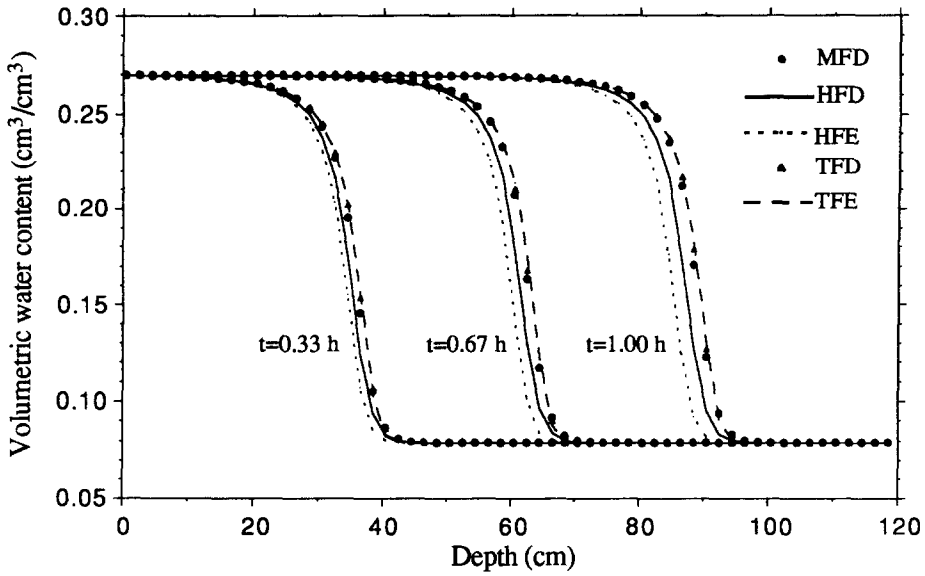

Figure 1. Homogeneous case with constant pressure head at top of column. Moisture profiles at times: 0.33, 0.67, and 1.0 h for five approximation methods.

Infiltration tests in homogeneous soil

This test case consists of fifteen simulations of infiltration into an unsaturated sand column with a depth $L = 120$ cm. The constitutive relationships of the sand used for all the fifteen runs are given in Equations (14) and the values of the parameters are given in Table 1. Ten of the simulations were carried out at Dirichlet conditions (constant pressure head) and five at Neumann conditions (constant rate). For all the runs we used $\Delta z = 2$ cm.

The initial and boundary conditions for the first group of five constant head runs were:

$$h(z, 0) = -100 \text{ cm } (\theta = 0.07903 \text{ cm}^3/\text{cm}^3)$$

$$h(0, t) = -20 \text{ cm } (\theta = 0.269 \text{ cm}^3/\text{cm}^3)$$

$$h(L, t) = -100 \text{ cm}.$$

The water content profiles calculated by the MFD, HFD, HFE, TFD, and TFE methods at three different times are shown in Figure 1, whereas the trend of

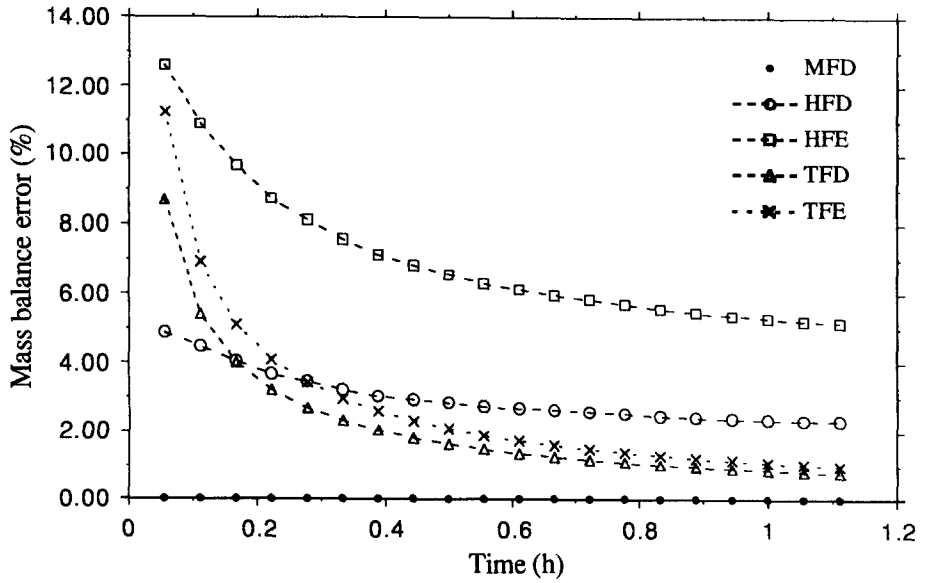

Figure 2. Homogeneous case with constant pressure head at top of column. Mass-balance errors vs time for five approximation methods.

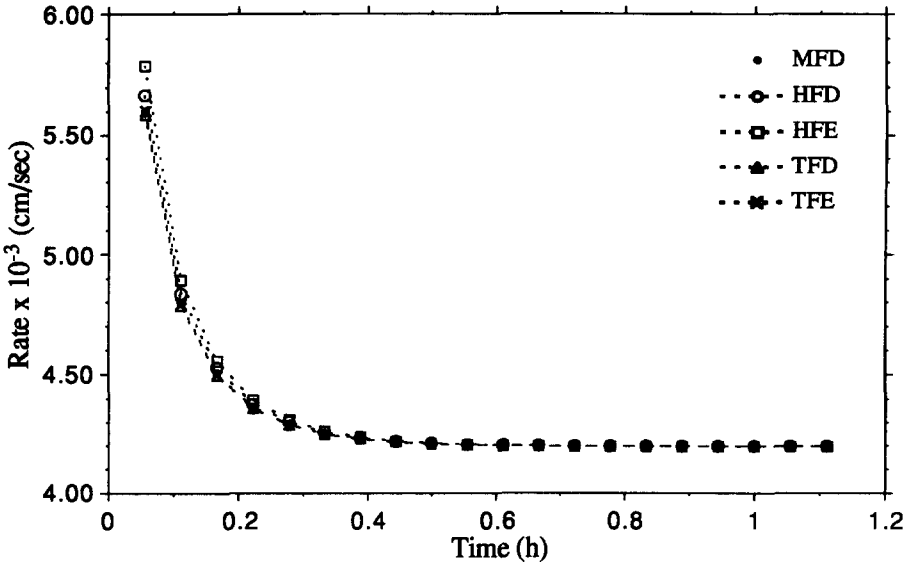

Figure 3. Homogeneous case with constant pressure head at top of column. Water infiltration rate at top of column vs time for five approximation methods.

the mass-balance errors for the same situations is shown in Figure 2. Figure 1 shows good agreement among the different water content profiles and Figure 2 shows how the MFD approximation results in the most mass conservative solution. The plot versus time of the water rate entering the system from the top of the column is shown in Figure 3.

The initial and boundary conditions for the second group of five constant head runs were:

$$h(z, 0) = -61.5 \text{ cm } (\theta = 0.10 \text{ cm}^3/\text{cm}^3)$$

$$h(0, t) = -20.73 \text{ cm } (\theta = 0.267 \text{ cm}^3/\text{cm}^3)$$

$$h(L, t) = -61.5 \text{ cm}.$$

The water content profiles calculated by all the given numerical methods at three different times are compared in Figure 4 with the quasi-analytical solution by Philip (Haverkamp and others, 1977). As shown in Figure 4, for the five runs all the numerical solutions are almost coincident and in good agreement with the quasi-analytical solution. The good match obtained confirms the validity of the numerical methods.

