Simulation of one-dimensional water flow, including temperature and hysteresis effects

Agronomy and Soils Departmental Series No. 102
Alabama Agricultural Experiment Station
Auburn University Auburn University, Alabama
Gale A. Buchanan, Director September 1985

Simulation of One-dimensional Water Flow, Including Temperature and Hysteresis Effects

J.W. Hopmans and J.H. Dane

Graduate Research Assistant and Associate Professor

of Agronomy and Soils

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

SUMMARY

The pressure head form of the general water flow equation was numerically solved using the predictor-corrector method (2). The model accounts for temperature effects on the soil hydraulic properties. Hysteresis was considered by implementing Mualems modified dependent domain theory. Scanning curves predicted by the model were compared with those predicted by Mualem (8). The effects of temperature and hysteresis on soil water movement were investigated for various boundary conditions. Results indicated that these effects depend on the type of surface boundary condition applied and that hysteresis tends to dominate temperature effects. A description and listing of the model is included in the Appendix.

INTRODUCTION

A previous Departmental Series (2), reported investigation of the effect of temperature dependent hydraulic properties on soil water movement for a variety of initial and boundary conditions. Although not considered, it was anticipated that hysteresis in the soil water content - pressure head relationship ($\theta(h)$) would influence the water content distributions, especially when wetting and drying cycles occur. As a result of hysteresis, the $\theta(h)$ function determined during drainage is not the same as the one found upon rewetting. Furthermore, the relation between θ and h will depend on the volumetric water content at which the reversal from drainage to wetting (or wetting to drainage) occurs. Therefore, instead of a single valued isothermal $\theta(h)$ function, we actually deal with a multivalued hysteresis functional (4).

A first attempt in modeling soil-water hysteresis, using the independent domain concept, was carried out by Poulovassilis (10). Inherent to the independent domain concept are two assumptions: (1) the pore space is made up of pores or domains, with each pore size defined by two pressure head values, one at which the pore drains and one at which the pore fills, i.e., the draining and filling of each pore takes place independent of the state of the remaining pores in the system, and (2) the water volume difference between the drained and the filled status of each pore is independent of the pressure head.

The independent domain concept was tested by Topp and Miller (15), Talsma (12) and Topp (13). Their conclusion was that, in general, the theory predicted scanning curves moderately well, except possibly at the high water content ranges. This failure was associated with air-entry values of different pore sizes on the main drying curve. Poulovassilis and Childs (11) and Topp (14) resolved this inadequacy by including a domain dependence factor so that the draining and wetting of each pore were assumed to be dependent on the state of neighboring pores.

A computational scheme based on the independent domain model was introduced by Mualem (5,6). Mualem and Dagan (9) and Mualem (7) introduced a domain dependence factor, $P_d(\theta)$, to account for the impedance of air entry by water, which is especially important for soils having high air-entry values. Mualem (8) defined $P_d(\theta)$ as the ratio between the volume of pores actually emptied and the volume which could have been emptied had all pores guaranteed access to air from neighboring pores.

After treating the temperature aspects of soil hydraulic properties and a review of Mualem's hysteresis theory, the hysteresis model of Mualem (8) was incorporated in the soil water flow model introduced by Hopmans and Dane (2). The objective was to investigate the combined effects of temperature and hysteresis on soil water movement, which may be especially important in predicting actual field situations.

THEORETICAL

Temperature Dependent Hydraulic Properties

Assuming one-dimensional flow, the general water flow equation in its pressure head form can be written as (4):

$$C(h,T)\frac{\partial h}{\partial t} = \frac{\partial}{\partial z}\{K(h,T)[\frac{\partial h}{\partial z} + 1]\} \qquad , [1]$$

where C is the specific water capacity (slope of water retention curve), T is temperature, z is distance (0 at reference level and > 0 above it), t is time, h is soil-water pressure head, and K denotes the hydraulic conductivity. Hopmans and Dane (2) derived the following expression to compute the soil-water pressure head (h_T) at any temperature (T), assuming knowledge of a reference soil-water pressure head value (h_{ref}) at a reference temperature (T_{ref}):

$$h_T = \alpha(T)h_{ref} \qquad , [2]$$

where $\alpha(T)$ is defined as

$$\alpha(T) = 1 + (T-T_{ref})\gamma(T) \quad ,$$

and $\gamma(T)$ denotes the temperature coefficient of surface tension of soil water. Furthermore, the water capacity C(h) was found to be dependent on temperature by

$$C(h_T) = (1/\alpha(T))C(h_{ref}) \quad , \quad [3]$$

while the hydraulic conductivity, $K_T(\theta)$, at any temperature, T, was calculated from

$$K_T = (\mu_{ref}/\mu_T)K_{ref} \quad , \quad [4]$$

where μ_{ref} and μ_T denote the viscosity of water (Ns/m^2) at the reference temperature and the soil temperature in question, respectively, and K_{ref} is the hydraulic conductivity value at the reference temperature. The effect of temperature dependent hydraulic properties on soil water movement was investigated by Hopmans and Dane (2), using Eq. [2] through [4]. Hopmans and Dane (3) employed Eq. [2] through [4] to obtain solutions to Eq. [1] using the Douglas-Jones approximation (implicit method with implicit linearization).

Hysteresis

The following analyses pertain to various hysteresis theories, which were initially introduced by Poulovassilis (see references) and further developed by Mualem. A main drying curve refers to the relationship between soil water pressure head and volumetric water content when the soil water pressure head in an initially "saturated" soil is decreased until a limiting low water content is reached. A main wetting curve is defined as the relationship between soil water pressure head and water content when the soil water pressure head is increased until saturation, starting at the limiting water content value. When drying starts at some soil water pressure head along the main wetting curve, one refers to a primary drying curve.

I. Universal independent domain model

Mualem (6) distinguishes between two parameters that characterize the pores or channels in a hypothetical porous medium. These parameters are r, the radii of the openings of the pores, and ρ, the radii of the pores within a group with openings r, figure 1. The soil water potentials at which each

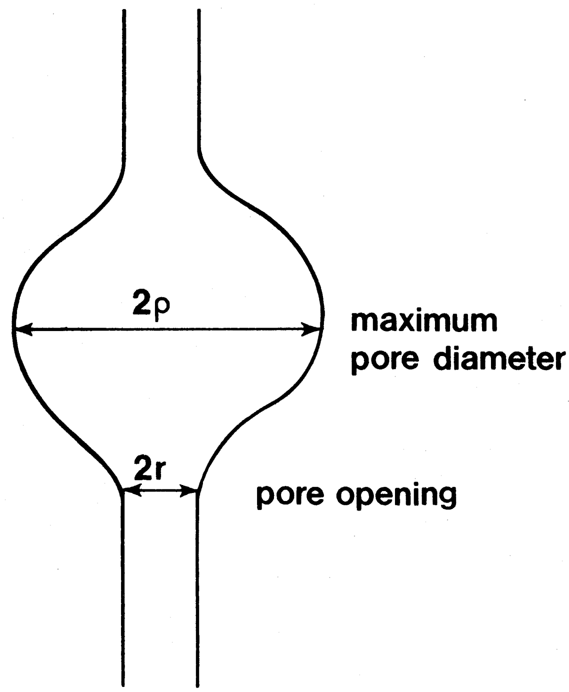


FIG. 1. Schematic diagram of a pore.

pore fills or drains is characterized by the parameters ρ and r, respectively. A bivariate pore water distribution function $f(r,\rho)$ can be defined that describes the relative volume of pores of radii $\rho \rightarrow \rho+d\rho$ having openings of radii $r \rightarrow r+dr$:

$$f(r,\rho)drd\rho = dV_p(r \rightarrow r+dr, \rho \rightarrow \rho+d\rho)/V \quad , \qquad [5]$$

V being the total volume of the sample and dV_p the change in water-filled pore volume. In normalizing r and ρ we obtain:

$$\bar{r} = \frac{r-R_{min}}{R_{max}-R_{min}} \qquad \text{and} \qquad \bar{\rho} = \frac{\rho-R_{min}}{R_{max}-R_{min}} \quad , \qquad [6]$$

where R denotes specific values for ρ or r. By the capillary law, all size measures, r, ρ, and R, are proportional to $1/h$, where h is the corresponding soil water pressure head. R_{max} and R_{min} correspond to h_{max} (at maximum water content, θ_u) and h_{min} (at residual water content, θ_{min}), respectively, figure 3. The radii $\bar{r}$ and $\bar{\rho}$ change in the range from zero to one, assuming that both r and ρ vary between R_{min} and R_{max}. In addition, we define $\Theta=\theta-\theta_{min}$, where θ and θ_{min} are the actual and residual water content. After any number of wetting and drying cycles, $\Theta(\bar{R})$ can then be obtained from integration of $f(\bar{r},\bar{\rho})$ over the domain of water-filled pores. $\bar{R}$ is defined by Eq. [6], where ρ or r is replaced by R, and $\Theta(\bar{R})$ conveniently replaces $\Theta(\bar{r},\bar{\rho})$.

However, so far no indication has been given as to how $f(\bar{r},\bar{\rho})$ can be determined. Assuming that the probability density functions of $\bar{r}$ and $\bar{\rho}$ are independent, the bivariate distribution function can be written as the product of the two marginal distribution functions, i.e.,

$$f(\bar{r},\bar{\rho}) = b(\bar{r})l(\bar{\rho}) \quad . \quad [7]$$

By definition, $f(\bar{r},\bar{\rho})$, $b(\bar{r})$, and $l(\bar{\rho})$ are stricktly positive. Equation [7] states that the pores of any group $\bar{r}$ have the same distribution function $l(\bar{\rho})$. Similarly, any $\bar{\rho}$ has a pore distribution defined by $b(\bar{r})$. The pore water distribution function, $f(\bar{r},\bar{\rho})$, is mapped in figure 2. The area of the rectangles represents the total pore volume probability space ($0 \leqslant \bar{r} \leqslant 1$, $0 \leqslant \bar{\rho} \leqslant 1$). In figure 2a, it is assumed that when $h(\bar{R})$ changes to $h(\bar{R}+d\bar{R})$, as during wetting, all pores having radii $\bar{R} \leqslant \bar{\rho} \leqslant \bar{R}+d\bar{R}$ become water-filled. In a drainage process, when $h(\bar{R})$ diminishes to $h(\bar{R}-d\bar{R})$, the pores of the group with radii of openings $\bar{r}$ in the range $\bar{R}-d\bar{R} \leqslant \bar{r} \leqslant 1$ having pore radii $\bar{R}-d\bar{R} \leqslant \bar{\rho} \leqslant \bar{R}$ are drained (Figure 2b).

Since the domains of $\bar{r}$ and $\bar{\rho}$ are positive definite, $L(\bar{R})$ and $B(\bar{R})$ can be defined as:

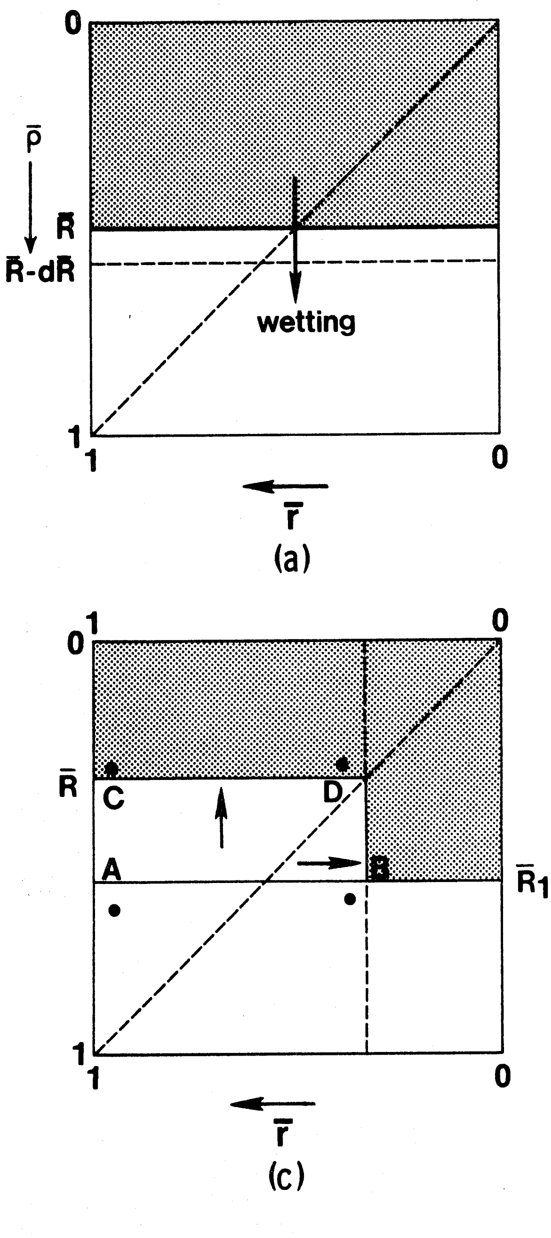


FIG. 2. The filled pore diagrams in the $\bar{r}$-$\bar{\rho}$ plane (shadowed domain) for (a) the main wetting process, (b) the main drying process, and (c) for the primary drying scanning curve.

$$L(\bar{R}) = \int_0^{\bar{R}} l(\bar{\rho})d\bar{\rho} \quad \text{and} \quad B(\bar{R}) = \int_0^{\bar{R}} b(\bar{r})d\bar{r} \quad , \quad [8]$$

so that L(0) = B(0) = 0. The effective water content after wetting along the main wetting curve ($\Theta_w(\bar{R})$) can be calculated from (Fig. 2a):

$$\Theta_w(\bar{R}) = \int_0^{\bar{R}} l(\bar{\rho})d\bar{\rho} \int_0^{1} b(\bar{r})d\bar{r} = L(\bar{R})B(1) \quad . \quad [9]$$

Assuming B(1) = 1, and since $h(\bar{R})$ is uniquely defined (whether $\bar{R}$ is $\bar{\rho}$ or $\bar{r}$), we can therefore calculate L(h):

$$L(h) = \Theta_w(h) \quad , \quad [10]$$

from which it can be shown that $L(1) = L(h_{max}) = \Theta_u$.