The initial and boundary conditions for the five constant rate tests were:

$$h(z, 0) = -100 \text{ cm } (\theta = 0.07903 \text{ cm}^3/\text{cm}^3) \quad z \neq 0$$

$$h(0, 0) = -50 \text{ cm } (\theta = 0.1241 \text{ cm}^3/\text{cm}^3)$$

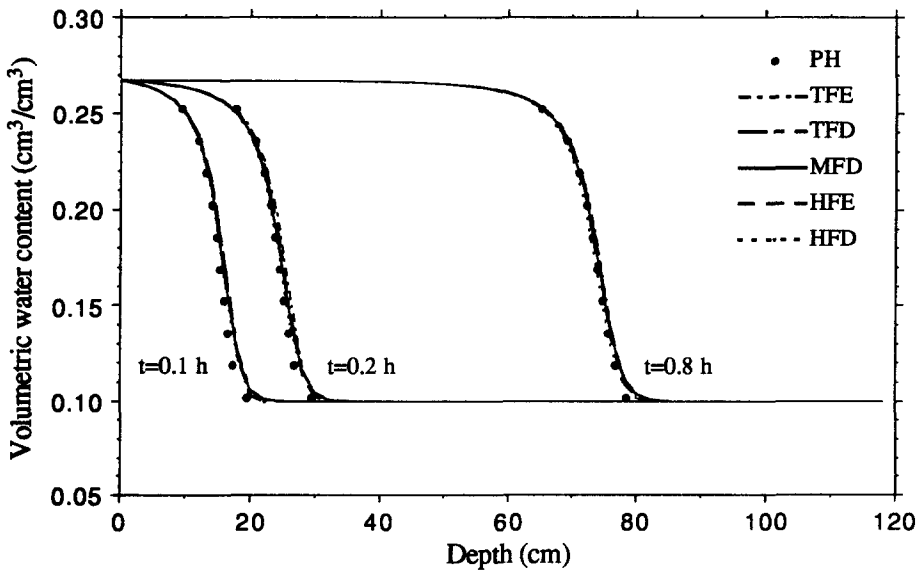

Figure 4. Homogeneous case with constant pressure head at top of column. Moisture profiles at times: 0.1, 0.2, and 0.8 h for five approximation methods and for quasi-analytical solution by Philip (PH).

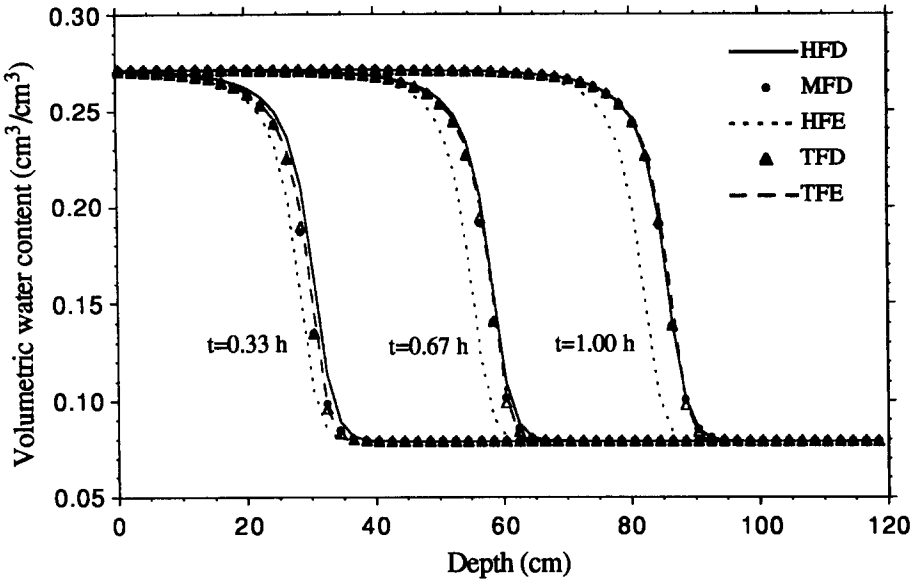

Figure 5. Homogeneous case with constant rate at top of column. Moisture profiles at times: 0.33, 0.67, and 1.0 h for five approximation methods.

$h(L, t) = -100$ cm

$q(0, t) = 0.0045$ cm/sec.

The water content profiles at three different times obtained in this case by the previously five mentioned methods are shown in Figure 5; the graphs of the mass-balance errors versus time are shown in Figure 6. With the exception of the HFE method, the other four methods show a good agreement in the water profiles, whereas, for what concerns the mass balance errors, all five methods present comparable results.

Infiltration tests in layered soil

This test case consists of four simulations of infiltration into an unsaturated layered column of $L = 75$ cm in depth, two of which were carried out at constant pressure head and two at constant rate conditions. For all the runs we used $\Delta z = 1$ cm and five layers, each 15 cm thick. Starting from the top of the column the different types of soil used for each layer were: Sand, Glendale clay loam, Berino loamy fine sand, Yolo light clay and Sand. The constitutive relationships for Sand, Yolo light clay, Berino loamy fine sand, and Glendale clay loam are given in

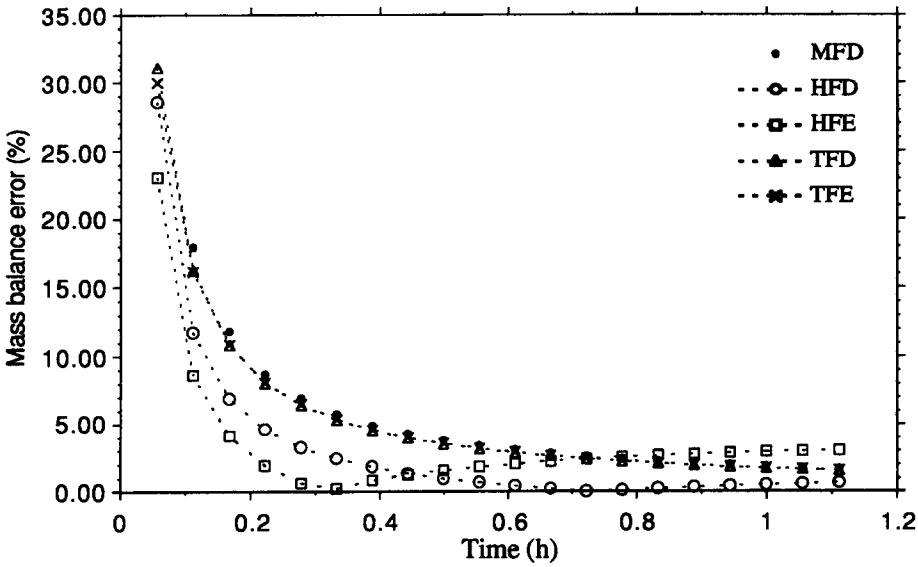

Figure 6. Homogeneous case with constant rate at top of column. Mass-balance errors vs time for five approximation methods.

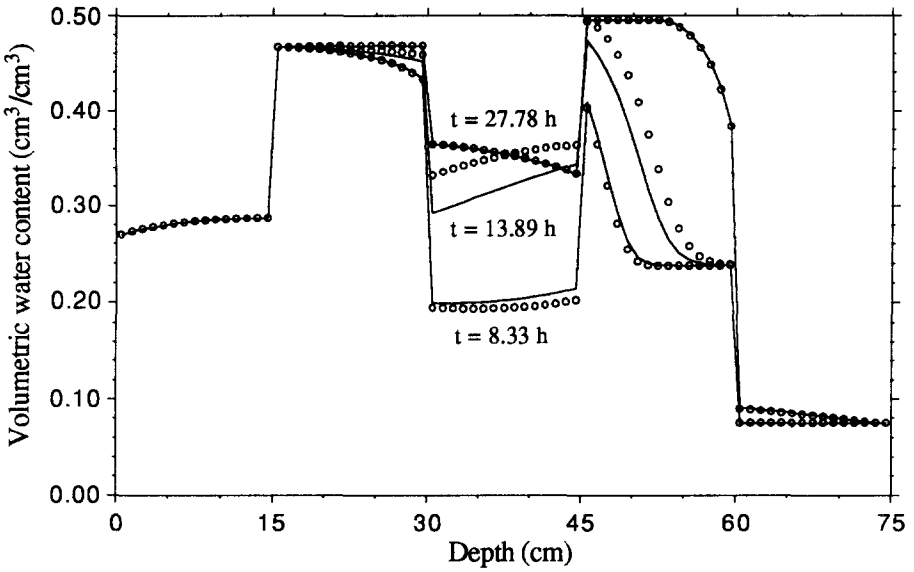

Figure 7. Layered case with constant pressure head at top of column. Moisture profiles at times: 8.33, 13.89, and 27.78 h for HFD (——) and HFE (○) approximation methods.

Equations (14), (15), and (16), respectively, and the values of the parameters are given in Table 1. These typical high- and low-permeability soils have been used widely in the soil science literature (Haverkamp and others, 1977; Hills and others, 1989).

In the situation of layered soils, while the pressure must be continuous across the interference between two soils, the water content need not to be. This discontinuity renders the θ-based and the mixed-based forms of Richards' equations not applicable for modeling water flow in layered soils.

The initial and boundary conditions for the tests at constant pressure head were:

$$h(z, 0) = -600 \text{ cm}$$

$$h(0, t) = -20 \text{ cm}$$

$$h(L, t) = -600 \text{ cm}.$$

The water content profiles obtained by the HFD and the HFE methods, and shown in Figure 7, present a fairly good match.

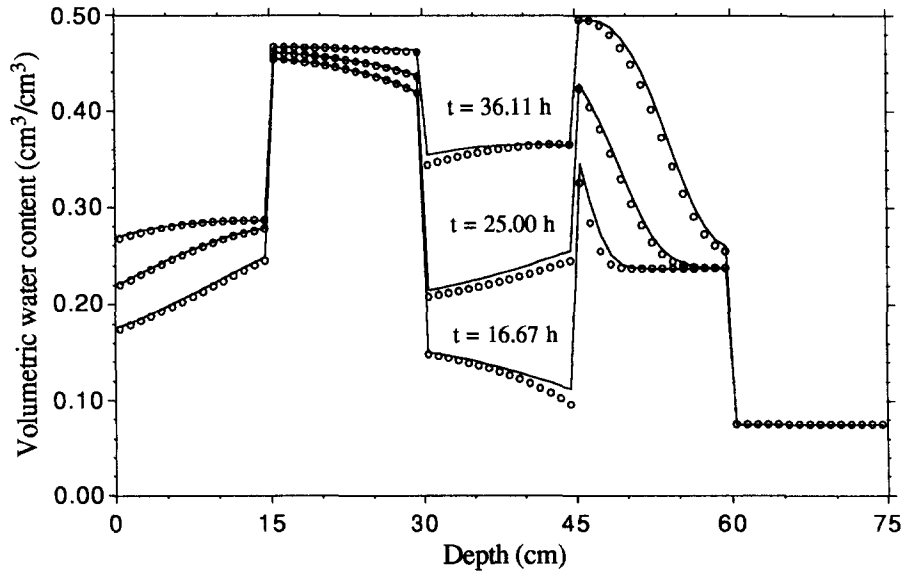

Figure 8. Layered case with constant pressure rate at top of column. Moisture profiles at times: 16.67, 25.00, and 36.11 h for HFD (——) and HFE (○) approximation methods.

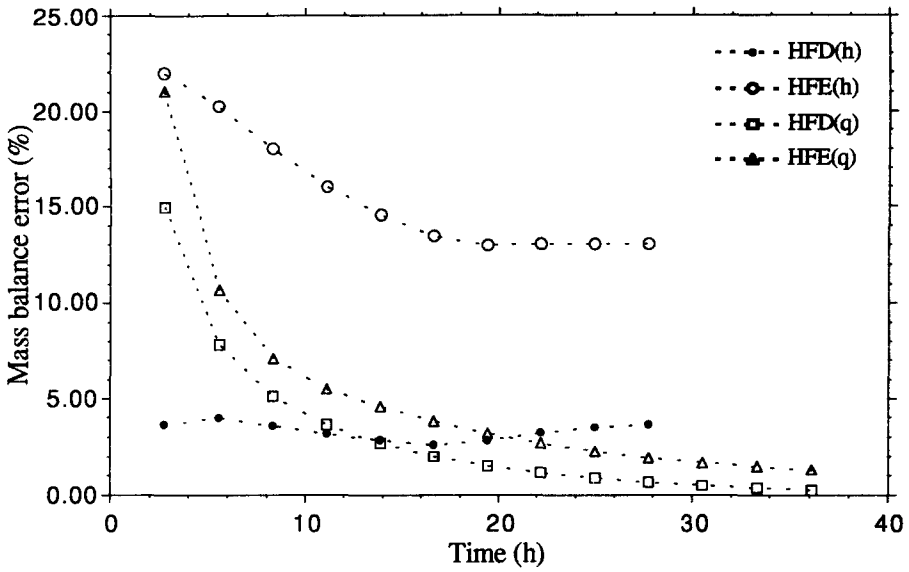

Figure 9. Mass balance errors for layered test case vs time for HFD and HFE approximation methods.

The initial and boundary conditions for the constant rate tests were:

$$h(z, 0) = -600 \text{ cm} \quad z \neq 0$$

$$h(0, 0) = -20 \text{ cm}$$

$$h(L, t) = -600 \text{ cm}$$

$$q(0, t) = 0.0001 \text{ cm/sec.}$$

The water content profiles at three different times obtained in this case are shown in Figure 8, whereas the plots versus time of the mass-balance errors for the constant pressure head cases and the constant rate cases are shown in Figure 9. Also for these tests a good match between the HFD and HFE methods is evident. As far as it concerns the mass-balance errors, the HFD method results more mass conservative than the HFE method.

CONCLUDING REMARKS

Five numerical methods have been tested for the solution of the three different forms of Richards' equation, governing the flow into unsaturated soils. The code presented allows the user to solve infiltration problems into unsaturated soils by using a great variety of options concerning the method of approximation, the type of boundary condition, and the method for calculating the interblock quantities (k, D, and c).

On the basis of the simulations carried out in homogeneous and in layered soils, the use of the five methods implemented in the presented code resulted in the following main points:

—In homogeneous soil the MFD method gave the most conservative solutions. In this situation also the θ-based approximations (TFD and TFE) present acceptable mass-balance errors.

—In layered soils, where only the h-based form of the Richards equation is applicable, the HFD approximation, both for the tests at constant pressure head and the tests at constant rate, resulted in greater mass conservation than the HFE approximation.

REFERENCES

Babajimopoulos, C., 1991, A Douglas–Jones predictor–corrector program for simulating one-dimensional unsaturated flow in soil: Ground Water, v. 29, no. 2, p. 267–270.

Bear, J., 1978, Hydraulics of groundwater: McGraw-Hill Intern. Book Co., New York and London, 569 p.

Celia, M. A., Bouloutas, E. T., and Zarba, R. L., 1990, A general mass-conservative numerical solution for the unsaturated flow equation: Water Resources Res., v. 26, no. 7, p. 1483–1496.

Connor, J. J., and Brebbia, C. A., 1976, Finite element techniques for fluid flow: Newnes–Butterworths, London, 310 p.

van Genuchten, M. Th., 1980, A closed-form equation for predicting the hydraulic conductivity of unsaturated soils: Soil Sci. Soc. America Jour., v. 44, no. 5, p. 892–898.

van Genuchten, M. Th., 1982, A comparison of numerical solutions of the one-dimensional unsaturated–saturated flow and mass transport equations: Adv. Water Resources, v. 5, no. 1, p. 47–55.

Haverkamp, R., Vauclin, M., Touma, J., Wierenga, P. J., and Vachaud, G., 1977, A comparison of numerical simulation models for one-dimensional infiltration: Soil Sci. Soc. America Jour., v. 41, no. 2, p. 285–294.

Haverkamp, R., and Vauclin, M., 1979, A note on estimating finite difference interblock hydraulic conductivity values for transient unsaturated flow problems: Water Resources Res., v. 15, no. 1, p. 181–187.

Hillel, D., 1980, Fundamentals of soil physics: Academic Press, New York, 413 p.

Hills, R. G., Porro, I., Hudson, D. B., and Wierenga, P. J., 1989, Modeling one-dimensional infiltration into very dry soils, 1. Model development and evaluation: Water Resources Res., v. 25, no. 6, p. 1259–1269.

Lapidus, L., and Pinder, G. F., 1982, Numerical solution of partial differential equations in science and engineering: John Wiley International, New York, 677 p.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., 1986, Numerical recipes: Cambridge Univ. Press, Cambridge, 818 p.

Reeder, J. W., Freyberg, D. L., Franzini, J. B., and Remson, I., 1980, Infiltration under rapidly varying surface water depths: Water Resources Res., v. 16, no. 1, p. 97–104.

von Rosenberg, D. U., 1969, Methods for the numerical solution of partial differential equations: Elsevier, New York, 128 p.

APPENDIX 1

Richards Program User Manual

USER MANUAL FOR PROGRAM RICHARDS

Note:

- All line data with a period (;) in the first column are considered comment lines by the program. There is no limit to the number of the comment lines because these lines are skiped by the program during reading input file.

- All input data are read in free format, this imply that also the zero must be given.

c.n	unit of measure	symbol	description
1	-	TITLE	Title of the problem.
2	-	NNODE	Total number of blocks/nodes.
	-	NMAT	Number of available materials.
	-	NLAY	Number of layers.
	-	KODE	Computing method code: KODE=1 for the MFD method, KODE=2 for the HFD method, KODE=3 for the HFE method, KODE=4 for the TFD method, KODE=5 for the TFE method.
	-	KREST	Restart code: KREST=0 no restart, KREST=1 for restart.
	-	KBOUN	Boundary condition code: KBOUN=0 for prefixed pressure head at the first and last node. KBOUN=1 for prefixed rate at the first node and prefixed pressure head at the last node.
	-	KBLOCK	Code for the method of computing interblock quantities (k, c or D): KBLOCK=1 for arithmetic mean, KBLOCK=2 for harmonic mean, KBLOCK=3 for geometric mean, KBLOCK=4 for upstream method.