Following the main drying curve, the water content $\Theta_d(\bar{R})$ is obtained from figure 2b:

$$\Theta_d(\bar{R}) = \int_0^{\bar{R}} l(\bar{\rho})d\bar{\rho} \int_0^{1} b(\bar{r})d\bar{r} + \int_{\bar{R}}^{1} l(\bar{\rho})d\bar{\rho} \int_0^{\bar{R}} b(\bar{r})d\bar{r} \quad , \quad [11]$$

which, after integrating and rearranging, becomes

$$B(\bar{R}) = \frac{\Theta_d(\bar{R}) - \Theta_w(\bar{R})}{\Theta_u - \Theta_w(\bar{R})} \quad , \qquad [12]$$

or as a function of h

$$B(h) = \frac{\Theta_d(h) - \Theta_w(h)}{\Theta_u - \Theta_w(h)} \quad , \qquad [13]$$

Given the functions B(h) and L(h), we can now derive $h(\Theta)$-curves for any arbitrary scanning process. For example, the primary drying scanning curve for the drainage depicted in figure 2c can be calculated from:

$$\Theta\begin{pmatrix} & \bar{R}_1 \\ 0 & \bar{R} \end{pmatrix} = \int_0^{\bar{R}} l(\bar{\rho})d\bar{\rho} \int_0^{1} b(\bar{r})d\bar{r} + \int_{\bar{R}}^{\bar{R}_1} l(\bar{\rho})d\bar{\rho} \int_0^{\bar{R}} b(\bar{r})d\bar{r} \quad , \qquad [14]$$

which becomes

$$\Theta\begin{pmatrix} & h_1 \\ h_{min} & h \end{pmatrix} = \Theta_w(h) + \frac{[\Theta_w(h_1)-\Theta_w(h)]\ [\Theta_d(h)-\Theta_w(h)]}{[\Theta_u-\Theta_w(h)]}, \qquad [15]$$

where $\Theta\begin{pmatrix} & h_1 & \\ h_{min} & & h \end{pmatrix}$ indicates a wetting process from

$h = h_{min}$ to h_1, followed by a drying process where h decreases from $h = h_1$ to h.

With this model, the Θ(h)-relationship in the hysteresis domain can be explicitly predicted from the two main boundary curves alone. In comparing his hysteresis model with experimental results, Mualem noted that the proposed model deviated substantially from measured scanning curves if hysteresis takes place at h-values larger than the air-entry value. This behavior is characteristic of independent domain models.

II. A dependent domain model

Pores can only drain when, in addition to a continuous water phase, there is a continuous air phase. Therefore, drainage of pores in the domain of air-entry values will depend on the status of neighboring pores (9).

Pore water blockage against air entry is defined by the function $\bar{P}_d(\Theta)$ (bar indicates a verage domain dependence factor $P_d(\Theta)$ for main wetting and drying curve):

$$\bar{P}_d(\Theta) = \Delta\Theta\begin{pmatrix} h_1 \\ h \end{pmatrix} / \Delta\Theta_o \begin{pmatrix} h_1 \\ h \end{pmatrix} \quad , [16]$$

where $\Delta\Theta$ is the actual decrease in water content on drying from h_1 to h while $\Delta\Theta_o$ is the value predicted for a drying process where pore blockage does not occur (i.e., independent domain). For sufficiently low water contents, air-entry restrictions are negligible, so that $\bar{P}_d(\Theta) = 1$. $\bar{P}_d(\Theta)$ is assumed to be a function of the water content only and not of the history of the drying process (14).

Upon wetting at low water contents, one might expect pore air blockage against water entry. It was verified from experiments that air blockage was only of minor importance (9) and will therefore be neglected.

In using the dependent domain model we need to determine, in addition to L(h) and B(h), a third functional relationship, $\bar{P}_d(\Theta)$. This function can be calculated from the two main curves, and a primary drying curve.

L(h) can still be calculated from Eq. (10). From the main drying curve we can write

$$\Delta\Theta = \Theta_u - \Theta_d(h) = \bar{P}_d(\Theta)\Delta\Theta_o = \bar{P}_d(\Theta)[\Theta_u - \Theta_{do}(h)] \quad , \quad [17]$$

where $\Theta_u - \Theta_d(h)$ is the volume of pores actually drained in the drying process, while $\Theta_u - \Theta_{do}(h)$ pertains to the drained water volume as calculated with the independent domain model. Equation [17] can be transformed, using Eq. [11], to

$$\Theta_u - \Theta_d(h) = \bar{P}_d(\Theta)[1-B(h)].[\Theta_u - L(h)] \quad . \quad [18]$$

Since $\bar{P}_d(\Theta)$ was assumed to be a function of the final water content only, a third expression can be found from the primary drying curve (Eq. [14])

$$\Delta\Theta = \Theta_w(h_1) - \Theta\begin{pmatrix} & h_1 \\ h_{min} & h \end{pmatrix} = \bar{P}_d(\Theta)\Delta\Theta_o$$

$$= \bar{P}_d(\Theta)[1-B(h)][L(h_1)-L(h)] \quad . \quad [19]$$

After solving for L(h) as in the independent domain model, Eq. [18] and [19] can be used to determine simultaneously B(h) and $\bar{P}_d(\Theta)$ with the aid of the measured main drying curve and one measured primary drying curve. Once L(h), B(h) and $\bar{P}_d(\Theta)$ have been obtained from the main wetting, main drying, and one primary drying curve, any hysteretic path can be predicted.

III. Extended similarity hypothesis.

Less experimental data are required if one of the distribution functions $b(\bar{r})$ or $l(\bar{\rho})$ is known. Mualem (7) assumed

$$f(\bar{r},\bar{\rho}) = b(\bar{r})b(\bar{\rho}) \quad . \quad [20]$$

In such a case we have only one unknown function b, and the required data for determination of $f(\bar{r},\bar{\rho})$ are therefore reduced to only one of the two main curves. The assumption, $l(\bar{\rho}) = b(\bar{\rho})$, is valid if the areal and volumetric porosity are equal, as in homogeneous porous media (7).

Combination of Eq. [9] and [20] results in

$$\Theta_w(\bar{R}) = B(\bar{R})B(1) \quad , \quad [21]$$

which yields upon substitution of $\bar{R} = 1$ $(\Theta_w(1) = \Theta_u)$:

$$B(1) = (\Theta_u)^{1/2}, \quad [22]$$

and therefore

$$B(h) = \Theta_w(h)/(\Theta_u)^{1/2} \quad . \quad [23]$$

Substitution of $l(\bar{\rho}) = b(\bar{\rho})$ into Eq. [11] yields, for the main drying curve:

$$\Theta_d(h) = [2\Theta_u - \Theta_w(h)] \cdot [\Theta_w(h)/\Theta_u] \quad . \quad [24]$$

Equation [24] represents a relationship between the main wetting and the main drying curve.

IV. Modified dependent domain theory.

This last modified analysis combines the results of the dependent domain model (II) with the extended similarity hypothesis (III). Mualem's (8) expression for the domain dependence factor is obtained by substitution of $\Theta_d(h)$ in Eq. [24] for $\Theta_{do}(h)$ in Eq. [17]:

$$\bar{P}_d(\Theta) = \frac{\Theta_u(\Theta_u - \Theta_d(h))}{(\Theta_u - \Theta_w(h))^2} \quad . \quad [25]$$

Similarly, Mualem (8) derived expressions for the primary and secondary wetting and drying scanning curves. For example, the primary drying scanning curve can be obtained from:

$$\Theta\begin{pmatrix} & h_1 & \\ h_{min} & & h \end{pmatrix} = \Theta_w(h_1) - \bar{P}_d(\Theta)\Delta\Theta_o \begin{pmatrix} & h_1 & \\ h_{min} & & h \end{pmatrix}$$

where $\Delta\Theta_o$ is determined from integration over the rectangle ABCD in Fig. 2c. By substitution of B(h), as defined by Eq. [23], Θ can be expressed solely in terms of Θ_u and the main wetting curve:

$$\Theta\begin{pmatrix} & h_1 & \\ h_{min} & & h \end{pmatrix} = \Theta_w(h_1) - \bar{P}_d(\Theta) \times$$

$$\frac{(\Theta_u - \Theta_w(h))\ (\Theta_w(h_1) - \Theta_w(h))}{\Theta_u} , \qquad [26a]$$

Similarly, the primary wetting and secondary wetting and drying curves can be derived:

$$\Theta\begin{pmatrix} h_{max} & & h \\ & h_1 & \end{pmatrix} = \Theta_d(h_1) + \bar{P}_d(\Theta_1) \times$$

$$\frac{(\Theta_u - \Theta_w(h_1))\ (\Theta_w(h) - \Theta_w(h_1))}{\Theta_u} , \qquad [26b]$$

$$\Theta\begin{pmatrix} & h_1 & h \\ h_{min} & h_2 & \end{pmatrix} = \Theta\begin{pmatrix} & h_1 & \\ h_{min} & & h_2 \end{pmatrix} + \bar{P}_d(\Theta_2) \times$$

$$\frac{(\Theta_u-\Theta_w(h_2))\ (\Theta_w(h)-\Theta_w(h_2))}{\Theta_u} \quad , \qquad [26c]$$

and

$$\Theta\begin{pmatrix} h_{max} & & h_2 & \\ & h_1 & & h \end{pmatrix} = \Theta\begin{pmatrix} h_{max} & & h_2 \\ & h_1 & \end{pmatrix} - \bar{P}_d(\Theta)\ x$$

$$\frac{(\Theta_u-\Theta_w(h))\ (\Theta_w(h_2)-\Theta_w(h))}{\Theta_u} \quad . \qquad [26d]$$

METHODS

Hysteresis Model

A FORTRAN computer program, Appendix A, based on the modified dependent domain theory developed by Mualem (8) was written to simulate hysteresis. Mualem compared predicted scanning curves with experimental data for three porous media: glass beads, sand, and a sandy loam soil. Hysteresis simulations derived using this computer program were compared with those presented by Mualem (8) for the sandy soil to check for correspondence with Mualem's computer simulations. Mualem's main wetting and drying curves were fitted using van Genuchten's (16) closed-form analytical model. The domain dependence factor, $\bar{P}_d(\Theta)$, for this sandy soil was presented, as a function of Θ, in Mualem (8).

Water Flow Model

The general water flow equation in its pressure head form (Eq. [1]) can be solved, provided the specific water capacity function is known. Because there are infinitely many hysteretic curves, the water capacity function is not uniquely defined. However, the water capacity at any point along a scanning curve can be computed using the soil water pressure head derivatives of Mualem's scanning curves (i.e., Eq. [26a] to [26d]). Differentiating Eq. [26b] with respect to h, for example, yields

$$\left.\frac{d\Theta}{dh}\right|_w = \bar{P}_d(\Theta_1)\cdot\left(\frac{d\Theta_w(h)}{dh} - \frac{\Theta_w(h_1)}{\Theta_u}\cdot\frac{d\Theta_w(h)}{dh}\right) \quad , \quad [27]$$

where the derivatives on the right hand side denote the slope of the main wetting curve at soil water pressure head h.

The volumetric water content at any soil-water pressure head value can be calculated from the soil-water pressure head value at the last reversal point, provided the main wetting and drying curves, figure 3, and $\bar{P}_d(\theta)$ are defined. All hysteresis calculations (Eq. [25] and Eq. [26]) were done at the reference temperature, using Eq. [2] for

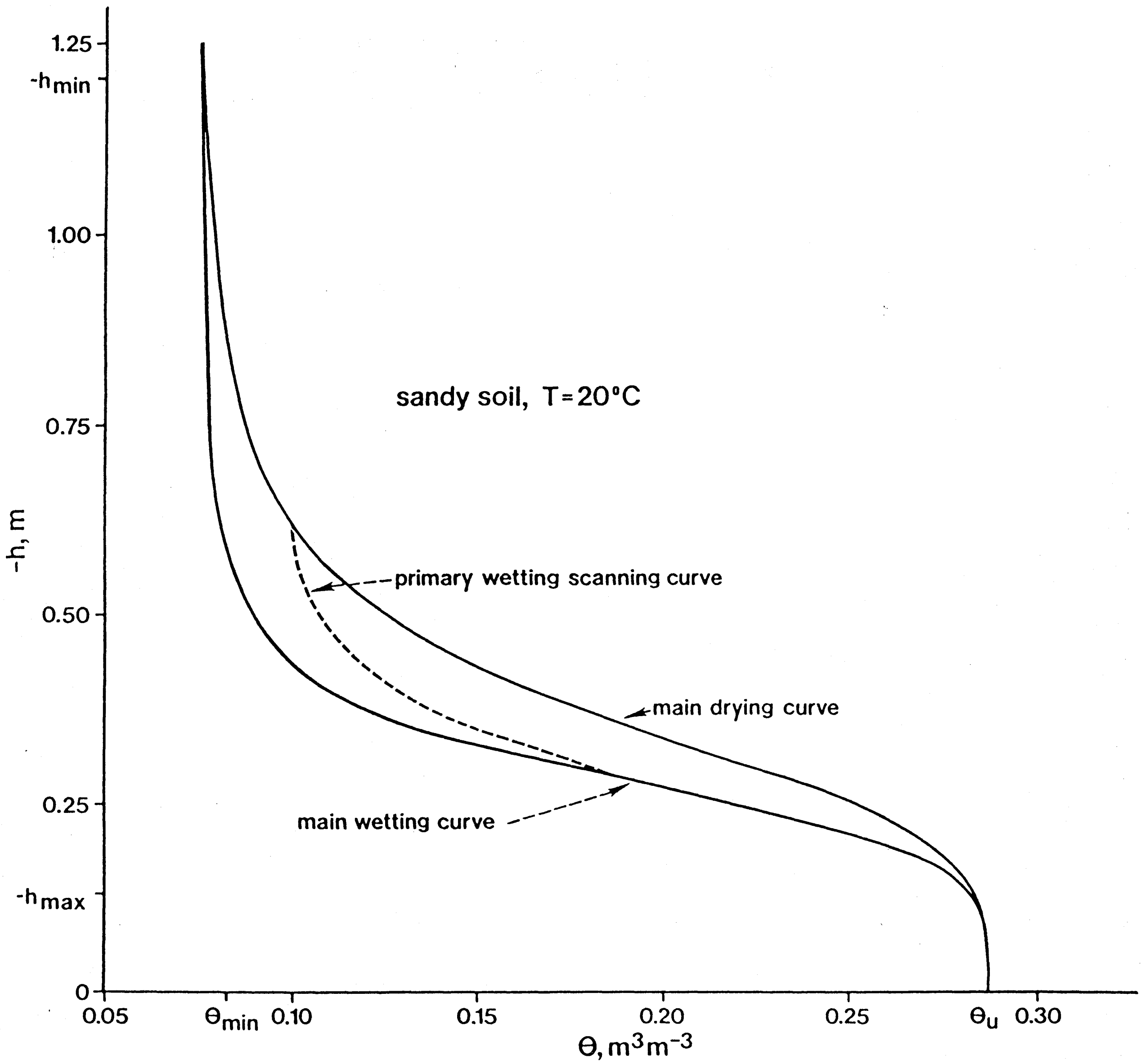


FIG. 3. Hysteretic curves at 20 °C for the sandy soil used in the computer simulations.

conversion to the reference temperature. As in Eq. [3], the temperature dependent water capacity function can be written as

$$C(h_T) = \frac{1}{\alpha(T)} \left. \frac{d\Theta}{dh_{ref}} \right|_w \quad . \qquad [28]$$

For the water flow simulations it was assumed that the K(θ)-function was dependent only on temperature (i.e., it was not subject to hysteresis). The hydraulic properties of the sandy soil, for which water movement was simulated, were experimentally determined by Haverkamp et al. (1) and are identical to those employed in simulation no.2 in Hopmans and Dane (3).

The main drying and wetting curves need to be defined to include hysteresis in the θ(h) relationship. The water retention curve determined by by Haverkamp et al. (1) was assumed to be the main drying curve. The main wetting curve was calculated from an analytical relationship between the main wetting and drying curve (Eq. [24]).

Using the same dependency of the soil's hydraulic properties on temperature as in Hopmans and Dane (3), infiltration was simulated at soil temperatures of 20 and 40 °C for both a pressure head and flux boundary condition at the soil surface. The initial and boundary conditions were:

$$h(z,0,T) = -0.615 \text{ m} , -0.8 \leqslant z \leqslant 0 \text{ m}$$

$$h(0,t,T) = -0.3 \text{ m or } q(0,t,T) = -0.1369 \text{ m h}^{-1} , \quad t > 0$$

$$h(-0.8,t,T) = -0.615 \text{ m} , \quad t > 0 \quad , \quad [29]$$

where h, t, and T were previously defined and q is the flux density of water ($q < 0$ for downward flow).

RESULTS AND DISCUSSION

Hysteresis model

Our predicted primary wetting scanning curves, figure 4, and the scanning loops, figure 5, for the sandy soil were identical from those presented by Mualem (8). These results indicated correspondence between Mualem's and the present computer simulation output. Therefore, the source code was added to an already existing water flow model (2). Computer simulations of temperature and hysteresis effects on one-dimensional, unsaturated soil water flow were performed using the expanded Hopmans and Dane model.

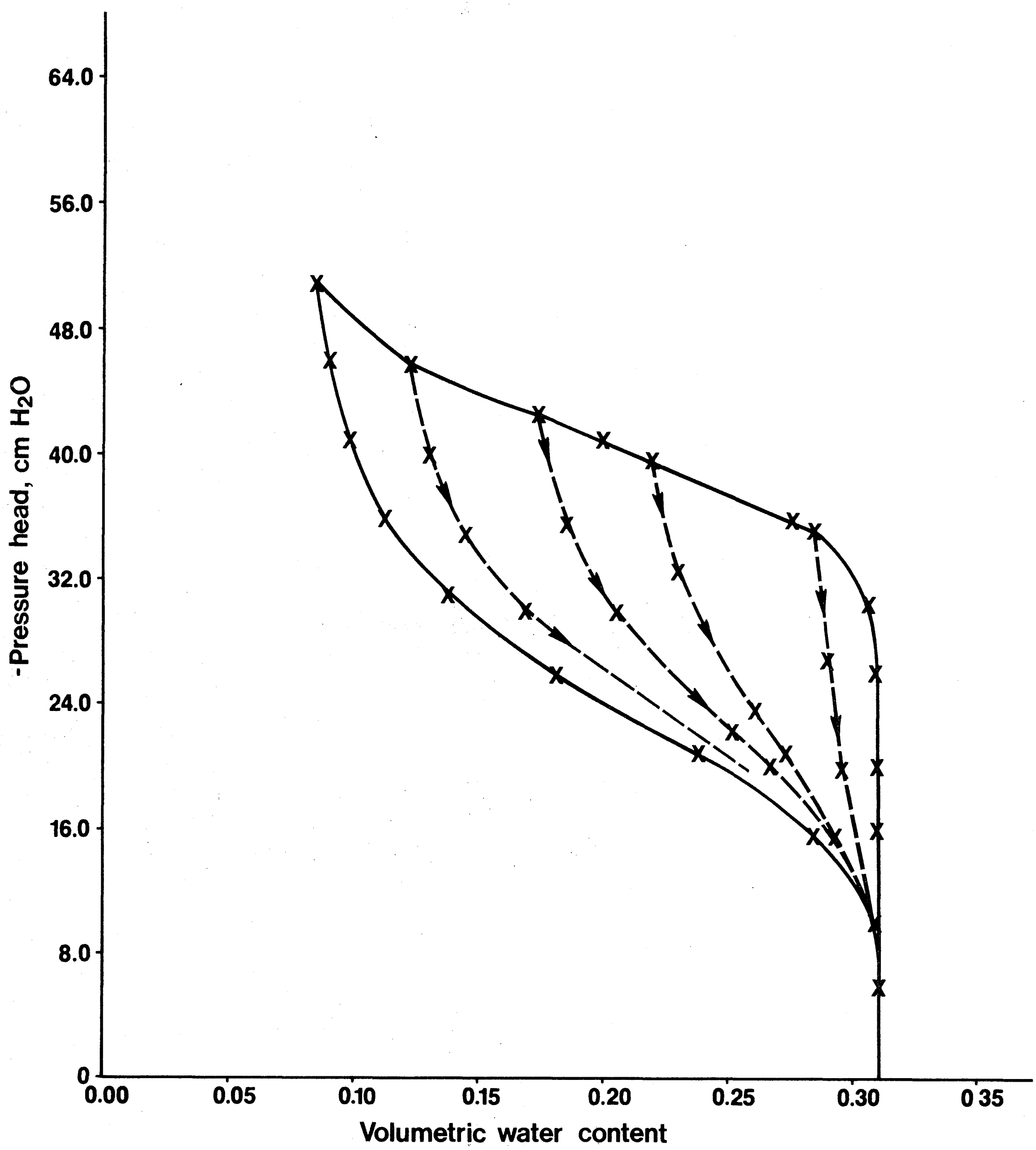


FIG. 4. Wetting scanning curves for the sandy soil predicted by Mualem's modified dependent domain theory.

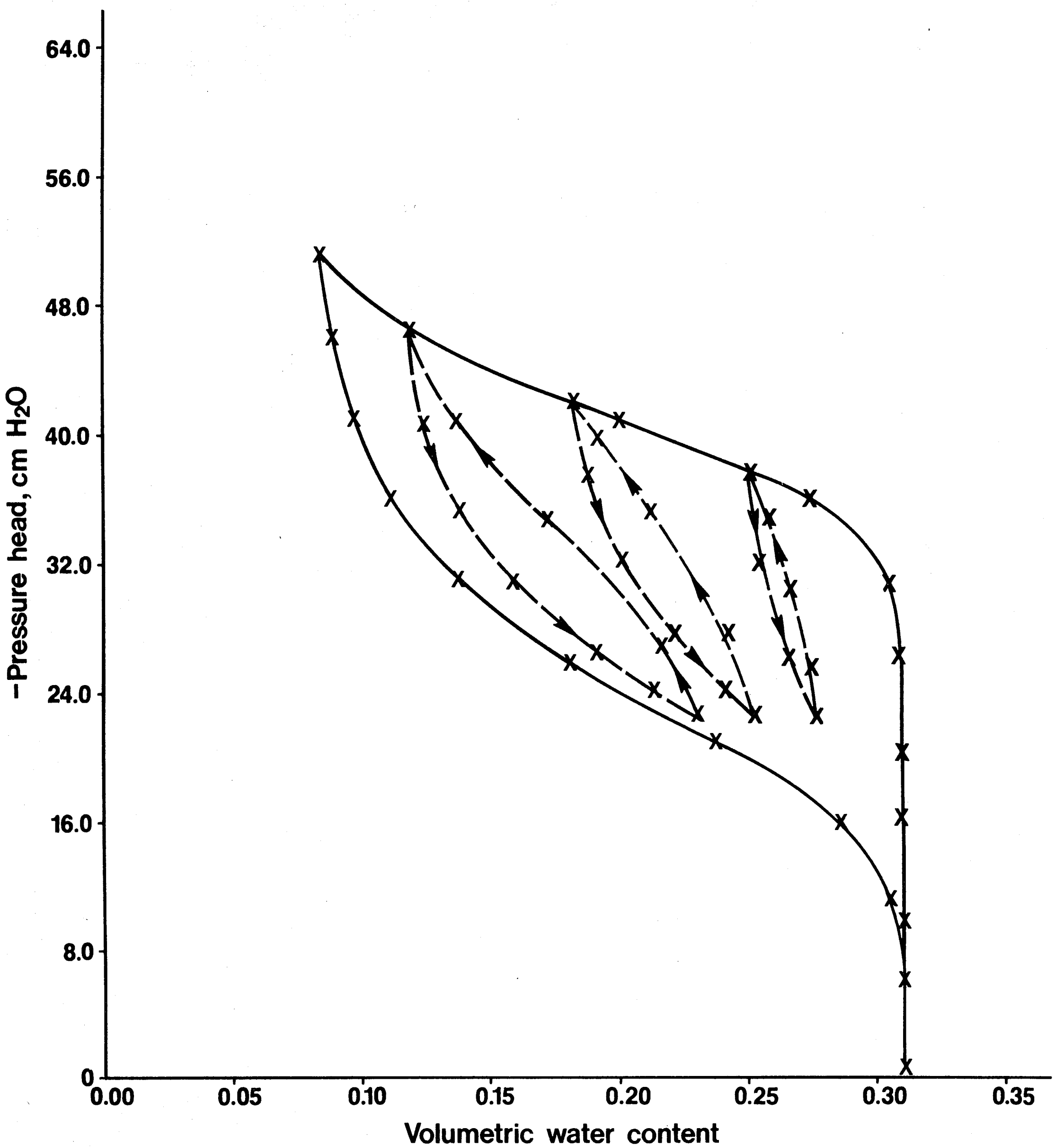


FIG. 5. Primary loops for the sandy soil predicted by Mualem's modified dependent domain theory.

Water Flow Model

Isothermal infiltration into the sandy profile, with an initial pressure head condition of -0.615 m, followed the primary wetting scanning curve (Fig. 3). Changing the soil temperature from 20 to 40 °C shifts all θ(h)-curves with equal proportion.

Constant flux infiltrations were simulated with soil temperatures of either 20 °C or 40 °C. The increase in temperature resulted in lower water content values in the transmission zone, figure 8, (3). Inclusion of hysteresis calculations had no effect on the water content distribution for either of the two temperature regimes. The water content in the wetted transmission zone attained a value that sustained a hydraulic conductivity near, or equal to, the applied flux density at the surface. This flux density was unaltered when hysteresis calculations were included. The θ(h) relationships were altered, however, producing differences in the soil water pressure head profiles.

Water content profiles resulting from a constant pressure head at the top boundary are shown in figure 6. Water storage after one hour of infiltration, ignoring hysteresis, had increased 42.4 mm in the 20 °C profile (solid line, no. 1) and 52.3 mm in the 40 °C profile (dashed line, no. 1). The increase in amount of infiltrated water with the increase in soil temperature was primarily caused by a corresponding increase in hydraulic conductivity.

The amount of infiltrated water was significantly less when hysteresis was considered (19.7 and 22.5 mm for soil temperatures of 20 °C and 40 °C, respectively). Following the primary wetting scanning curve, the water content corresponding to a surface pressure head boundary condition of -0.30 m was only .175 at 20 °C. If hysteresis was ignored (figure 6, profiles numbered as 1) the corresponding water content at a pressure head of -0.30 m was .223 (main drying curve in figure 5). The hydraulic conductivity and, therefore, the flux density sharply decreased when hysteresis was considered. The water content profiles in figure 6 indicate that the temperature effect is less pronounced when hysteresis is considered.

A program description and listing of the water flow model is given in Appendix B. With this model, actual field conditions may be simulated. However, the present model assumes a homogeneous soil profile and must be altered to include various soil layers with different hydraulic properties if it is to be used for field situations.

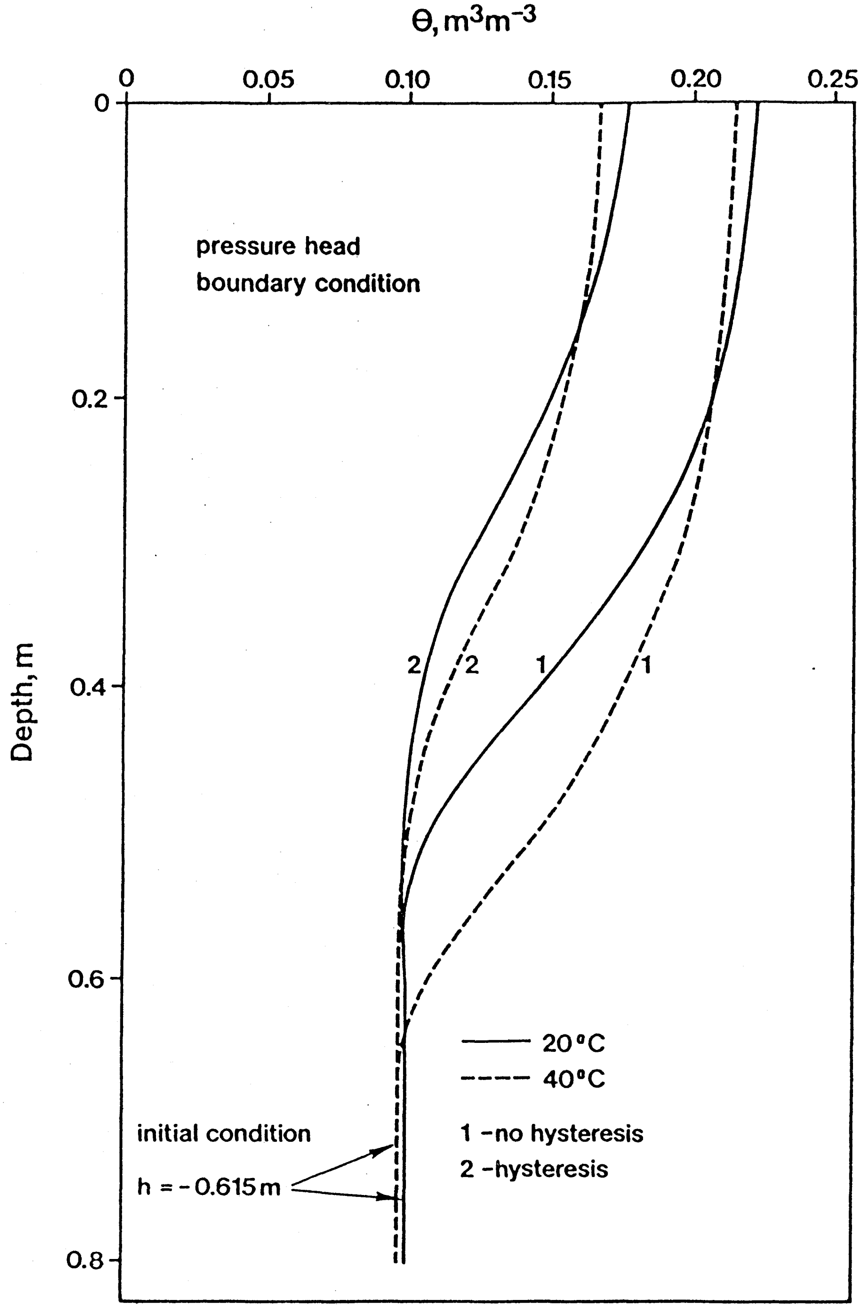


Fig. 6. Volumetric water content (θ) profiles at temperatures of 20 (solid lines) and 40 °C (dashed lines) after 1 hour of infiltration with a pressure head (h) top boundary condition.