SOIL PARAMETERS

3	-	IM	Material index.
	cm/sec	KS	Saturated hydraulic conductivity.
	-	PA	Parameter (A) appearing in Eqs. 14a and 15a for the k=k(h)relationship.
	-	ALFA	Parameter (α) appearing in Eqs.14b and 15b 16c for the θ=θ(h) relationship.
	-	BETA	Parameter (β) appearing in Eqs.14a and 15a for k=k(h) relationship.
	-	GAMMA	Parameter (γ) or (n) appearing in Eqs.14b, 15b or 16d for the θ=θ(h) relationship.
	-	TETAS	Saturated moisture content appearing in Eqs.14b,15b and 16b for the θ=θ(h) relationship.
	-	TETAR	Residual moisture content appearing in Eqs.14b,15b and 16b for the θ=θ(h) relationship.

Note:Read NMAT lines of this type.

		TIME CONTROL DATA	
4	sec	DTMIN	Minimum time increment.
	sec	DTPRIN	Print time increment.
	sec	TPRINT	Time for the print of the first report, subsequent reorts will be printed at time: TPRINT+DTPRINT.
	-	ITERMX	Maximum number of iteration for a time step. If ITERMX is exceded the time step becomes: DT=DT*DDIV.
	-	NSTEPS	Maximum number of time steps for the run.
	sec	TINIZ	Initial simulation time.
	sec	TFINAL	Final simulation time.
	-	EPS	Maximum absolute ratio of the discart allowed for two subsequent iterations: $\lvert h^{m+1}-h^{m}\rvert/h^{m}$.
5	cm	ZMIN	Coordinate of the firs node or block.
	cm	DZ	Distance between to subsequent blocks or nodes.
	sec	DT	Initial time step.
	sec	DTMAX	Mamimum value allowed for the time step.
	-	DMUL	Multiplying factor for increasing the time step when the previous step has converged with less than NLIM iterations.
	-	DDIV	Multiplyin factor for reducing the time step when the previous time step has made use of more than NLIM iterations for reaching the convergence.
	-	NLIM	limit number of iterations in a time step.
Read these data only if KODER=0			INITIAL DATA
6	-	NP	Index of node or block.
	-	HO	Initial value array for the pressure head when the MFD, HFD and HFE methods are used (KODE=1,2,3); initial value array for the moisture contents for the TFD and TFE methods (KODE=4,5).

Note: In general one line must be provided for each node for a total of NNODE lines. If the data for a node is omitted the program attributes to this node the value of the previous node. In this way, if the initial values are all equal it is sufficient to read the values for the first and for the last node.

Read these cards only if KODER=0			
7	-	NP	Index of node or block.
	-	IMAT	Material index vector for node NP. For IMAT=1 to 10 the program uses the costitutive relationships k=k(h) and θ=θ(h) of type 1 (formulas 14a and 14b). For IMAT=11 to 20 the program uses the costitutive relationships k=k(h) and θ=θ(h) of type 2 (formulas 15a and 15b). For IMAT=21 to 30 the program uses the costitutive relationships k=k(h) and θ=θ(h) of type 3 (formulas 14a and 14b). In this way it is possible to have 30 different materials, ten for each of the three types of constitutive relationships.

Note: In general one line must be provided for each node for a total of NNODE lines. If the data for a node is omitted the program attributes to this node the value of the previous node. In this way, if the initial values are all equal it is sufficient to read the values for the first and for the last node.

```
-----------------------------------------------------------------
Read this cards only if KODER=1  INITIAL DATA FOR RESTART
-----------------------------------------------------------------
8     -      I        Node or block number.
      -      IMAT     Material index vector for node I.
     cm      Z        Coordinate of node I.
     cm      HO       Initial pressure head value for node I.
     cm      TET      Initial moisture content value for node I.

Note: In general one line must be provided for each node for a total of
NNODE lines. If the data for a node is omitted the program attributes to
this node the value of the previous node. In this way, if the initial values
are all equal it is sufficient to read the values for the first and for the
last node.
-----------------------------------------------------------------
Read this card only if KODER=1  INITIAL DATA FOR MATERIAL BALANCE
-----------------------------------------------------------------
9
    cm3      VWI      Initial volume of water present in the
                      system.
    cm3      CWIN     Cumulative water that entered the system from
                      time zero.
    cm3      CWOU     Cumulative water that leaved the system from
                      time zero.
Note: this data are available from the output of the previous run.
-----------------------------------------------------------------
Read this card only if KBOUN=1
-----------------------------------------------------------------
10   cm/sec RATE      Prefixed rate at the first node.
-----------------------------------------------------------------
```