LITERATURE CITED

(1) Haverkamp, R., M. Vauclin, J. Touma, P.J. Wierenga, and G. Vachaud. 1977. A Comparison of Numerical Simulation Models for One-dimensional Infiltration. Soil Sci. Soc. Am. J. 41:285-293.

(2) Hopmans, J.W. and J.H. Dane. 1984. Numerical Solution of the One-dimensional Water Flow Equation with and without Temperature Dependent Hydraulic Properties. Dept. of Agronomy and Soils. Departmental Series No. 94. Alabama Agricultural Experiment Station. Auburn University.

(3) ______________. 1985. Effect of Temperature Dependent Hydraulic Properties on Soil Water Movement. Soil Sci. Soc. Am. J. 49:51-58.

(4) Klute, A. 1969. The Movement of Water in Unsaturated Soils. The Progress of Hydrology. IN: Proc. of the First Int. Seminar for Hydrology Prof., Urbana, Ill., Vol. II, 821-886.

(5) Mualem, Y. 1973. Modified Approach to Capillary Hysteresis Based on a Similarity Hypothesis. Water Resour. Res. 9(5):1324-1331.

(6) ________. 1974. A Conceptual Model of Hysteresis. Water Resour. Res. 10(3):514-520.

(7) ________. 1977. Extension of the Similarity Hypothesis Used for Modeling the Soil Water Characteristics. Water Resour. Res. 13(4) :773-780.

(8) ________. 1984. A Modified Dependent-domain Theory of Hysteresis. Soil Sci. 137(5):283-291.

(9) ________ and G. Dagan. 1975. A Dependent Domain Model of Capillary Hysteresis. Water Resour. Res. 11(3):452-460.

(10) Poulovassilis, A. 1962. Hysteresis of Pore Water, an Application of the Concept of Independent Domains. Soil Sci. 93:405-412.

(11) ________________ and E.E. Childs. 1971. The Hysteresis of Pore Water: The Nonindependence of Domain. Soil Sci. 112:301-312.

(12) Talsma, T. 1970. Hysteresis in Two Sands and the Independent Domain Model. Water Resour. Res. 6(3):964-970.

(13) Topp, G.C. 1971a. Soil Water Hysteresis in Silt Loam and Clay Loam Soils. Water Resour. Res. 7(4):914-920.

(14) _________. 1971b. Soil-water Hysteresis: the Domain Model Theory Extended to Pore Interaction Conditions. Soil Sci. Soc. Am. Proc. 35:219-225.

(15) _________ and E.E. Miller. 1966. Hysteretic Moisture Characteristics and Hydraulic Conductivities for Glass-bead Media. Soil Sci. Soc. Am. Proc. 30:156-162.

(16) Van Genuchten, M.Th. 1978. Calculating the Unsaturated Conductivity with a New Closed-form Analytical Model. Water Resour. Program. Dept. of Civil Eng., Princeton Univ., 78-WR-08.

APPENDIX A

Fortran Program for Simulation of Hysteresis

Program description

Hysteresis is simulated thereby using Eq. [25] and [26]. After reading the input data (series of pressure head values) in DAIN1, the parameter IPA is set to either 0 or 1. IPA defines whether initially the soil is following the main wetting (IPA = 0) or main drying curve (IPA = 1). After each pressure head value, the program checks if a reversal point occurs. In that case:

$$UU = [H(I-1)-H(I-2)]*[H(I-1)-H(I)] > 0 \quad ,$$

where the numbers in parentheses indicate successive pressure head values with time. For UU > 0, water content values are then determined following statement number 100 in the program listing through statement number 500 (Appendix Table A-1). IPA is updated (soil is wetting or drying) and N (number of times a reversal point occurs) is increased with 1. A(N+1) takes the value of the volumetric water content at this last reversal point.

Two different procedures are followed to determine the domain dependance factor $\bar{P}_d(\theta)$. If wetting occurs (e.g. Eq.

[26a]), $\bar{P}_d$ is calculated for the water content at the last reversal point ($\bar{P}_d(\theta_N)$ in function P). However, when drying occurs (e.g. Eq. [26b]), $\bar{P}_d$ can only be implicitly determined. For a given starting value of θ (last known water content), an iteration process starts, until the two latest calculated water content values differ by less than 0.0001. The main wetting and drying curves, necessary to compute the water content at any scanning curve, are defined in the function FTH. Finally, a plot is generated that shows the scanning curves for the given pressure head sequence (Fig. 3 and 4).

Appendix Table A-1. Listing of Hysteresis Program.

```
//HYSTER JOB (AYL59,124),'JAN HOPMANS',NOTIFY=AYL59JH,MSGCLASS=P
/*ROUTE PRINT RMT4
/*JOBPARM LINES=4K,TIME=09
// EXEC FTGVCLG
//SYSIN DD *
C   GOPTIONS DEVICE=CALCOMP;
C THIS PROGRAM PLOTS THE MAIN WETTING AND DRYING CURVE FOR THE SANDY
C SOIL, DESCRIBED IN MUALEM(1984).
       DIMENSION A(1000),TH(1000),U(1000),UR(1000)
       COMMON THS
       CALL PLOTS(0,0,0)
C FIRST WE PLOT THE MAIN DRYING AND WETTING CURVE
C
       IPAR=0
       J=0
       DO 40 I=1,61,5
       J=J+1
       U(J)=-1.0*I
       TH(J)=FTH(IPAR,U(J))
       WRITE(6,1) J,TH(J),U(J)
40    CONTINUE
1     FORMAT(I5,2X,2E12.5)
       IPAR=1
       DO 55 I=1,61,5
       J=J+1
       U(J)=-1.0*I
       TH(J)=FTH(IPAR,U(J))
       WRITE(6,1) J,TH(J),U(J)
55    CONTINUE
       J=J+1
       JJ=J
       J8=J+7
       J9=J+8
       J16=J+15
       J17=J+16
       J24=J+23
       J25=J+24
       J32=J+31
C -- READ PRESSURE HEAD DATA FOR SCANNING CURVES -------
C
       READ(2,10) (U(I),I=J,J8)
       WRITE(6,10) (U(I),I=J,J8)
       READ(2,10) (U(I),I=J9,J16)
       WRITE(6,10) (U(I),I=J9,J16)
       READ(2,10) (U(I),I=J17,J24)
```

```
      WRITE(6,10) (U(I),I=J17,J24)
      READ(2,10) (U(I),I=J25,J32)
      WRITE(6,10) (U(I),I=J25,J32)
10    FORMAT(8F8.1)
C
C -- INITIALLY THE SOIL IS FOLLOWING THE MAIN WETTING OR DRYING CURVE--
C    IPA=0 -- WETTING   ,  IPA=1 -- DRYING
C
      IPA = 1
      IPAR=IABS(IPA-1)
      N = 0
      I=1
      A(JJ) = FTH(IPA,U(JJ))
      TH(JJ) = A(JJ)
      WRITE(6,1000) JJ,IPA,IPAR,U(JJ),TH(JJ)
      JJJ=J32
      JJ1=JJ+1
      DO 600 I=JJ1,JJJ
C     IF(I.EQ.2.AND.U(2).LT.U(1).AND.IPA.EQ.0) GO TO 100
C     IF(I.EQ.2.AND.U(2).GT.U(1).AND.IPA.EQ.1) GO TO 100
      IF(I.EQ.JJ+1) GO TO 50
      UU = (U(I-1)-U(I-2))*(U(I-1)-U(I))
      WRITE(6,11) UU
11    FORMAT(3X'UU ',E12.5)
      IF(UU.GT.0.0) GO TO 100
50    IF(N.EQ.0) TH(I)=FTH(IPA,U(I))
      IF(N.EQ.0) GO TO 500
      GO TO 300
100   IPAR=IPA
      IPA=IABS(IPA-1)
      N=N+1
      UR(N)=U(I-1)
      IF(N.EQ.1) A(N+1)=FTH(IPAR,UR(N))
      IF(N.GT.1) A(N+1)=TH(I-1)
      WRITE(6,101) N,UR(N),A(N+1)
101   FORMAT(' N UR A(N+1) ',I5,2E12.5)
300   IF(IPA.EQ.0) TH(I) = A(N+1) +P(IPA,UR(N),U(I),A(N+1))
      IF(IPA.EQ.0) GO TO 500
      XXX=TH(I-1)
350   TH(I)= A(N+1) +P(IPA,UR(N),U(I),XXX)
      IF(ABS(TH(I)-XXX).LE.0.0001) GO TO 500
      XXX=TH(I)
      WRITE(6,1000) I,IPA,IPAR,U(I),TH(I)
      GO TO 350
500   IF(TH(I).GT.THS) TH(I)=THS
      IF(IPA.EQ.1.AND.TH(I).GT.FTH(1,U(I))) TH(I)=FTH(1,U(I))
      IF(IPA.EQ.0.AND.TH(I).LT.FTH(0,U(I))) TH(I)=FTH(0,U(I))
600   WRITE(6,1000) I,IPA,IPAR,U(I),TH(I)
      CONTINUE
1000  FORMAT(3X,3I6,5X,2E12.5)
      CALL PLOT(0.5,0.5,-3)
      CALL AXIS(0.0,0.0,'VOLUMETRIC WATERCONTENT',+23,8.0,0.0,0.0,0.05)
```

```
      CALL AXIS(0.0,0.0,'PRESSURE HEAD',+13,8.0,90.0,0.0,8.0)
      DO 1500 K=1,JJJ
      U(K)=-U(K)
      TH(K)=TH(K)+0.08
      WRITE(6,1100) U(K),TH(K)
1100  FORMAT(2X,2(E12.5,2X))
1500  CONTINUE
      TH(JJJ+1)=0.0
      TH(JJJ+2)=0.05
      U(JJJ+1)=0.0
      U(JJJ+2)=8.0
      CALL LINE(TH,U,JJJ,1,+1,3)
      CALL PLOT(0.0,0.0,+999)
      STOP
      END
      FUNCTION FTH(IPA,U)
C THIS SUBROUTINE WILL CALCULATE THETA FROM PRESSURE HEAD AT EITHER
C THE MAIN WETTING (PAR=0.0) OR MAIN DRYING CURVE (PAR=1.0)
C
      REAL N1,M1
      COMMON THS
      UU=-U
      IF (IPA.EQ.0) GO TO 100
C  NEXT 7 STATEMENTS FOR DRYING CURVE
C
C     N1=6.97894
C     A1=0.00809
C     R1=0.0340
C     S1=0.365
      N1=12.5284
      A1=0.02422
      R1=0.06268
      S1=0.310
      THS=S1-0.08
      M1=1. - (1./N1)
      TE = (1.0/(1.0+(A1*UU)**N1))**M1
      FTH=R1+(S1-R1)*TE-0.08
C     FTH=(S1-R1)*TE
C     Q = (1.0+(A1*UU)**N1)**(-M1-1.0)
C     CC = (S1-R1)*M1*Q*N1*(A1**N1)*UU**(N1-1)
C     WRITE(6,50) UU,CC
50    FORMAT(' PRESSURE',E12.5,'WATER CAP',E12.5)
      GO TO 300
C NEXT 7 STATEMENTS FOR WETTING CURVE
100   CONTINUE
C     N1=4.25205
C     A1=0.01662
C     R1=0.0340
C     S1=0.365
      N1=5.06956
      A1=0.0428
      R1=0.07346
```

```
      S1=0.310
C     THS=S1-R1
      THS=S1-0.08
      M1=1. - (1./N1)
      TE = (1.0/(1.0+(A1*UU)**N1))**M1
      FTH=R1+(S1-R1)*TE -0.08
C     FTH=(S1-R1)*TE
C      Q = (1.0+(A1*UU)**N1)**(-M1-1.0)
C      CC = (S1-R1)*M1*Q*N1*(A1**N1)*UU**(N1-1)
C      WRITE(6,50) UU,CC
300   RETURN
      END
      FUNCTION P(IPA,UU1,UU2,TT)
C
C -- TO DETERMINE THE DOMAIN DEPENDENCE FACTOR ........--
C     P1=1.0
C     TU=0.365
C     TU=TU-.0329
C     P1=1.0
C     P1=1.06962+0.64649*TT-22.875297*TT**2+31.2176*TT**3
      TU=0.31-0.08
      TT=TT+0.08
      P1=0.881787+4.46257*TT-45.8645*TT**2+71.920154*TT**3
      TT=TT-0.08
      IF(P1.LT.0.0) P1=0.0
      IF(P1.GT.1.0) P1=1.0
      IF(IPA.EQ.0) GO TO 100
C FIRST FOR DRYING
      A1=(FTH(0,UU1)-FTH(0,UU2))/TU
      A2 = (TU-FTH(0,UU2))*A1
      P = -P1*A2
      WRITE(6,23) A1,A2,TT,P1,P
23    FORMAT(' A1 A2 TH P1 P',5E12.5)
      GO TO 200
C NEXT FOR WETTING
100   A1=(FTH(0,UU2)-FTH(0,UU1))/TU
      A2 = (TU-FTH(0,UU1))*A1
      P = +P1*A2
      WRITE(6,23) A1,A2,TT,P1,P
200   RETURN
      END
```

INPUT DATA FILE DAIN1

-10.0	-20.0	-30.0	-40.0	-39.9	-39.8	-22.5	-25.0	0001
-30.3	-35.0	-38.0	-42.0	-37.8	-32.2	-27.3	-24.1	0002
-22.5	-27.5	-35.4	-39.9	-42.0	-46.5	-41.0	-35.4	0003
-31.0	-26.4	-24.0	-22.5	-27.0	-35.0	-41.0	-46.5	0001

APPENDIX B

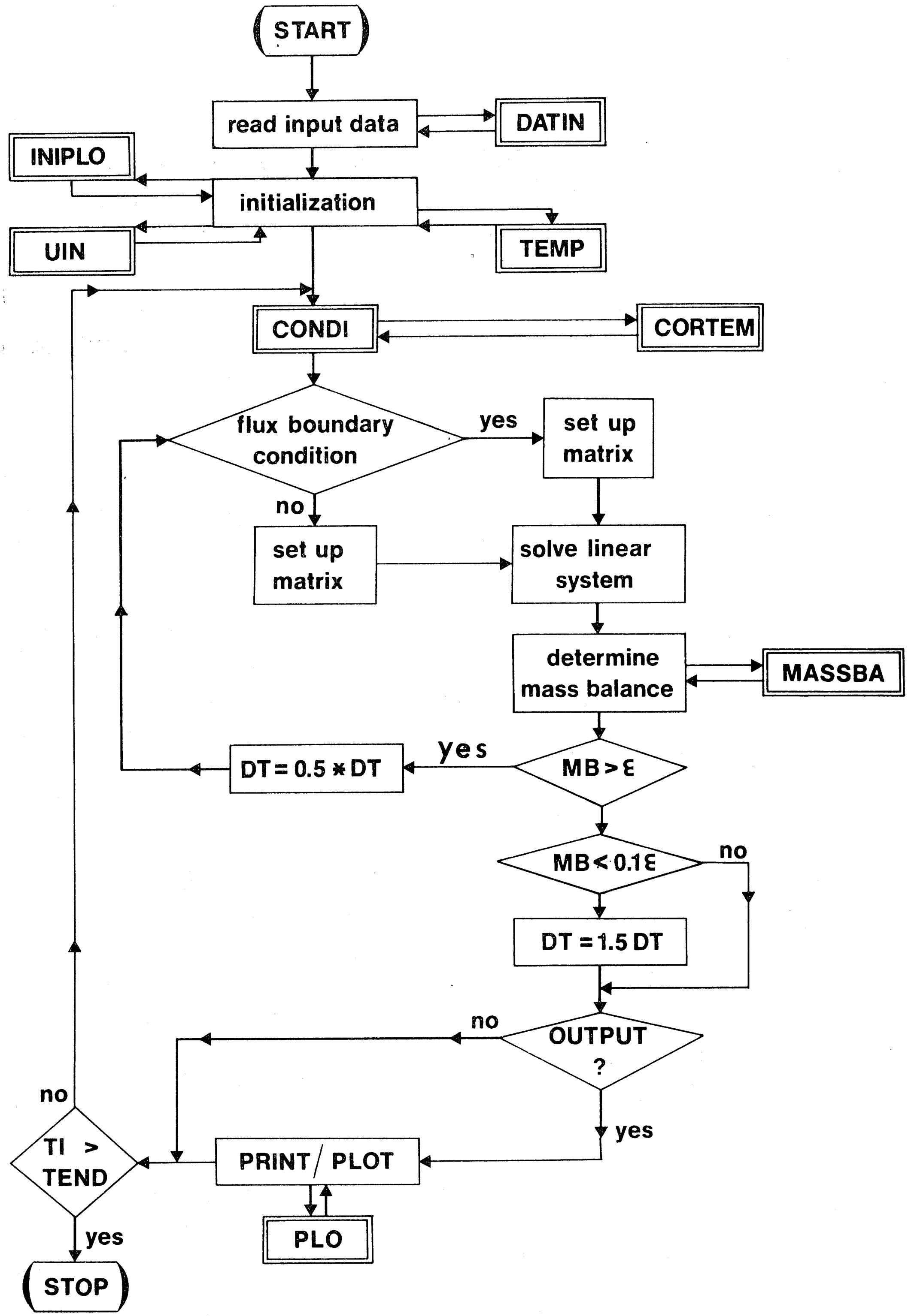
Water Flow Simulation Model

Execution of the program

The simulation model consists of a main program, 6 subroutines (INIPLO, TEMP, CORTEM, PLO, MASSBA, and CONDI), and 7 functions (FTH, FTE, PP, FC, FCC, FK, and UIN). Appendix Figure 1 shows a flow chart of the model. The input data are read from the data file DATIN. Scanning curves are determined in FTH, while FTE defines the main wetting and drying curves. Calculation of the water capacity function occurs in FC and FCC, and the hydraulic conductivity function is defined in FK. The function UIN provides the initial pressure head condition versus depth.

Upon execution of the model, a listing is printed of the soil's hydraulic properties and the initial conditions. INIPLO generates a plot of the initial pressure head and water content distribution, while TEMP sets the initial temperature distribution. CORTEM determines the temperature coefficients of pressure head and hydraulic conductivity as a function of temperature.

If the solution does not satisfy the criterion of the mass balance equation (calculations done in MASSBA), the time step is decreased and the solution process is repeated. The simulation, on the other hand, proceeds if the mass balance criterion (EPS) is met. The time step size will increase for subsequent calculations if the mass balance is less than 1/10 of the imposed criterion. The simulation stops when the maximum simulation time (TEND) is reached. The subroutine PLO genererates a plot of the water content and soil water pressure head distributions at the predefined times (O). CONDI allows for transient top and bottom boundary conditions as well as for a variable temperature distribution in both time and space.



APP. FIG. B-1. Flow chart of predictor-corrector model.

Appendix Table B-1. Definition of Main Program Variables and Subroutines

F(I)	temperature correction factor to water retention curve for grid point i
H0(I)	pressure head value at time TI and grid point i
H1(I)	pressure head value at time TIII and grid point i
H2(I)	pressure head value at time TII and grid point i
HYS1(I,1)	value of IPA at grid point i
HYS1(I,2)	nr. of reversal points at grid point i and at time TI
HYS1(I,3)	=1, if grid point i follows one of the two main curves =0, if grid point i follows any scanning curve
HYS2(I,1)	pressure head at last reversal point for grid point i
HYS2(I,2)	H2(I)
HYS2(I,3)	TH(I)
HYS2(I,4)	water content at last reversal point for grid point i
O(10)	array containing the times (s) that output is desired
TE(I)	temperature at grid point i

TH(I)	water content at grid point i
V(I)	temperature correction for hydraulic conductivity at grid point i
V0(I)	water flux at grid point i and time TI (cm s^{-1})
V2(I)	water flux at grid point i and time TII (cm s^{-1})
WAT(2)	amount of water stored in profile, cm
Z(I)	depth of grid point i (< 0), cm
ALP	specifies whether the top boundary condition is a pressure head or flux: = 0, pressure head = 1, flux
DELMO	change in stored water over current time step (cm), calculated from 2 consecutive water content profiles
DELFLU	change in stored water over current time step (cm), calculated from fluxes at boundaries
DT	time step (s)
DZ	space step (cm)
EMB	absolute mass balance at current time step (cm)
EPS	criterion for mass balance

IPA	specifies whether grid point is drying or wetting
	= 1, drying
	= 0, wetting
NO	number of times output is desired
NZ	number of space steps
NZ1	number of grid points, NZ+1
N1Z	NZ1 - 1
N2Z	NZ1 - 2
OVERAL	relative mass balance since start of simulation (%)
REMB	relative mass balance at current time step (%)
SCALF	residual water content of specific soil
TEND	end of simulation (s)
THS	saturated water content of specific soil
TI	time since start of simulation (time level j)
TII	time since start of simulation (time level j+1)
TIII	time since start of simulation (time level j+1/2)
TIHR	TII (hrs)
UBOT	bottom boundary condition (cm)
UTOP	top boundary condition, pressure head (cm) or flux (cm s^{-1})

ZBOT	depth of profile (cm)
CONDI	provides boundary conditions and temperature distribution with depth and time
CORTEM	calculates temperature correction for water retention curve and hydraulic conductivity function
FC,FCC	calculates water capacity from pressure head and temperature
FK	computes hydraulic conductivity from pressure head and temperature
FTH,FTE	computes water content from pressure head and temperature
INIPLO	plots initial pressure head and temperature distribution
MASSBA	determines mass balance
PLO	at specified times output is printed and plotted
PP	computes domain-dependent factor
TEMP	provides initial temperature distribution
UIN	provides initial pressure head distribution

Appendix Table B-2. Required Input Data

1. Input data file

Column	Format	Variable	Description
1-5	I5	NZ	number of space steps
11-20	F10.4	ZBOT	profile depth (cm)
21-30	F10.4	UTOP	top boundary condition, pressure head (cm) or flux (cm h^{-1})
31-40	F10.4	UBOT	bottom boundary condition (cm)
41-50	F10.4	DT(1)	initial time step (s)
51-60	F10.2	TEND	simulation time (s)
61-70	F10.4	EPS	error criterion mass balance
1-5	F5.1	ALP	see Appendix Table 1
6-10	I5	NO	number of times that output must be printed (TEND included)
11-15	I5	IPA	see Appendix Table 1
1-80	8F10.1	O(8)	array containing times (s) that output must be printed (TEND included)

2. Initial conditions

The initial conditions are listed in the function UIN(Z), which accepts only pressure head values, and in the subroutine TEMP(Z,TE) which contains initial temperature distribution.

3. Soil properties

Analytical expressions for $\theta(h)$, K(h), and C(h) are defined in the functions FTE, FK, and FCC.