APPENDIX 2

Richards Program Listing

```
C     ****************
C     PROGRAM RICHARDS
C     ****************
C        by G.Gottardi and M.Venutelli
C        University of Bologna Italy.
C        July 1992
C
C     Program to integrate the three forms of Richards
C     infiltration equation.
C
C     KODE=1 (MFD)  (mixed_form finite_difference method)
C     KODE=2 (HFD)  (head_form finite_differences method)
C     KODE=3 (HFE)  (head_form finite element method)
C     KODE=4 (TFD)  (teta_form finite difference method)
C     KODE=5 (TFE)  (teta_form finite element method)
C-------------------------------------------------------------
C        SUBROUTINES: ASSFD, ASSFE, ASST,SKIP, TRIDAG
C        FUNCTIONS:   F_KT, FUN_K, FUN_C, FUN_H, FUN_T
C-------------------------------------------------------------
      CHARACTER*1 ANS,COM

      PARAMETER(COM=';')
      PARAMETER(NMAX=500)
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 KM,KS
C
      INTEGER T1,T2
      CHARACTER*20 F15,F16,F1
      CHARACTER*80 TITLE
```

```
C
      COMMON/C02/KS(30),PA(30),ALFA(30),BETA(30),GAMMA(30),
     &           TETAS(30),TETAR(30)
      COMMON/C04/KM(NMAX),IMAT(NMAX),DT,DZ,DZ2,HB1,HBN,
     &           KODE,KREST,KBOUN,KBLOCK,RATE
      DIMENSION A(NMAX),B(NMAX),C(NMAX),R(NMAX),Z(NMAX),
     &          HO(NMAX),H(NMAX),U(NMAX)
C
      RATE=0.D0
    1 WRITE(*,1001)
      READ(*,1010) F15
      WRITE(*,1020)
      READ(*,1010) F16
      WRITE(*,1060)
      READ(*,1010) ANS
      IF(ANS.EQ.'E') STOP
      IF(ANS.EQ.'N') GOTO 1
C
C
1001  FORMAT(/,1X,' INPUT FILE NAME    > ',$)
1010  FORMAT(A)
1020  FORMAT(/,1X,' OUTPUT FILE NAME   > ',$)
1060  FORMAT(/,1X,' OK ?  (Y/N/E)      > ',$)
C
      CALL TIME(T1)
C
C--- OPENING OF FILES
C
      OPEN(15,FILE=F15,STATUS='OLD',FORM='FORMATTED')
      OPEN(16,FILE=F16,STATUS='NEW',FORM='FORMATTED')

C
      WRITE(*,*) '**** please wait program RICHARDS is running ***'
C
C     **************
C      INPUT OF DATA
C     **************
C
      WRITE(16,2000)

 2000 FORMAT(T25,'*********************************'/,
     &       T25,'*     UNIVERSITY OF BOLOGNA     *'/,
     &       T25,'*  INSTITUTE OF MINING SCIENCE *'/,
     &       T25,'*********************************'////)
      WRITE(16,2005)
 2005 FORMAT(T25,'PROGRAM RICHARDS',/,
     &       T25,'****************',//)

        CALL SKIP(15,COM,IFINE)
        READ(15,'(A)') TITLE
        WRITE(16,'(A)') TITLE
        WRITE(16,'(//)')
        CALL SKIP(15,COM,IFINE)
        READ(15,*) NNODE,NMAT,NLAY,KODE,KREST,KBOUN,KBLOCK
        WRITE(16,1000) NNODE,NMAT,NLAY,KODE,KREST,KBOUN,KBLOCK
 1000 FORMAT(T5,'NUMBER OF NODES                  (NNODE)...:',I3,/,
     &         T5,'NUMBER OF SOIL MATERIALS        (NMAT)....:',I3,/,
     &         T5,'NUMBER OF LAYERS                (NLAY)....:',I3,/,
     &         T5,'KODE   OF SOLUTION METHOD       (KODE)....:',I3,/,
     &         T5,'KODE   FOR RESTART READING      (KREST)...:',I3,/,
     &         T5,'KODE   FOR BOUNDARY CONDITIONS  (KBOUN)...:',I3,/,
     &         T5,'KODE FOR COMP.INTEBLOCK COND.   (KBLOCK)..:',I3,//)

      CALL SKIP(15,COM,IFINE)

      WRITE(16,1110)
```

```
 1110   FORMAT(T30,'SOIL PARAMETERS',/,
     &          T30,'***************',//,
     &  1X,T5,'IM',T10,'KS(IM)',T21,'PA(IM)',T30,'ALFA(IM)',
     &  T40,'BETA(IM)',T50,'GAMMA(IM)',T60,'TETAS(IM)'
     &  ,T70,'TETAR(IM)'/)
C
C-----READING MATERIAL PARAMETERS
C
      DO 5 I=1,NMAT
        CALL SKIP(15,COM,IFINE)
        READ(15,*)IM, KS(IM),PA(IM),ALFA(IM),BETA(IM),GAMMA(IM),
     &  TETAS(IM),TETAR(IM)
        WRITE(16,1055)IM, KS(IM),PA(IM),ALFA(IM),BETA(IM),GAMMA(IM),
     &  TETAS(IM),TETAR(IM)
    5 CONTINUE
 1055 FORMAT(1X,I3,3(1P,E11.3),4(0PF9.3))

        CALL SKIP(15,COM,IFINE)

        READ(15,*) DTMIN,DTPRINT,TPRINT,ITERMX,NSTEPS,TINIZ,
     &             TFINAL,EPS
        CALL SKIP(15,COM,IFINE)

      READ(15,*)ZMIN,DZ,DT,DTMAX,DMUL,DDIV,NLIM
      WRITE(16,1200)DTMIN,DTPRINT,TPRINT,ITERMX,NSTEPS,TINIZ,
     &             TFINAL,EPS,ZMIN,DZ,DT,DTMAX,DMUL,DDIV,NLIM

 1200 FORMAT(//,T30,'RUN CONTROL DATA',/,
     &         T30,'****************',//,
     &  T5,'DTMIN:',E10.2,T25,'DTPRINT:',E10.3,T45,'TPRINT:',
     &      E10.3,T66,'ITERMX:',I5,
     &  //T5,'NSTEPS:',I9,T25,'TINIZ:',E10.3,T45,'TFINAL:',E10.3,
     &  T66,'EPS:',E10.3,//
     &  T5,'ZMIN: ',F8.2,T25,'DZ: ',E10.2,T45,'DT:',E10.3,T66,
     &  'DTMAX: ',E10.3//
     &  T5,'DMUL:',F5.2,T25,'DDIV:',F5.2,T45,'NLIM:'I5///)
C
C------NODE COORDINATES
C
      IF(KREST.EQ.0) THEN
        Z(1)=ZMIN
        DO 10 I=2,NNODE
        Z(I)=Z(I-1)+DZ
   10 CONTINUE
C
C------INITIAL VALUES
C
        CALL SKIP(15,COM,IFINE)
      NP=0
      DO 25 N=1,NNODE
       IF(NP-N) 15,25,20
   15    READ(15,*) NP,HO(NP)
         IF(N.EQ.NP) GOTO 25
   20   HO(N)=HO(NP)
   25 CONTINUE
C
C-------BLOCK MATERIAL INDICES
C
        CALL SKIP(15,COM,IFINE)
       NP=0
       DO 40 N=1,NNODE
        IF(NP-N) 30,40,35
   30     READ(15,*) NP,IMAT(NP)
          IF(N.EQ.NP) GOTO 40
   35   IMAT(N)=IMAT(NP)
   40 CONTINUE
```

```
C
          VWI=0.D0
          CWIN=0.D0
          CWOU=0.D0
        DO 50 I=1,NNODE
          IF(KODE.LE.3) VWI=VWI+FUN_T(HO(I),IMAT(I))*DZ
          IF(KODE.GE.4) VWI=VWI+HO(I)*DZ
   50   CONTINUE
C
C-------INPUT FOR RESTART
C
      ELSE IF(KREST.EQ.1) THEN

        CALL SKIP(15,COM,IFINE)
          DO 45 I=1,NNODE
            READ(15,*) NP,IMAT(NP), Z(NP),HO(NP),TET
   45     CONTINUE
        CALL SKIP(15,COM,IFINE)
          READ(15,*) VWI,CWIN CWOU
        ENDIF

        CALL SKIP(15,COM,IFINE)
         IF (KBOUN.EQ.1) READ(15,*) RATE
C
C------ PRINT INITIAL DATA
C
        WRITE(16,1061)
 1061   FORMAT(T30,'INITIAL CONDITION TABLE',/,
     &         T30,'************************',//)

        WRITE(16,1070)

       DO 55 I=1,NNODE
        H(I)=HO(I)
        IF(KODE.LE.3) THEN
        WRITE(16,1100) I,IMAT(I),Z(I),HO(I),FUN_T(HO(I),IMAT(I)),
     &                 FUN_K(HO(I),IMAT(I))
        ELSE IF(KODE.GE.4) THEN
        WRITE(16,1100) I,IMAT(I),Z(I),FUN_H(HO(I),IMAT(I)),HO(I),
     &                 F_KT(HO(I),IMAT(I))