4. Transient boundary conditions and temperature distributions can be defined in the subroutine CONDI.

Appendix Table B-3. Listing of Simulation Model

```
//HYSMODE JOB (AYL59,124),'JAN HOPMANS',NOTIFY=AYL59JH,MSGCLASS=P
//*  MSGCLASS=A
/*ROUTE PRINT RMT4
//*JOBPARM LINES=20K,TIME=00900,FORMS=6201
/*JOBPARM LINES=20K,TIME=009
//*OUTPUT L64
//* EXEC LIST,PARM='BEG=13'
//*SYSPRINT DD SYSOUT=(A,,L64)
//*SYSIN DD DSNAME=AYL59JH.INFTEM.LIB(VARTE),DISP=SHR
// EXEC ZAP
//SYSIN DD *
       DSNAME=AYL59JH.OUTDAT.CNTL
/*
//*EXEC FORTGCLG,PARM='XREF,MAP'
//*EXEC FTGVCLG,PARM='LIST,MAP'
//*EXEC FTGVCLG
//*EXEC FTGCCLG
//********* EXEC FTGVCLG
// EXEC FORTHCLG,PARM='XREF,MAP'
//*EXEC WATFIV
//*FT01F001 DD DSNAME=AYL59JH.OUTMOD.DATA,UNIT=DISK,
//* DISP=(NEW,CATLG),SPACE=(TRK,(5,5),RLSE),LABEL=RETPD=3,
//* DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//*FT03F001 DD DSN=AYL59JH.MODEL.LIB(DATIN),DISP=SHR,LABEL=(,,,IN)
//*WATFIV.SYSIN DD *
//*JOB DUMMY,PAGES=1000,TIME=1900
//FORT.SYSIN DD *
C
C      DEBUG UNIT(9),INIT(SCALF,HYS1)
C *****************************************************************
C * M O D E L                                                     *
C *             A ONE DIMENSIONAL SIMULATION MODEL                *
C *             USING THE PREDICTOR-CORRECTOR METHOD              *
C *             TIME STEP : VARIABLE                              *
C *             SPACE STEP: FIXED                                 *
C *             POSSIBLE BOUNDARY CONDITIONS:                     *
C *             1.CONSTANT PRESSURE HEAD FOR TOP AND BOTTOM       *
C *               BOUNDARY CONDITION                              *
C *             2.VARIABLE FLUX TOP BOUNDARY AND VARIABLE         *
C *               PRESSURE HEAD BOTTOM BOUNDARY CONDITION.        *
C *           - ACCOUNTS FOR TEMPERATURE EFFECTS ON HYDRAULIC     *
C *             PROPERTIES.                                       *
C *           - ACCOUNTS FOR HYSTERESIS                           *
C *                                                               *
C *  JAN HOPMANS                      VERSION DECEMBER  1984      *
```

```
C ****************************************************************
C
C
C
C
C  ***** THE DATA ARE READ FROM DISK *****************************
       INTEGER TT,HYS1(220,3)
       REAL H0(220),H1(220),H2(220),TH(220),V0(220),V2(220),DT(2)
       REAL Z(220),A(220),B(220),C(220),CC(220),D(220),WAT(2),O(10)
       REAL TE(220),F(220),V(220),CO(220),GR(220),HYS2(220,5),SCALF
       REAL SLO(220),SLOP(220)
       COMMON AAA,JJJ,NZ1,ZBOT,ALP,UTOP,EMB,REMB,DELMO,TT,DELFLU,
      1 TEND,HYS1,HYS2,SCALF,THS,IFLAG
       READ(3,25) NZ,ZBOT,UTOP,UBOT,DT(1),TEND,EPS,ALP,NO,IPA
       READ(3,26) (O(I),I=1,NO)
26    FORMAT(8F10.1)
C
25    FORMAT(I5,5X,4F10.4,F10.2,F10.4,/,F5.1,2I5)
C
C CONVERT FLUX TOP BOUNDARY TO CM/SEC
       IF(ALP.EQ.1.0) UTOP = UTOP/3600
C
C ****** THESE DATA ARE WRITTEN TO UNIT 6 ***********************
       WRITE(6,45) NZ,ZBOT,UTOP,ALP,UBOT,DT(1),TEND,EPS,IPA,
      1 (O(I),I=1,NO)
45    FORMAT(' INITIALIZATIONS AND BOUNDARY CONDITIONS ',2X,/,
      1'     NR. OF SPACE STEPS ........',I5,/,
      2'     DEPTH OF PROFILE (CM) .....',F10.5,/,
      3'     TOP BOUNDARY CONDITION ....',F10.6,'   ALPHA = ',F5.1,/,
      4'     BOTTOM BOUNDARY CONDITION .',F10.5,/,
      5'     INITIAL TIME STEP (SECON)',F10.5,/,
      6'     MODEL STOPS AT ............',F10.2,' SECON',/,
      7'     ERROR CRITERION MASS BALANCE',F10.5,3(/),
      8'     WETTING OR DRYING ..........',I5,/,
      9'     OUTPUT IS PRINTED AT ',2(/),
      92X,8F10.1)
C
C ****************************************************************
C                                                                C
C TOP BOUNDARY CONDITION:                                        C
C                     FLUX: ALP = 1.0                            C
C            PRESSURE HEAD: ALP = 0.0                            C
C BOTTOM BOUNDARY CONDITION:                                     C
C            PRESSURE HEAD ONLY                                  C
C                                                                C
C                                                                C
C                                                                C
C  R E M E M B E R   T H E   D A R C Y   C O N V E N T I O N     C
C                                                                C
C        POSITIVE FLUX ----> UPWARD FLOW                         C
C        NEGATIVE FLUX ----> DOWNWARD FLOW                       C
C                                                                C
```

```
C ****************************************************************
C
C
C
C
C     SOME INTIALIZATIONS
C
55    DELM = 0.0
       DELF = 0.0
       EPS = .001
       IFLAG=0
       NO = 1
       TT = 1
       DZ = -ZBOT/NZ
       NZ1=NZ+1
       N1Z=NZ-1
       N2Z=NZ-2
       TI=0.0
       AAA = -0.30
       JJJ = 0
       KKK = 1
       CALL PLOTS(0,0,0)
       SCALF=0.075
       DO 56 I=1,NZ1
       HYS1(I,2)=0
       HYS1(I,3)=1
       HYS2(I,1)=0.0
       HYS2(I,4)=0.0
       HYS1(I,1)=IPA
56    CONTINUE
       DO 57 I=1,NZ1
       WRITE(6,58) HYS1(I,1)
57    CONTINUE
58    FORMAT(' IPA',I5)
C
C     INITIAL VALUES OF Z,U,V AND TH AT TIME ZERO
       PMAX=0.
       DO 60 I=1,NZ1
         Z(I) = FLOAT(I-1)*DZ
         H0(I)=UIN(Z(I))
         PMAX = AMIN1(PMAX,H0(I))
60    CONTINUE
C
C THIS SUBROUTINE GIVES AND PLOTS THE TEMP. DISTRIBUTION IN PROFILE.
C AT TIME ZERO (ONLY TO BE USED IF TEMPERATURE IS CONTSTANT WITH TIME):
C
C     CALL TEMP(Z,TE)
C
C DETERMINATION OF TEMPERATURE COEFFICIENTS OF PRESSURE HEAD AND
C HYDRAULIC CONDUCTIVITY:
C
C     CALL CORTEM(TE,Z,F,V)
```

```
C
C FOR TEMP. DISTRIBUTION AND/OR BOUNDARY CONDITIONS,IF TRANSIENT:
       IF(ALP.EQ.1.0) CALL CONDI(Z,TE,UTOP,UBOT,TI,F,V,H0(NZ))
          WRITE(6,8)
8        FORMAT(1H1,' DEPTH,TEMPERATURE AND TEMPERATURE CORRECTION ',
     1 'FACTORS FOR'/'  PRESSURE HEAD AND HYDRAULIC CONDUCITIVITY RESP')
          DO 61 I=1,NZ1
          WRITE(6,9) Z(I),TE(I),F(I),V(I)
61       CONTINUE
9        FORMAT(2X,4(2X,F9.4))
C
       DO 62 I=1,NZ1
          TH(I)=FTE(IPA,H0(I),F(I)) + SCALF
           C(I)=FCC(IPA,H0(I),F(I))
           SLO(I)=C(I)
           SLOP(I)=C(I)
           HYS2(I,2)=H0(I)
           HYS2(I,3)=TH(I)
62    CONTINUE
C
C     LISTING OF THE SOIL'S PHYSICAL PROPERTIES .....
       WRITE(6,10)
10    FORMAT(1H1,' THE FOLLOWING TABLE GIVES THE HYDRAULIC PROPERTIES',
      1' OF THE SOIL CONSIDERED'/' SOIL TEMPERATURE IS REFERENCE TEMP',
      22(/),'     PRESSURE   WATER CONTENT    ',
      3'CONDUCTIVITY     WATER CAPACITY',/)
       FF= 1.
       VV= 1.
       DO 20 I=10,100,5
       U=-1.0*I
       THET = FTE(IPA,U,FF) + SCALF
       COND = FK(U,VV,FF,KKK)
       CAP  = FCC(IPA,U,FF)
       WRITE(6,50) U,THET,COND,CAP
C     WRITE(1,50) U,THET,COND,CAP
20    CONTINUE
       DO 30 I=100,1400,100
       U=-1.0*I
       THET = FTE(IPA,U,FF) + SCALF
       COND = FK(U,VV,FF,KKK)
       CAP  = FCC(IPA,U,FF)
       WRITE(6,50) U,THET,COND,CAP
C     WRITE(1,50) U,THET,COND,CAP
30    CONTINUE
       DO 40 I=1500,15500,1000
       U=-1.0*I
       THET = FTE(IPA,U,FF)+ SCALF
       COND = FK(U,VV,FF,KKK)
       CAP  = FCC(IPA,U,FF)
       WRITE(6,50) U,THET,COND,CAP
C     WRITE(1,50) U,THET,COND,CAP
50    FORMAT(2X,F8.1,7X,F5.3,8X,E12.5,7X,E12.5)
```

```
40     CONTINUE
C
C A PLOT OF INITIAL CONDITIONS:
C
        CALL INIPLO(Z,H0,TH,PMAX)
        DO 65 I=2,NZ1
              CON = -FK(.5*(H0(I)+H0(I-1)),.5*(V(I)+V(I-1)),.5*(F(I)+F(I-1
       1 )),I)
              V0(I) = CON*((H0(I)-H0(I-1))/DZ) + CON
65     CONTINUE
        V0(1) =-FK(H0(1),V(1),F(1),1)
C
C **LIST THE INITIAL VALUES OF DEPTH,THETA,PRESSURE HEAD AND FLUX RESP.
C                                                         + TEMPERATURE.
        WRITE(6,66)
66     FORMAT(1H1,/,' INITIAL CONDITIONS ARE: ',2(/),
       1'     NODE            DEPTH         THETA        PRESSURE HEAD ',
       2'      FLUX','     TEMPERATURE',' WATER CAPACITY',2(/))
        DO 70 I=1,NZ1
        WRITE(6,75) I,Z(I),TH(I),H0(I),V0(I),TE(I),C(I)
70     CONTINUE
75     FORMAT(2X,I5,4X,4(3X,E12.5),6X,F6.2,3X,E12.5)
C
        DELMO = 0.0
        DO 80 I=1,NZ
              DELMO=DELMO-(TH(I)+TH(I+1))
80     CONTINUE
        WAT(1)       = DZ*DELMO/2.
        WRITE(6,81) TI,WAT(1)
81     FORMAT(1H1,' AT TIME ',F10.5,' WATER IN PROFILE IS ',E12.5,' CM')
C
C
C IF CONTSTANT FLUX AT TOP THEN:
        IF(ALP.EQ.1.0) V0(1)=UTOP
        IF(ALP.EQ.1.0) GO TO 300
C
C FOR CONSTANT PRESSURE HEAD TOP  AND BOTTOM BOUNDARY CONDITION
        TII = TI + DT(TT)
C
CCCCCCCCCCCCCCCCC   P R E D I C T O R   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
85     DO 90 I=2,NZ
            A(I)=((2*DZ**2)/DT(TT))*FC(TH(I),H0(I),F(I),I,E)/
       1   FK(H0(I),V(I),F(I),I)
        B(I)=(FK(H0(I+1),V(I+1),F(I+1),I+1)-FK(H0(I-1),V(I-1),F(I-1),
       1 I-1))/(4*FK(H0(I),V(I),F(I),I))
            CC(I)=2*DZ*B(I)
90     CONTINUE
C      WRITE(6,91) TII
91     FORMAT(2(/),' PREDICTOR AT TIME ',F10.5,' SEC ')
        H1(1)=UTOP
        H1(NZ1)=UBOT
```

```
      D(2)=-A(2)*H0(2) - B(2)*H0(3) - CC(2) - H1(1) + B(2)*H0(1)
      D(NZ)= B(NZ)*H0(N1Z)-A(NZ)*H0(NZ)- CC(NZ)- H1(NZ1)-B(NZ)*H0(NZ1)
      DO 100 I=3,N1Z
         D(I) = B(I)*H0(I-1) - A(I)*H0(I) - B(I)*H0(I+1) - CC(I)
100   CONTINUE
      DO 105 I=2,NZ
C     WRITE(6,104) Z(I),A(I),B(I),CC(I),D(I)
104   FORMAT(' Z A B CC D',2X,5E15.5)
105   CONTINUE
C
C  SOLVE FOR PRESSURE HEAD BY THOMAS ALGORITHM.
C
      C(2)= -1.0/(2.0 + A(2))
      D(2)= -D(2)/(2.0 + A(2))
      DO 120 I=3,NZ
         Y = -2.0 - A(I) - C(I-1)
         C(I) = 1.0/Y
         D(I) = (D(I) - D(I-1))/Y
120   CONTINUE
C
      DO 130 I=2,NZ
C     WRITE(6,125) I,C(I),D(I)
125   FORMAT(2X,'C D ',I3,2E15.5)
130   CONTINUE
      C(NZ)=0.0
      H1(NZ) = D(NZ)
      N2Z = N1Z - 1
      DO 140 I=1,N2Z
         J = NZ - I
         H1(J) = D(J) - C(J)*H1(J+1)
140   CONTINUE
      DO 160 I=1,NZ1
C     WRITE(6,150) Z(I),H0(I),H1(I)
150   FORMAT(' Z U0 U1',2X,3E15.5,/)
160   CONTINUE
C
CCCCCCCCCCCCCCCCCCCC  C O R R E C T O R   CCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     WRITE(6,161) TII
161   FORMAT(2(/),' CORRECTOR AT TIME ',F10.5,' SEC ')
      DO 180 I=2,NZ
       D(I)=FTH(H1(I),F(I),I)
         A(I)=((2.0*DZ**2)/DT(TT))*FC(D(I),H1(I),F(I),I,E)/
     1   FK(H1(I),V(I),F(I),I)
      B(I)=(FK(H1(I+1),V(I+1),F(I+1),I+1) - FK(H1(I-1),V(I-1),F(I-1),
     1  I-1))/(4*FK(H1(I),V(I),F(I),I))
         CC(I)=2.0*DZ*B(I)
180   CONTINUE
      H2(1)=UTOP
      H2(NZ1)=UBOT
      D(2)=(2.0-A(2))*H0(2)-H0(3)-2.0*B(2)*H1(3)+2.0*B(2)*H1(1)-
     1 2.0*CC(2) -H2(1) - H0(1)
```

```
      D(NZ)=-H0(N1Z)+(2.0-A(NZ))*H0(NZ)+2.0*B(NZ)*H1(N1Z)
     1 -2.0*B(NZ)*H2(NZ1) - 2.0*CC(NZ) - H2(NZ1) - H0(NZ1)
      DO 200 I=3,N1Z
         D(I) = -H0(I-1) + (2.0-A(I))*H0(I) - H0(I+1) + 2.*B(I)*H1(I-1)
     1 - 2.0*B(I)*H1(I+1) - 2.0*CC(I)
200   CONTINUE
      DO 205 I=2,NZ
C     WRITE(6,104) Z(I),A(I),B(I),CC(I),D(I)
205   CONTINUE
C
C  SOLVE FOR PRESSURE HEAD BY THOMAS ALGORITHM.
C
      C(2)= -1.0/(2.0 + A(2))
      D(2)= -D(2)/(2.0 + A(2))
      DO 220 I=3,NZ
         Y = -2.0 - A(I) - C(I-1)
         C(I) = 1.0/Y
         D(I) = (D(I) - D(I-1))/Y
220   CONTINUE
C
      DO 230 I=2,NZ
C     WRITE(6,125) I,C(I),D(I)
230   CONTINUE
      C(NZ)=0.0
      H2(NZ) = D(NZ)
      DO 240 I=1,N2Z
         J = NZ - I
         H2(J) = D(J) - C(J)*H2(J+1)
240   CONTINUE
      DO 260 I=1,21
C     WRITE(6,250) Z(I),H0(I),H1(I),H2(I)
250   FORMAT(' Z H TH V TEMP C',3X,6E13.4)
260   CONTINUE
C
C  SET-UP MASS BALANCE:
      GO TO 335
C
C  FOR VARIABLE FLUX TOP BOUNDARY CONDITION ::
C
300   TII = TI + DT(TT)
C
CCCCCCCCCCCCCCCCCCCCCCCCC   PREDICTOR   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
301   CONTINUE
      UUU=H0(NZ)
      TIII=TI + .5*DT(TT)
      CALL CONDI(Z,TE,UTOP2,UBOT2,TIII,F,V,UUU)
      CALL CONDI(Z,TE,UTOP1,UBOT1,TI,F,V,UUU)
      D1 = 2*DZ*(UTOP1 + FK(H0(1),V(1),F(1),1))/FK(H0(1),V(1),F(1),1)
      D2 = 2*DZ*(UTOP2 + FK(H0(1),V(1),F(1),1))/FK(H0(1),V(1),F(1),1)
      FU1 = H0(2) + D1
      DO 303 I=1,NZ
```

```
          PPPP=FC(TH(I),H0(I),F(I),I,SLO(I))
          A(I)=((2*DZ**2)/DT(TT))*PPPP/
     1   FK(H0(I),V(I),F(I),I)
      SLO(I)=PPPP
           IF(I.EQ.1) GO TO 302
      B(I)=(FK(H0(I+1),V(I+1),F(I+1),I+1) - FK(H0(I-1),V(I-1),F(I-1),
     1   I-1))/(4*FK(H0(I),V(I),F(I),I))
302     IF(I.EQ.1) B(I) = (FK(H0(I+1),V(I+1),F(I+1),I+1)-
     1   FK(FU1,V(I),F(I),I))/(4*FK(H0(I),V(I),F(I),I))
          CC(I)=2*DZ*B(I)
303  CONTINUE
C    WRITE(6,91) TII
      H1(NZ1) = UBOT2
      D(1)=-A(1)*H0(1) - CC(1) + B(1)*D1 - D2
      D(NZ)= B(NZ)*H0(N1Z)-A(NZ)*H0(NZ)- CC(NZ)- H1(NZ1)-B(NZ)*H0(NZ1)
      IF(TII.GT.95000.0) WRITE(6,308) TII,H0(1),H0(2)
308   FORMAT(2X,3E12.5)
      DO 304 I=2,N1Z
          D(I) = B(I)*H0(I-1) - A(I)*H0(I) - B(I)*H0(I+1) - CC(I)
304  CONTINUE
      DO 305 I=1,NZ
C    WRITE(6,104) Z(I),A(I),B(I),CC(I),D(I)
305  CONTINUE
C
C SOLVE FOR PRESSURE HEAD BY THOMAS ALGORITHM:
      C(1)= -2.0/(2.0 + A(1))
      D(1)= -D(1)/(2.0 + A(1))
      DO 306 I=2,NZ
          Y = -2.0 - A(I) - C(I-1)
          C(I) = 1.0/Y
          D(I) = (D(I) - D(I-1))/Y
306  CONTINUE
C
      DO 307 I=1,NZ
C    WRITE(6,125) I,C(I),D(I)
307  CONTINUE
      C(NZ)=0.0
      H1(NZ) = D(NZ)
      DO 310 I=1,N1Z
          J = NZ - I
          H1(J) = D(J) - C(J)*H1(J+1)
310  CONTINUE
      DO 311 I=1,NZ1
C    WRITE(6,150) Z(I),H0(I),H1(I)
311  CONTINUE
C
CCCCCCCCCCCCCCCCCCCCCCCCCC   CORRECTOR   CCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      UUU = H1(NZ)
C    WRITE(6,161) TII
      CALL CONDI(Z,TE,UTOP,UBOT,TII,F,V,UUU)
      CALL CONDI(Z,TE,UTOP2,UBOT2,TIII,F,V,UUU)
```

```
       D2 = 2*DZ*(UTOP2 + FK(H1(1),V(1),F(1),1))/FK(H1(1),V(1),F(1),1)
       D3 = 2*DZ*(UTOP+ FK(H1(1),V(1),F(1),1))/FK(H1(1),V(1),F(1),1)
       FU1 = H1(2) + D2
C     WRITE(6,312) D2,FU1
312   FORMAT(' D2,FU1',2E12.5)
       DO 315 I=1,NZ
          D(I)= FTH(H1(I),F(I),I)
          PPPP=FC(D(I),H1(I),F(I),I,SLOP(I))
          A(I)=((2.0*DZ**2)/DT(TT))*PPPP/
     1   FK(H1(I),V(I),F(I),I)
       SLOP(I)=PPPP
          IF(I.EQ.1) GO TO 313
       B(I)=(FK(H1(I+1),V(I+1),F(I+1),I+1) - FK(H1(I-1),V(I-1),F(I-1),
     1   I-1))/(4*FK(H1(I),V(I),F(I),I))
313      IF(I.EQ.1) B(I) = (FK(H1(I+1),V(I+1),F(I+1),I+1)-
     1   FK(FU1,V(I),F(I),I))/(4*FK(H1(I),V(I),F(I),I))
C     WRITE(6,314) I,B(I)
314   FORMAT(' I B(I)',I3,E12.5)
          CC(I)=2.0*DZ*B(I)
315   CONTINUE
       H2(NZ1)=UBOT
       D(1)=(2.0-A(1))*H0(1)-2*H0(2)+2.0*B(1)*D2-D1-D3-2*CC(1)
       D(NZ)=-H0(N1Z)+(2.0-A(NZ))*H0(NZ)+2.0*B(NZ)*H1(N1Z)
     1 -2.0*B(NZ)*H2(NZ1) - 2.0*CC(NZ) - H2(NZ1) - H0(NZ1)
       DO 320 I=2,N1Z
          D(I) = -H0(I-1) + (2.0-A(I))*H0(I) - H0(I+1) + 2.*B(I)*H1(I-1)
     1 - 2.0*B(I)*H1(I+1) - 2.0*CC(I)
320   CONTINUE
       DO 321 I=1,NZ
C     WRITE(6,104) Z(I),A(I),B(I),CC(I),D(I)
321   CONTINUE
C
C SOLVE FOR PRESSURE HEAD BY THOMAS ALGORITHM:
       C(1)= -2.0/(2.0 + A(1))
       D(1)= -D(1)/(2.0 + A(1))
       DO 325 I=2,NZ
          Y = -2.0 - A(I) - C(I-1)
          C(I) = 1.0/Y
          D(I) = (D(I) - D(I-1))/Y
325   CONTINUE
C
       DO 326 I=1,NZ
C     WRITE(6,125) I,C(I),D(I)
326   CONTINUE
       C(NZ)=0.0
       H2(NZ) = D(NZ)
       DO 330 I=1,N1Z
          J = NZ - I
          H2(J) = D(J) - C(J)*H2(J+1)
330   CONTINUE
C     DO 331 I=1,16
C     WRITE(6,250) Z(I),H0(I),H1(I),H2(I),TH(I)
```

```
331    CONTINUE
C
335    CALL MASSBA(TH,H2,Z,V0,V2,DT,DZ,V,F,CO,GR)
C
C      DO 3 I=1,NZ1
C      WRITE(6,2)HYS1(I,1),HYS1(I,2),HYS1(I,3)
C2     FORMAT(3I7)
C3     CONTINUE
C      DO 5 I=1,NZ1
C      WRITE(6,4)HYS2(I,1),HYS2(I,2),HYS2(I,3),HYS2(I,4),HYS2(I,5)
C4     FORMAT(5E12.5)
C5     CONTINUE
C      IF(TII.GT.1.00) EPS = 0.001
        WRITE(6,501) TII,DELMO,DELFLU,EMB,REMB
501    FORMAT(' TIME DELMO DELFLU EMD REMB ',5E15.5)
        TT = 1
        IF(EMB.GT.EPS) GO TO 510
        IF(EMB.LT.0.1*EPS) DT(TT) = 1.5*DT(TT)
        IF(TII.LT.7200.AND.DT(TT).GT.100.0) DT(TT)=100.0
C
C TIME STEP IS DECREASED IF THE REL. MASS BALANCE IS TOO LARGE:
C
C      IF(TII.GT.50.0.AND.REMB.GT.0.5) GO TO 510
        GO TO 520
510    DT(TT) = 0.5* DT(TT)
        TII = TI + DT(TT)
        IF(TII.GT.O(NO)) GO TO 900
        IF(TI.EQ.0.0) GO TO 520
        IF(TI.GT.1000.0.AND.DT(TT).LT.0.5) GO TO 520
C      DO 515 I=1,NZ1
C      TH(I)=FTH(H0(I),F(I),I)
C515    CONTINUE
        IF(ALP.EQ.1.0) GO TO 301
        GO TO 85
520    TI = TII
        WAT(1) = WAT(1) + DELMO
        DELF = DELF + DELFLU
        DELM = DELM + DELMO
        OVERAL = ((ABS(DELM-DELF))/DELF)*100
        WRITE(6,502) TII,WAT(1),EMB,REMB,OVERAL,UTOP,UBOT
502    FORMAT(' TI WAT EMB RE OVER TOP BOT',F8.1,6(E11.3))
C      WRITE(6,451) TII, WAT(1)
451    FORMAT(/,' WATER IN PROFILE AT TIME ',F10.2,'SEC:',E12.5,'CM')
        IF(TI.EQ.TEND) GO TO 1000
        IF(TI.EQ.7200.0) DT(TT)=5.0
        TII = TI + DT(TT)
        IF(TI.EQ.O(NO)) GO TO 530
        IF(TII.GT.O(NO)) GO TO 900
530    TIHR=TI/3600
        IF(TI.EQ.O(NO)) WRITE(6,531) TIHR
531    FORMAT(1H1,' DEPTH, PRESSURE HEAD, THETA, FLUX AND TEMPERATURE ',
       1/,' AT TIME:   ',F10.3,'  HOURS',1(/))
```

```
      DO 700 I=1,NZ1
          HYS2(I,3)=TH(I)
          TH(I) = FTH(H2(I),F(I),I)
      IF(TI.EQ.O(NO)) WRITE(6,250) Z(I),H2(I),TH(I),V2(I),TE(I),SLOP(I)
          H0(I)=H2(I)
          HYS2(I,2)=H0(I)
          V0(I) =V2(I)
700   CONTINUE
      IF(TI.EQ.O(NO)) GO TO 800
      GO TO 750
800   AAA = AAA - 0.3
      JJJ = JJJ + 1
      WRITE(6,451) TI, WAT(1)
      CALL PLO(Z,TH,TI,O,PMAX,H0)
      NO = NO + 1
      IF(TII.GT.O(NO)) GO TO 900
750   CONTINUE
      WRITE(6,751) H0(1),TH(1),H0(40),TH(40)
751   FORMAT(' H AND THETA',4E12.5)
      IF(ALP.EQ.1.0) GO TO 301
      GO TO 85
900   TT=2
      DT(TT) = O(NO) - TI
      TII= TI + DT(TT)
      DO 940 I=1,NZ1
          HYS2(I,3)=TH(I)
          H0(I)=H2(I)
          TH(I) = FTH(H0(I),F(I),I)
          V0(I) =V2(I)
          HYS2(I,2)=H0(I)
940   CONTINUE
      IF(ALP.EQ.1.0) GO TO 301
      GO TO 85
1000  TIHR=TI/3600
      WRITE(6,531) TIHR
      DO 950 I=1,NZ1
            TH(I) = FTH(H2(I),F(I),I)
            WRITE(6,250) Z(I),H2(I),TH(I),V2(I),TE(I)
950   CONTINUE
      AAA = AAA - 0.3
      JJJ = JJJ + 1
      WRITE(6,451) TI, WAT(1)
      CALL PLO(Z,TH,TI,O,PMAX,H2)
      NO = NO + 1
      CALL PLOT(0.0,0.0,-999)
C     CALL PLOCO(CO,GR,Z)
      CALL PLOT(0.0,0.0,+999)
      STOP
      END
        FUNCTION FTH(U,F,IJ)
C ** COMPUTES WATER CONTENT AT ANY SCANNING CURVE FROM THE TWO
C ** MAIN CURVES (DEFINED IN FTE)...
```

```
C       DEBUG UNIT(9),INIT(SCALF,HYS1)
         REAL U,F,HY2(220,5),AA,UR,SCALF
         INTEGER HY1(220,3),MC(220)
         COMMON A1,J1,J2,A2,A3,A4,A5,A6,A7,J3,A8,A9,HY1,HY2,SCALF,THS
         IPA=HY1(IJ,1)
         IPAR=IABS(IPA-1)
         N = HY1(IJ,2)
         IF(N.GT.0) UR= HY2(IJ,1)
         AA=HY2(IJ,4)
         MC(IJ)=HY1(IJ,3)
C       WRITE(6,1) IJ,N,IPA,UR,AA,SCALF
1       FORMAT(3I6,3E12.5)
         IF(IPA.EQ.0.AND.(U-HY2(IJ,2)).LT.-0.001) GO TO 100
         IF(IPA.EQ.1.AND.(U-HY2(IJ,2)).GT.0.001) GO TO 100
         IF(HY1(IJ,2).EQ.0) FTH=FTE(IPA,U,F)
         IF(HY1(IJ,2).EQ.0) MC(IJ)=1
         IF(HY1(IJ,2).EQ.0) GO TO 500
         GO TO 300
100     IPAR=IPA
         IPA=IABS(IPA-1)
         HY1(IJ,1)=IPA
         N=N+1
         UR=HY2(IJ,2)
         MC(IJ)=0
C       WRITE(6,1) IJ,N,IPA,UR,AA,SCALF
         IF(N.EQ.1) AA=FTE(IPAR,UR,F)
         IF(N.GT.1) AA=HY2(IJ,3) -SCALF
         HY2(IJ,4)=AA
300     IF(IPA.EQ.0) FTH=AA+PP(IPA,UR,U,AA,F)
C       WRITE(6,2) AA,FTH
         IF(IPA.EQ.0) GO TO 500
         XXX=HY2(IJ,3)-SCALF
350     FTH=AA+PP(IPA,UR,U,XXX,F)
         XYZ = ABS(FTH-XXX)
         IF(XYZ.LE.0.0001) GO TO 500
         XXX=FTH
C       WRITE(6,351) XXX,XYZ
351     FORMAT(' ITER',2E12.5)
         GO TO 350
500     IF(FTH.GT.THS) FTH=THS
         IF(IPA.EQ.1.AND.FTH.GE.FTE(1,U,F)) FTH=FTE(1,U,F)
         IF(IPA.EQ.1.AND.FTH.GE.FTE(1,U,F)) MC(IJ)=1
         IF(IPA.EQ.0.AND.FTH.LE.FTE(0,U,F)) FTH=FTE(0,U,F)
         IF(IPA.EQ.0.AND.FTH.LE.FTE(0,U,F)) MC(IJ)=1
         IF(N.GT.0) HY2(IJ,1)=UR
         HY1(IJ,1)=IPA
         HY1(IJ,2)=N
         HY1(IJ,3)=MC(IJ)
C       WRITE(6,2) FTH,SCALF
2       FORMAT(2X,2E12.5)
         FTH=FTH+SCALF
C       WRITE(6,2) FTH,SCALF
```

```
      RETURN
      END
      FUNCTION FTE(IPA,U,F)
C     DEBUG UNIT(9),INIT(SCALF,HYS1)
C ** DEFINITION OF MAIN WETTING AND DRYING CURVE....
      REAL U,F,UU,TE,HY2(220,5)
      REAL M1,N1
      INTEGER HY1(220,3)
      COMMON B1,J1,J2,B2,B3,B4,B5,B6,B7,J3,A8,A9,HY1,HY2,SCALF,THS
      SCALF=0.075
      UU=-(1./F)*U
C     IF(HH.GE.-1.0) GO TO 10
C     GO TO 30
C     WRITE(6,1) U,UU,IPA