        ENDIF

   55 CONTINUE
 1070   FORMAT(1X,T3,'NODE',T10,'IMAT',T26,'Z',T45,'HO',T64,'TETA',
     &         T83,'K'/)
        WRITE(16,1080) VWI
        IF(KBOUN.EQ.1) WRITE(16,1085) RATE
 1080   FORMAT(//,1X,'INITIAL VOLUME OF WATER : ',1P,E12.4)
 1085   FORMAT(   1X,'PREFIXED RATE           : ',1P,E12.4)
 1100 FORMAT(1X,2I5,4(1P,E20.6))
C
        HB1=0.D0
        HBN=0.D0
        NNODE1=NNODE-1
        TIME=TINIZ
        DTO=DT
        DZ2=DZ*DZ
C
C------MAIN LOOP
C
      DO 60 N=1,NSTEPS
C
C------ASSEMBLE MATRIX AND RHS VECTOR
C
      ITER=1
      IFLAG=0
```

```
   65 IF(KODE.LE.2) CALL ASSFD(HO,H,A,B,C,R,NNODE)
        IF(KODE.EQ.3) CALL ASSFE(HO,H,A,B,C,R,NNODE)
        IF(KODE.GE.4) CALL ASST(HO,H,A,B,C,R,NNODE)
C
C-------SOLVE TRIDIAGONAL SET OF EQUATIONS
C
      CALL TRIDAG(A,B,C,R,U,NNODE)
        DHMAX=1.D-30
      DO 70 I=1,NNODE
        UH=U(I)/H(I)
        IF(DABS(UH).GT.DHMAX) DHMAX=DABS(UH)
   70 CONTINUE
C
C-------CONVERGENCE TEST
C
      IF(DHMAX.GT.EPS.AND.ITER.LT.ITERMX) THEN
      DO 75 I=1,NNODE
       H(I)=H(I)+U(I)
   75 CONTINUE
       ITER=ITER+1
       GOTO 65
C
C-------LOWER THE TIME STEP
C
      ELSE IF(DHMAX.GT.EPS.AND.ITER.GE.ITERMX.AND.DT.GE.DTMIN) THEN
      DT=DT*DDIV
      IFLAG=IFLAG+1
      DO 80 I=1,NNODE
         H(I)=H(I)+U(I)
         H(I)=HO(I)+(H(I)-HO(I))*DT/DTO
C          H(I)=HO(I)
   80 CONTINUE
      ITER=1
      GOTO 65
C
C------ABNORMAL EXIT
C
      ELSE IF(DHMAX.GT.EPS.AND.ITER.GE.ITERMX.AND.DT.LE.DTMIN) THEN
      WRITE(9,*) 'CONVERGENCE NOT REACHED'
      PAUSE
      STOP
C
C------CONVERGENCE REACHED
C
      ELSE IF(DHMAX.LE.EPS) THEN
C
      TIME=TIME+DT
       WRITE(9,1320) TIME, ITER,DT,DHMAX
 1320   FORMAT(1X,'TIME: ',1P,E12.3,3X,'ITER: ',I4,3X,'DT: ',
     &  1P,E12.3,3X,'DHMAX: ',1P,E12.3)
      IF(TIME.GE.TPRINT) THEN
      CALL TIME(T2)
      WRITE(16,1065) TIME,T2-T1
 1065   FORMAT(//1X,T15,'REPORT AT TIME: ',1P,E13.3,
     &              T50,'CPU_TIME:',I8,/,
     &              T15,'**************',//)
C
C------UPDATE DATA
C
      DO 85 I=1,NNODE
         H(I)=H(I)+U(I)
   85 CONTINUE
C
      ENDIF
C
C-------MASS BALANCE
```

```
C
        N1=NNODE-1
        IF(KODE.LE.3) THEN
          QWIN=KM(1)*((H(1)-H(2))/DZ+1.D0)
          QWOU=KM(N1)*((H(N1)-H(NNODE))/DZ+1.D0)
        ELSE IF (KODE.GE.4) THEN
         QWIN=KM(1)*((FUN_H(H(1),IMAT(1))-
     &         FUN_H(H(2),IMAT(2)))/DZ+1.D0)
         QWOU=KM(N1)*((FUN_H(H(N1),IMAT(N1))-
     &         FUN_H(H(NNODE),IMAT(NNODE)))/DZ+1.D0)
        ENDIF
C
C------UPDATE CUMULATIVE VOLUMES OF INPUT AND OUTPUT WATER
C
        CWIN=CWIN+QWIN*DT
        CWOU=CWOU+QWOU*DT
C
        IF(TIME.GE.TPRINT) THEN

         VWP=0.D0
        DO 90 I=1,NNODE
          IF(KODE.LE.3) VWP=VWP+FUN_T(H(I),IMAT(I))*DZ
          IF(KODE.GE.4) VWP=VWP+H(I)*DZ
   90   CONTINUE

        RATIO=(VWP-VWI)/(CWIN-CWOU)
        BALER=100.D0*(1.D0-RATIO)
        WRITE(16,1226) QWIN, CWIN,QWOU,CWOU,VWI,VWP, BALER,N
 1226   FORMAT(/1X,'WATER ENTERED IN STEP:',T25,1P,E12.4,T40,
     &          'CUM. WATER ENTERED:',T65,1P,E12.4,
     &      /1X,'WATER OUT IN THE STEP:',T25,1P,E12.4,T40,
     &          'CUM. WATER OUT:',T65,1P,E12.4,
     &      /1X,'INITIAL WATER VOLUME:',T25,1P,E12.4,T40,
     &          'ACTUAL WATER VOLUME: ',T65,1P,E12.4,
     &      /1X,'WATER BALANCE ERROR %:',T25,1P,E12.4,T40,
     &          'NUMBER OF TIME STEPS :',T65,I9,//)
C
C-----PRINT RESULTS
C
      WRITE(16,1325)
 1325   FORMAT(1X,T8,'NODE',T14,'IMAT',T22,'Z',T38,'H',T58,'T',/)

       DO 95 I=1,NNODE
         IF(KODE.LE.3)WRITE(16,1220)I,IMAT(I), Z(I),H(I),
     &                                  FUN_T(H(I),IMAT(I))
         IF(KODE.GE.4)WRITE(16,1220)I,IMAT(I), Z(I),
     &                               FUN_H(H(I),IMAT(I)),H(I)
   95  CONTINUE
 1220 FORMAT( 5X,I5,I5, F8.2,T25,2(1P,E20.6))

        TPRINT=TPRINT+DTPRINT
       ENDIF

       IF(TIME.GE.TFINAL) GOTO 110
       DTO=DT
       IF(ITER.LE.NLIM.AND.IFLAG.EQ.0) DT=DT*DMUL
       IF(DT.GT.DTMAX) DT=DTMAX
       IF(TIME+DT.GT.TPRINT) DT=TIME+DT-TPRINT
C
C-------ESTRAPOLATE
C
       DO 100 I=1,NNODE
              HS=H(I)+(H(I)-HO(I))*DT/DTO
              HO(I)=H(I)
              H(I)=HS
  100  CONTINUE
```

```
      ENDIF

   60 CONTINUE
C
C------NORMAL EXIT
C
  110       CALL TIME(T2)
            WRITE(16,1310) T2-T1
 1250 FORMAT(2X,5E12.4/)
 1260 FORMAT(2X,3(1P,E25.16))
 1300 FORMAT(2X,'STEP ='I5,5X,'X =',E16.7,/4(1P,E25.14))
 1310 FORMAT(5X,'CPU TIME = ',I10)
      CLOSE(15)
      CLOSE(16)
      CLOSE(1)
      STOP
      END
C     *********************************
      SUBROUTINE ASSFD(HO,H,A,B,C,U,N)
C     *********************************
C
C     ASSEMBLES TRIDIAGONAL MATRIX AND RIGHT SIDE VECTOR
C     FOR  THE MFD AND HFD METHODS.
C
        PARAMETER(NMAX=500)
        IMPLICIT REAL*8(A-H,O-Z)
        REAL*8 KS,KM
        COMMON/C02/KS(30),PA(30),ALFA(30),BETA(30),GAMMA(30),
     &  TETAS(30),TETAR(30)
        COMMON/C04/KM(NMAX),IMAT(NMAX),DT,DZ,DZ2,HB1,HBN,
     &             KODE,KREST,KBOUN,KBLOCK,RATE
        DIMENSION H(1),HO(1),A(1),B(1),C(1),U(1)
        DIMENSION FK(NMAX)
C
C

      N1=N-1
      DO 10 I=1,N
        FK(I)=FUN_K(H(I),IMAT(I))
   10   CONTINUE

        DO 15 I=1,N1
        IF(KBLOCK.EQ.1) KM(I)=0.5D0*(FK(I)+FK(I+1))
        IF(KBLOCK.EQ.2) KM(I)=2.D0*(FK(I)*FK(I+1))/(FK(I)+FK(I+1))
        IF(KBLOCK.EQ.3) KM(I)=SQRT(FK(I)*FK(I+1))
        IF(KBLOCK.EQ.4) THEN
           IF(H(I).GE.H(I+1)) KM(I)=FK(I)
           IF(H(I).LT.H(I+1)) KM(I)=FK(I+1)
        ENDIF
   15   CONTINUE

      DO 20 I=2,N1
       CC=FUN_C(H(I),IMAT(I))
       A(I)=-KM(I-1)/DZ2
       B(I)=CC/DT+(KM(I-1)+KM(I))/DZ2
       C(I)=-KM(I)/DZ2
       U(I)=(KM(I-1)*(H(I-1)-H(I))+
     &       KM(I)*(H(I+1)-H(I)))/DZ2-(KM(I)-KM(I-1))/DZ
         IF(KODE.EQ.1) THEN
          FT=FUN_T(H(I),IMAT(I))
          FTO=FUN_T(HO(I),IMAT(I))
            U(I)=U(I)-(FT-FTO)/DT
          ENDIF
       IF(KODE.EQ.2) U(I)=U(I)-CC*(H(I)-HO(I))/DT
   20 CONTINUE
C
C------CONSTANT HEAD BOUNDARY CONDITIONS
```

```
C
      IF(KBOUN.EQ.0) THEN
      U(1)=HB1
      U(2)=U(2)-A(2)*HB1
      U(N1)=U(N1)-C(N1)*HBN
      U(N)=HBN
      A(2)=0.D0
      A(N)=0.D0
      B(1)=1.D0
      B(N)=1.D0
      C(1)=0.D0
      C(N1)=0.D0
      RETURN
C
C------CONSTANT RATE BOUNDARY CONDITIONS
C
      ELSE IF(KBOUN.EQ.1) THEN

         U(N)=HBN
         B(N)=1.D0
         A(N)=0.D0
         U(N1)=U(N1)-C(N1)*HBN
         C(N1)=0.D0
         CC=FUN_C(H(1),IMAT(1))
         B(1)=CC/DT+KM(1)/DZ2
         C(1)=-KM(1)/DZ2
           IF(KODE.EQ.1) THEN
              FT=FUN_T(H(1),IMAT(1))
              FTO=FUN_T(HO(1),IMAT(1))
              U(1)=KM(1)*(H(2)-H(1))/DZ2-KM(1)/DZ-(FT-FTO)/DT+RATE/DZ
           ELSE IF(KODE.EQ.2) THEN
                U(1)=KM(1)*(H(2)-H(1))/DZ2-KM(1)/DZ-
     &          CC*(H(1)-HO(1))/DT+RATE/DZ
           ENDIF
          ENDIF
         RETURN
        END
C     ***********************************
       SUBROUTINE ASST(HO,H,A,B,C,U,N)
C     ***********************************
C
C      ASSEMBLES TRIDIAGONAL MATRIX AND RIGHT SIDE VECTOR
C        FOR THE TFD AND TFE METHODS
C
         PARAMETER(NMAX=500)
         IMPLICIT REAL*8(A-H,O-Z)
         REAL*8 KS,KM
         COMMON/C02/KS(30),PA(30),ALFA(30),BETA(30),GAMMA(30),
     &   TETAS(30),TETAR(30)
         COMMON/C04/KM(NMAX),IMAT(NMAX),DT,DZ,DZ2,HB1,HBN,
     &              KODE,KREST,KBOUN,KBLOCK,RATE
         DIMENSION H(1),HO(1),A(1),B(1),C(1),U(1)
         DIMENSION CC(NMAX),DM(NMAX),FK(NMAX),HT(NMAX)
C
         US=1.D0/6.D0
         DST=2.D0/3.D0
        N1=N-1
        DO 5 I=1,N
         FK(I)=F_KT(H(I),IMAT(I))
         HT(I)=FUN_H(H(I),IMAT(I))
         CC(I)=FUN_C(HT(I),IMAT(I))
    5   CONTINUE

        DO 10 I=1,N1
         P1=FK(I)
         P2=FK(I+1)
         R1=FK(I)/CC(I)
```

```
        R2=FK(I+1)/CC(I+1)
          IF(KBLOCK.EQ.1) KM(I)=0.5D0*(P1+P2)
          IF(KBLOCK.EQ.1) DM(I)=0.5D0*(R1+R2)
          IF(KBLOCK.EQ.2) KM(I)=2.D0*P1*P2/(P1+P2)
          IF(KBLOCK.EQ.2) DM(I)=2.D0*R1*R2/(R1+R2)
          IF(KBLOCK.EQ.3) KM(I)=SQRT(P1*P2)
          IF(KBLOCK.EQ.3) DM(I)=SQRT(R1*R2)
          IF(KBLOCK.EQ.4) THEN
            IF(H(I).GE.H(I+1)) THEN
                KM(I)=P1
                DM(I)=R1
            ELSE
                KM(I)=P2
                DM(I)=R2
            ENDIF
          ENDIF
   10    CONTINUE
        IF(KODE.EQ.4) THEN
        DO 20 I=2,N1
           A(I)=-DM(I-1)/DZ2
           B(I)=1.D0/DT+(DM(I-1)+DM(I))/DZ2
           C(I)=-DM(I)/DZ2
           U(I)=(DM(I-1)*(H(I-1)-H(I))+DM(I)*(H(I+1)-H(I)))/DZ2-
     &           (KM(I)-KM(I-1))/DZ-(H(I)-HO(I))/DT
   20 CONTINUE
        ELSE IF(KODE.EQ.5) THEN

          DO 30 I=2,N1
           A(I)=US/DT-DM(I-1)/DZ2
           B(I)=DST/DT+(DM(I-1)+DM(I))/DZ2
           C(I)=US/DT-DM(I)/DZ2
           U(I)=(DM(I-1)*(H(I-1)-H(I))+DM(I)*(H(I+1)-H(I)))/DZ2-
     &           (KM(I)-KM(I-1))/DZ-US*(H(I-1)-HO(I-1))/DT-
     &            DST*(H(I)-HO(I))/DT-US*(H(I+1)-HO(I+1))/DT
   30 CONTINUE

        ENDIF
C
C------CONSTANT HEAD BOUNDARY CONDITIONS
C
        IF(KBOUN.EQ.0) THEN
        U(1)=HB1
        U(2)=U(2)-A(2)*HB1
        U(N1)=U(N1)-C(N1)*HBN
        U(N)=HBN
        A(2)=0.D0
        A(N)=0.D0
        B(1)=1.D0
        B(N)=1.D0
        C(1)=0.D0
        C(N1)=0.D0
        RETURN
C
C-------CONSTANT RATE BOUNDARY CONDITIONS
C
        ELSE IF(KBOUN.EQ.1) THEN

         U(N)=HBN
         B(N)=1.D0
         A(N)=0.D0
         U(N1)=U(N1)-C(N1)*HBN
         C(N1)=0.D0
         B(1)=1.D0/DT+DM(1)/DZ2
         C(1)=-DM(1)/DZ2
           IF(KODE.EQ.4)U(1)=DM(1)*(H(2)-H(1))/DZ2-KM(1)/DZ-
     &                       (H(1)-HO(1))/DT+RATE/DZ
           IF(KODE.EQ.5)U(1)=DM(1)*(H(2)-H(1))/DZ2-KM(1)/DZ-
```

```
     &       DST*(H(1)-HO(1))/DT-US*(H(2)-HO(2))/DT+RATE/DZ
          RETURN

          ENDIF
        END
C     *********************************
       SUBROUTINE ASSFE(HO,H,A,B,C,U,N)
C     *********************************
C
C       ASSEMBLES TRIDIAGONAL MATRIX AND RIGHT SIDE VECTOR
C      THE HFE METHOD.
C
          PARAMETER(NMAX=500)
          IMPLICIT REAL*8(A-H,O-Z)
          REAL*8 KS,KM,K
          COMMON/C02/KS(30),PA(30),ALFA(30),BETA(30),GAMMA(30),
     &    TETAS(30),TETAR(30)
          COMMON/C04/KM(NMAX),IMAT(NMAX),DT,DZ,DZ2,HB1,HBN,
     &               KODE,KREST,KBOUN,KBLOCK,RATE
          DIMENSION H(1),HO(1),A(1),B(1),C(1),U(1)
          DIMENSION CC(NMAX),CTM(NMAX),CT(NMAX),FNK(NMAX)
C
          UD=1.D0/12.D0
          N1=N-1
          DO 10 I=1,N
           FNK(I)=FUN_K(H(I),IMAT(I))
           CC(I)=FUN_C(H(I),IMAT(I))
   10     CONTINUE

          DO 20 I=1,N1
            KM(I)=0.5D0*(FNK(I)+FNK(I+1))
            CTM(I)=UD*(CC(I)+CC(I+1))
   20     CONTINUE
C
         DO 30 I=2,N1
          CT(I)=UD*(CC(I-1)+6.D0*CC(I)+CC(I+1))
   30     CONTINUE
C
C------- COEFF. A(I), B(I), C(I)
C
          DO 40 I=2,N1
           A(I)=-KM(I-1)/DZ2+CTM(I-1)/DT
           B(I)=(KM(I-1)+KM(I))/DZ2+CT(I)/DT
           C(I)=-KM(I)/DZ2+CTM(I)/DT
           U(I)=(KM(I-1)*(H(I-1)-H(I))+KM(I)*(H(I+1)-H(I)))/DZ2-
      &      (KM(I)-KM(I-1))/DZ-
      &      CTM(I-1)*(H(I-1)-HO(I-1))/DT-CT(I)*(H(I)-HO(I))/DT-
      &      CTM(I)*(H(I+1)-HO(I+1))/DT
    40 CONTINUE
C
C-------CONSTANT HEAD BOUNDARY CONDITIONS
C
          IF(KBOUN.EQ.0) THEN
          U(1)=HB1
          U(2)=U(2)-A(2)*HB1
          U(N1)=U(N1)-C(N1)*HBN
          U(N)=HBN
          A(2)=0.D0
          A(N)=0.D0
          B(1)=1.D0
          B(N)=1.D0
          C(1)=0.D0
          C(N1)=0.D0
          RETURN
C
C-------CONSTANT RATE BOUNDARY CONDITIONS
```

```
C
      ELSE IF(KBOUN.EQ.1) THEN
         U(N)=HBN
         B(N)=1.D0
         A(N)=0.D0
         U(N1)=U(N1)-C(N1)*HBN
         C(N1)=0.D0
           CT(1)=UD*(6.D0*CC(1)+CC(2))
         B(1)=KM(1)/DZ2+CT(1)/DT
           C(1)=-KM(2)/DZ2+CTM(1)/DT
           U(1)=KM(1)*(H(2)-H(1))/DZ2-KM(1)/DZ-
     &     CT(1)*(H(1)-HO(1))/DT-CTM(1)*(H(2)-HO(2))/DT+RATE/DZ
         RETURN
         ENDIF
        END
C       **********************
        FUNCTION F_KT (X,IM)
C       **********************
C       COMPUTES THE LINK k=k(teta)
C       X=MOISTURE CONTENT, IM=MATERIAL INDEX

        IMPLICIT REAL*8(A-H,O-Z)
        REAL*8 KS,N,M
        COMMON/C02/KS(30),PA(30),ALFA(30),BETA(30),GAMMA(30),
     &  TETAS(30),TETAR(30)
C
        IF(X.LE.TETAR(IM)) THEN
        F_KT=0.D0
        RETURN
       ELSE IF(X.GE.TETAS(IM)) THEN
        F_KT=KS(IM)
        RETURN
       ENDIF
C
        IL=IM/10+1
        GOTO (100,200,300) IL
C
C-------SAND
C
  100   S=(X-TETAR(IM))/(TETAS(IM)-TETAR(IM))
        R=BETA(IM)/GAMMA(IM)
        A=ALFA(IM)/S-ALFA(IM)
        D=PA(IM)+A**R
        F_KT=KS(IM)*PA(IM)/D
        RETURN
C
C-------YOLO CLAY
C
  200   S=(X-TETAR(IM))/(TETAS(IM)-TETAR(IM))
        BG=BETA(IM)/GAMMA(IM)
        E=(ALFA(IM)/S-ALFA(IM))**BG
          D=PA(IM)+EXP(E)
          F_KT=TETAS(IM)*PA(IM)/D
      RETURN
C
C------VAN GENUCHTEN
C
  300   N=BETA(IM)
        M=1.D0-1.D0/N
        UM=1.D0/M
        S=(X-TETAR(IM))/(TETAS(IM)-TETAR(IM))
         A=1.D0-(1.D0-S**UM)**M
         AQ=A*A
         F_KT=TETAS(IM)*S**(0.5D0)*AQ
      RETURN
      END
```

```
C     **********************
      FUNCTION FUN_K(X,IM)
C     **********************
C       COMPUTES THE LINK k=k(h)
C       X=PRESSURE HEAD, IM=MATERIAL INDEX
C
        IMPLICIT REAL*8(A-H,O-Z)
        REAL*8 KS,M,N
        COMMON/C02/KS(30),PA(30),ALFA(30),BETA(30),GAMMA(30),
     &  TETAS(30),TETAR(30)
C
        IF(X.GE.0.D0) THEN
        FUN_K=KS(IM)
        RETURN
        ENDIF
C
        IL=IM/10+1
        GOTO (100,100,200) IL
C
C------SAND AND YOLO CLAY
C
  100   FUN_K=KS(IM)*PA(IM)/(PA(IM)+DABS(X)**BETA(IM))
        RETURN
C
C------VAN GENUCHTEN
C
  200   N=BETA(IM)
        M=1.D0-1.D0/N
        B=ALFA(IM)*DABS(X)
        ALFN=B**N
        ALFN1=B**(N-1)
        RN=(1.D0-ALFN1*(1.D0+ALFN)**(-M))**2
        RD=(1.D0+ALFN)**(M/2)
        FUN_K=KS(IM)*RN/RD
        RETURN
      END
C     **********************
      FUNCTION FUN_C(X,IM)
C     **********************
C       COMPUTES THE LINK C=C(h)
C       X= PRESSURE HEAD  IM= MATERIAL INDEX
C
        IMPLICIT REAL*8(A-H,O-Z)
        REAL*8 KS,M,N
        COMMON/C02/KS(30),PA(30),ALFA(30),BETA(30),GAMMA(30),
    &         TETAS(30),TETAR(30)
C
        IF(X.GE.0.D0) THEN
        FUN_C=0.D0
        RETURN
        ELSE IF(X.LT.-1.D35) THEN
        FUN_C=0.D0
        RETURN
        ENDIF
C
        IL=IM/10+1
        GOTO (100,200,300) IL
C
C------SAND
C
 100    A=(ALFA(IM)+DABS(X)**GAMMA(IM))**2
        GAMMA1=GAMMA(IM)-1.D0
        B=(ALFA(IM)*(TETAS(IM)-TETAR(IM))*GAMMA(IM)*DABS(X)**GAMMA1)
       FUN_C=B/A
       RETURN
C
C----YOLO CLAY
```

```
C
  200 IF(X.GE.-1.D0) THEN
       FUN_C=0.D0
       RETURN
        ENDIF

        A=(ALFA(IM)+LOG(DABS(X))**GAMMA(IM))**2
        GAMMA1=GAMMA(IM)-1.D0
        B1=ALFA(IM)*(TETAS(IM)-TETAR(IM))*GAMMA(IM)
        B2=(DLOG(DABS(X)))**GAMMA1/DABS(X)
        B=B1*B2
        FUN_C=B/A
       RETURN
C
C-----VAN GENUCHTEN
C
  300   N=BETA(IM)
        M=1.D0-1.D0/N
        A=ALFA(IM)*DABS(X)
        A1=A**N
        A2=A**(N-1.D0)
        RN=N*M*(TETAS(IM)-TETAR(IM))*A2*ALFA(IM)
        RD=(1.D0+A1)**(M+1.D0)
        FUN_C=RN/RD
        RETURN
        END
C       **********************
        FUNCTION FUN_H (X,IM)
C       **********************
C       COMPUTES THE LINK  h=h(teta)
C       X=MOISTURE CONTENT, IM=MATERIAL INDEX

         IMPLICIT REAL*8(A-H,O-Z)
         REAL*8 KS,N,M
         COMMON/C02/KS(30),PA(30),ALFA(30),BETA(30),GAMMA(30),
     &  TETAS(30),TETAR(30)
C
        IF(X.LE.TETAR(IM)) THEN
        FUN_H=-1.E30
        RETURN
         ELSE IF(X.GE.TETAS(IM)) THEN
         FUN_H=0.D0
         RETURN
       ENDIF
C
        IL=IM/10+1
        GOTO (100,200,300) IL
C
C-------SAND
C
  100   S=(X-TETAR(IM))/(TETAS(IM)-TETAR(IM))
        A=ALFA(IM)/S-ALFA(IM)
        FUN_H=-A**(1.D0/GAMMA(IM))
        RETURN
C
C-------YOLO CLAY
C
  200   S=(X-TETAR(IM))/(TETAS(IM)-TETAR(IM))
        UG=1.D0/GAMMA(IM)
        E=(ALFA(IM)/S-ALFA(IM))**UG
       FUN_H=-EXP(E)
       RETURN
C
C-------VAN GENUCHTEN
C
  300   N=BETA(IM)
        M=1.D0-1.D0/N
```

```
        S=(X-TETAR(IM))/(TETAS(IM)-TETAR(IM))
         A=S**(-1.D0/M)-1.D0
         FUN_H=-A**(1.D0/N)/ALFA(IM)
        RETURN
      END
C     ********************
      FUNCTION FUN_T(X,IM)
C     ********************
C       COMPUTES THE LINK theta=theta(h)
C       X=MOISTURE CONTENT, IM=MATERIAL INDEX

        IMPLICIT REAL*8(A-H,O-Z)
        REAL*8 KS,N,M
        COMMON/C02/KS(30),PA(30),ALFA(30),BETA(30),GAMMA(30),
     &  TETAS(30),TETAR(30)
C
        IF(X.GE.0.D0) THEN
        FUN_T=TETAS(IM)
         ELSE IF(X.LT.-1.D4) THEN
         FUN_T=TETAR(IM)
        RETURN
      ENDIF
C
        IL=IM/10+1
        GOTO (100,200,300) IL
C
C------SAND
C
 100    A=ALFA(IM)*(TETAS(IM)-TETAR(IM))
        B=ALFA(IM)+(DABS(X))**GAMMA(IM)
        FUN_T=A/B+TETAR(IM)
        RETURN
C
C-----YOLO LIGHT CLAY
C
  200 IF(X.GE.-1.D0)THEN
         FUN_T=TETAS(IM)
         RETURN
        ENDIF
       A=ALFA(IM)*(TETAS(IM)-TETAR(IM))
       B=ALFA(IM)+(DLOG(DABS(X)))**GAMMA(IM)
       FUN_T=A/B+TETAR(IM)
       RETURN
C
C-----VAN GENUCHTEN
C
  300   N=BETA(IM)
        M=1.D0-1.D0/N
        ALFN=(ALFA(IM)*DABS(X))**N
        RD=(1.D0+ALFN)**M
        FUN_T=(TETAS(IM)-TETAR(IM))/RD+TETAR(IM)
        RETURN

      END
C     ******************************
      SUBROUTINE TRIDAG(A,B,C,R,U,N)
C     ******************************
      PARAMETER (NMAX=500)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION GAM(NMAX),A(1),B(1),C(1),R(1),U(1)

      IF(B(1).EQ.0.)PAUSE
      BET=B(1)
      U(1)=R(1)/BET

      DO 11 J=2,N
        GAM(J)=C(J-1)/BET
```

```
        BET=B(J)-A(J)*GAM(J)
        IF(BET.EQ.0.)PAUSE
        U(J)=(R(J)-A(J)*U(J-1))/BET
11    CONTINUE

      DO 12 J=N-1,1,-1
        U(J)=U(J)-GAM(J+1)*U(J+1)
12    CONTINUE

      RETURN
      END
C     **********************************
      SUBROUTINE SKIP(IFILE,COM,IFINE)
C     **********************************
      CHARACTER*1 CAR,COM
      IFINE=0
    1 READ(IFILE,'(A1)',END=2) CAR
      IF (CAR.NE.COM) THEN
        BACKSPACE IFILE
        RETURN
      END IF
      GOTO 1
    2 IFINE=1
      RETURN
      END
```