1     FORMAT(2E12.5,I4)
      IF(IPA.EQ.0) GO TO 100
      UU = -UU
      FTE = 1.611E+06*.212/(1.611E+06+ABS(UU)**3.96)+0.075
      THS=.287-SCALF
C     WRITE(6,2) N1,M1
2     FORMAT(2E12.5)
      FTE=FTE-SCALF
C     WRITE(6,2) TE,FTE
      GO TO 300
100   N1=5.19408
      A1=0.0371
      R1=0.0744
      S1=0.287
      THS=S1-SCALF
      M1=1. - (1./N1)
      TE= (1.0/(1.0+(A1*UU)**N1))**M1
      FTE=R1+(S1-R1)*TE-SCALF
C     WRITE(6,2) FTE,SCALF
300   RETURN
      END
      FUNCTION PP(IPA,UU1,UU2,T2,F)
C     DEBUG UNIT(9),INIT(SCALF,HYS1)
C** COMPUTATION OF DOMAIN DEPENDENCE FACTOR
      REAL SCALF,UU1,UU2,T2,HY2(220,5)
      INTEGER HY1(220,3)
      COMMON B1,J1,J2,B2,B3,B4,B5,B6,B7,J3,A8,A9,HY1,HY2,SCALF,THS
      SCALF=0.075
      TU=0.283-SCALF
      T2=T2+SCALF
C     WRITE(6,3) SCALF,TU,T2
3     FORMAT(' SCALF TU T2 IN PP',3E12.5)
      P1=2.06228-24.54188*T2+168.526*T2**2-380.0273*T2**3
      T2=T2-SCALF
      IF(P1.LT.0.0) P1=0.0
      IF(P1.GT.1.0) P1=1.0
      IF(IPA.EQ.0) GO TO 100
C FIRST FOR DRYING
```

```
         A1=(FTE(0,UU1,F)-FTE(0,UU2,F))/TU
         A2=(TU-FTE(0,UU2,F))*A1
         PP=-P1*A2
         GO TO 200
C NEXT FOR WETTING
100     A1=(FTE(0,UU2,F)-FTE(0,UU1,F))/TU
         A2=(TU-FTE(0,UU1,F))*A1
         PP=+P1*A2
C       WRITE(6,2) A1,A2,PP,TT,SCALF
2       FORMAT(5E12.5)
200     RETURN
         END
         FUNCTION FC(T1,U,F,IJ,SL)
C       DEBUG UNIT(9),INIT(SCALF,HYS1)
C ** CALCULATION OF WATER CAPACITY AT ANY SCANNING CURVE FROM
C ** WATER CAPACTITY AT 2 MAIN CURVES (DEFINED IN FCC)...
         REAL U,F,UU,N1,M1,HY2(220,5),T1,TTT,SCALF,SL
         INTEGER HY1(220,3)
         COMMON A1,J1,J2,A2,A3,A4,A5,A6,A7,J3,A8,A9,HY1,HY2,SCALF,THS
         IPA=HY1(IJ,1)
         TU=0.283-0.075
         SCALF=0.075
         IF(HY1(IJ,3).EQ.1) GO TO 300
         IF(IPA.EQ.0) GO TO 200
100     B1=-FCC(0,U,F)
         B2=((-FTE(0,HY2(IJ,1),F))/TU)*FCC(0,U,F)
         B3=(2/TU)*(FTE(0,U,F))*FCC(0,U,F)
         B4=2.06228-24.54188*T1+168.526*T1**2-380.0273*T1**3
         IF(B4.LT.0.0) B4=0.0
         IF(B4.GT.1.0) B4=1.0
         FC =-B4*(B1 +B2+B3)
C       B5=-24.54188 + 337.052*T1 - 1140.0819*T1**2
C       B6 = SL
C       B7= FTE(0,HY2(IJ,1),F) - FTE(0,U,F) - FTE(0,U,F)*
C     1(FTE(0,HY2(IJ,1),F))/TU + ((FTE(0,U,F))**2)/TU
C       FC = FC -(B5*B6*B7)
         GO TO 400
200     B1 = FCC(0,U,F)
         B2=((-FTE(0,HY2(IJ,1),F))/TU)*FCC(0,U,F)
         TTT=HY2(IJ,4)+0.075
         B3=2.06228-24.54188*TTT+168.526*TTT**2-380.0273*TTT**3
C       WRITE(6,12) TTT,B3
12      FORMAT(' TTT B3',2E12.5)
         IF(B3.LT.0.0) B3=0.0
         IF(B3.GT.1.0) B3=1.0
         FC =+B3*(B1+B2)
         GO TO 400
300     FC = FCC(IPA,U,F)
C400     WRITE(6,11) TTT,U,F,FC
11      FORMAT('T1 U F FC',4E12.5)
400     RETURN
         END
```

```
      FUNCTION FCC(IP,U,F)
C ** WATER CAPACITY VALUES FROM PRESSURE HEAD DATA AT 2 MAIN CURVES.
C     DEBUG UNIT(9),INIT(SCALF,HYS1)
      REAL N1,M1,U,UU
      COMMON B1,J1,J2,B2,B3,B4,B5,A6,A7,J3,A8,A9,HY1,HY2,SCALF,THS
C
      UU=-U/F
      IF(U.LT.-130.0) FCC=0.0
      IF(U.LT.-130.0) GO TO 100
C     GO TO 30
      IF(IP.EQ.0) GO TO 50
      UU = -UU
      FCC=1.611E+06*.212*3.96*ABS(UU)**2.96
      FCC=FCC/(1.611E+06+ABS(UU)**3.96)**2
      IF(UU.GT.-1.0) FCC=0.0
      GO TO 100
50    N1=5.19408
      A1=0.0371
      R1=0.0744
      S1=0.287
      M1=1. - (1./N1)
      Q=(1.0+(A1*UU)**N1)**(-M1-1.0)
      FCC=(S1-R1)*M1*Q*N1*(A1**N1)*UU**(N1-1)
100   FCC=FCC/F
C     WRITE(6,11) F,UU,FCC
11    FORMAT(' UU FCC',3E12.5)
      RETURN
      END
      FUNCTION FK(H,V,F,JJ)
C ** HYDRAULIC CONDUCTIVITY VALUES FROM PRESSURE HEAD DATA.
C     DEBUG UNIT(9),INIT(SCALF,HYS1)
      REAL H,V,F,HH,WC
      INTEGER JJ
      HH= (1.0)*H
      IF(HH.GT.0.0) HH=0.0
      WC=FTH(HH,F,JJ)
      GO TO 30
C     GOTO 10
      FK=34.*1.175E+06/(3600*(1.175E+06+ABS(HH)**4.74))
C     FK=4.428E-02*124.6/(3600*(124.6+ABS(HH)**1.77))
      FK = V*FK
      GO TO 30
10    HH=ABS(H)
      F=-0.58420234 - 0.09268778*HH + 0.00051873*HH**2
      FK=10**F
30    FK=EXP(-23.256418+212.107988*WC-1013.24235*WC**2+
     1 1744.592101*WC**3)
      FK=V*FK
C 30  RETURN
      END
      FUNCTION UIN(Z)
C ** THE INITIAL CONDITIONS, EXPRESSED IN PRESSURE HEAD VALUES AS A
```

```
C ** FUNCTION OF DEPTH.
C       DEBUG UNIT(9),INIT(SCALF,HYS1)
        UIN = -61.5
        RETURN
        END
        SUBROUTINE INIPLO(Z,H0,TH,PMAX)
C       DEBUG UNIT(9),INIT(SCALF,HYS1)
        COMMON AAA,JJJ,NZ1,ZBOT
        REAL Z(220),TH(220),H0(220)
        CALL PLOT(1.0,9.0,-3)
        K=NZ1+1
        L = K + 1
        Z(K) = 0.0
        Z(L) =ZBOT/8.0
        H0(K)= 0.
        H0(L)=PMAX/4.
        TH(K)=0.
        TH(L)=0.05
       CALL AXIS(0.0,0.0,'PRESSURE HEAD THETA',+19,8.0,0.0,0.0,H0(L))
       CALL AXIS(0.0,0.0,'DEPTH CM',-8,8.0,270.0,0.0,Z(L))
       CALL LINE(H0,Z,NZ1,1,+1,1)
       CALL AXIS(0.0,0.3,' ',+1,8.0,0.0,0.0,0.05)
       CALL LINE(TH,Z,NZ1,1,+1,2)
       CALL SYMBOL(2.0,-8.2,0.10,'THETA AND PRESSURE HEAD',0.0,+23)
       CALL PLOT(0.0,0.0,-999)
       RETURN
       END
        SUBROUTINE TEMP(Z,TE)
C  SETS AND PLOTS INITIAL TEMPERATUE DISTRIBUTION IN PROFILE
C
C      DEBUG UNIT(9),INIT(SCALF,HYS1)
        COMMON AAA,JJJ,NZ1,ZBOT
        REAL Z(220),TE(220)
        DO 100 I=1,NZ1
C        TE(I) = ((+25.*Z(I))/ZBOT) + 40.
           TE(I) = 40.0
100    CONTINUE
        K= NZ1 + 1
        L = K+1
        Z(K) = 0.0
        Z(L) = ZBOT/8.0
        TE(K) = 10.0
        TE(L) = 5.0
       CALL PLOT(1.0,9.0,-3)
       CALL AXIS(0.0,0.0,'TEMPERATURE',+11,8.0,0.0,10.0,5.0)
       CALL AXIS(0.0,0.0,'DEPTH CM',-8,8.0,270.0,0.0,Z(L))
       CALL LINE(TE,Z,NZ1,1,+1,1)
       CALL SYMBOL(2.0,-8.2,0.10,'TEMPERATURE PROFILE',0.0,+19)
       CALL PLOT(0.0,0.0,-999)
       RETURN
       END
       SUBROUTINE CORTEM(TE,Z,FACT,VIS)
```

```
C      DEBUG UNIT(9),INIT(SCALF,HYS1)
C
C DETERMINES THE TEMP. COEFFICIENT OF PRESSURE HEAD AND HYDRAULIC
C CONDUCTIVITY:
C
       COMMON AAA,JJJ,NZ1
       REAL TE(220),Z(220),FACT(220),VIS(220),T(220)
       DO 200 I=1,NZ1
          SUM=0.0
          T(I)=10.0*TE(I)
          IT=INT(T(I))
          IF(IT.LE.200) GO TO 150
             DO 100 J=210,IT
             E = J/10.
             SIG = 75.594 -0.1328*E-0.000537*E**2+2.2719E-06*E**3
             DSIG= -.1328 - 0.001074*E + 6.8157E-06*E**2
             GAM = (2.0/SIG)*DSIG*.1
             SUM = SUM + GAM
100           CONTINUE
          FACT(I)= 1.0 + SUM
          GO TO 200
150      IF(IT.EQ.200) GO TO 195
             DO 190 J=IT,200
             E = J/10.
             SIG = 75.594 -0.1328*E-0.000537*E**2+2.2719E-06*E**3
             DSIG= -.1328 - 0.001074*E + 6.8157E-06*E**2
             GAM = (2.0/SIG)*DSIG*.1
             SUM = SUM + GAM
190           CONTINUE
195      FACT(I)= 1.0 - SUM
200      CONTINUE
          DO 400 I=1,NZ1
          IF(TE(I).LT.20.) GO TO 300
          A = 1.3272*(20.-TE(I)) -0.001053*(TE(I)-20.)**2
          B = TE(I) + 105.
          C = 10**(A/B)
          VI = 0.01002*C
          IF(TE(I).EQ.20.0) VI=.01002
          GO TO 350
300      A = 998.333+8.1855*(TE(I)-20.)+0.00585*(TE(I)-20.)**2
          B = (1301./A) - 3.30233
          VI = 10**B
350      VIS(I) = 0.01002/VI
400      CONTINUE
          RETURN
          END
        SUBROUTINE PLO(Z,TH,TIME,O,PMAX,H)
C ** THETA WILL BE PLOTTED VERSUS DEPTH FOR THE TIMES SPECIFIED
C ** IN THE INPUT DATA FILE.
C      DEBUG UNIT(9),INIT(SCALF,HYS1)
       REAL Z(220),TH(220),TIME,O(10),H(220),PMAX,P,HY2(220,5)
       INTEGER HY1(220,3)
```

```
      COMMON AAA,JJJ,NZ1,ZBOT,ALP,UTOP,EMB,REMB,DELMO,TT,DELFLU,
     1 TEND,HY1,HY2,SCALF,THS
      DO 3 I=1,NZ1
      WRITE(6,2)HY1(I,1),HY1(I,2),HY1(I,3)
2     FORMAT(3I7)
3     CONTINUE
      DO 5 I=1,NZ1
      WRITE(6,4)HY2(I,1),HY2(I,2),HY2(I,3),HY2(I,4)
4     FORMAT(4E12.5)
5     CONTINUE
      K = NZ1 + 1
      L = K + 1
      Z(K) = 0.0
      Z(L) = ZBOT/8.0
      H(K)=0.0
      H(L)=-PMAX/4.
      P = PMAX/4.
      TH(K) = 0.0
      TH(L) = 0.05
      IF(TIME.GT.O(1)) GO TO 20
      CALL PLOT(10.0,9.0,-3)
      CALL AXIS(0.0,0.0,'VOLUMETRIC WATERCONTENT',+23,8.0,0.0,0.0,0.05)
      CALL AXIS(0.0,0.0,'DEPTH CM',-8,8.0,270.0,0.0,Z(L))
      CALL AXIS(0.0,-8.0,'PRESSURE HEAD',+13,8.0,180.0,0.0,P)
      CALL SYMBOL(1.0,-0.3,0.10,'SEC',0.0,+3)
      CALL SYMBOL(6.0,-6.0,0.10,'JAN HOPMANS',0.0,+11)
20     CALL LINE(TH,Z,NZ1,1,+1,JJJ)
      CALL LINE(H,Z,NZ1,1,+1,JJJ)
      CALL SYMBOL(0.3,AAA,0.10,JJJ,0.0,-1)
      CALL SYMBOL(0.5,AAA,0.10,'TIME',0.0,+4)
      CALL NUMBER(1.0,AAA,0.10,TIME,0.0,-1)
      CALL PLOT(0.0,0.0,+3)
      RETURN
      END
      SUBROUTINE PLOCO(C,G,Z)
C     DEBUG UNIT(9),INIT(SCALF,HYS1)
      REAL Z(100),C(100),G(100)
      COMMON AAA,JJJ,NZ1,ZBOT,ALP,UTOP,EMB,REMB,DELMO,TT,DELFLU,TEND
      NZ=NZ1 -1
      K=NZ1
      L = K + 1
      DO 10 J=1,NZ
      Z(J)=Z(J+1)
10    CONTINUE
      DO 20 I=1,NZ
      WRITE(6,25) Z(I),C(I),G(I)
20    CONTINUE
25    FORMAT( ' Z C G',2X,3E12.5)
      C(K)=0.0
      C(L)=0.0005
      G(K)=0.0
      G(L)=1.0
```

```
      Z(K)=0.0
      Z(L)=ZBOT/8.0
      CALL PLOT(2.0,9.0,-3)
      CALL AXIS(0.0,0.0,'DEPTH CM',-8,8.0,270.0,0.0,Z(L))
      CALL AXIS(0.0,0.0,'PRESSURE GRADIENT',17,8.0,0.0,0.0,1.00)
      CALL AXIS(0.0,.43,'CONDUCITIVITY',12,10.0,0.0,0.0,0.0005)
      CALL LINE(G,Z,NZ,1,+1,1)
      CALL LINE(C,Z,NZ,1,+1,2)
      CALL PLOT(0.0,0.0,-999)
      RETURN
      END
      SUBROUTINE MASSBA(TH,H2,Z,V0,V2,DT,DZ,V,F,CO,GR)
C
C CALCULATES MASS BALANCE OVER EACH TIME PERIOD.
C
C     DEBUG UNIT(9),INIT(SCALF,HYS1)
      REAL TH(220),H2(220),Z(220),V0(220),V2(220),DT(2),V(220),F(220)
      REAL CO(220),GR(220),P(220)
      INTEGER TT,JJJ
      COMMON AAA,JJJ,NZ1,ZBOT,ALP,UTOP,EMB,REMB,DELMO,TT,DELFLU,TEND
      NZ = NZ1 -1
      DO 10 I=1,NZ1
      P(I) = FTH(H2(I),F(I),I) - TH(I)
C     WRITE(6,349) I,TH(I),H2(I),P(I)
349   FORMAT(' I TH H DELTH',I4,3E15.5)
10    CONTINUE
      DELMO=0.0
      DO 20 I=1,NZ
         DELMO = DELMO - (P(I) + P(I+1))
399     FORMAT(' DELTH ',E12.5)
20    CONTINUE
C       WRITE(6,399) DELMO
      DELMO = DELMO *DZ / 2.0
C
      IF(ALP.EQ.1.0) V0(1)=UTOP
      DO 50 I=2,NZ1
      GR(I-1)=(H2(I)-H2(I-1))/DZ
      CO1=-FK(H2(I),V(I),F(I),I)
      CO2=-FK(H2(I-1),V(I-1),F(I-1),I-1)
      CON=(CO1+CO2)/2.
C     CON = -FK(.5*(H2(I)+H2(I-1)),.5*(V(I)+V(I-1)),.5*(F(I)+F(I-1)))
         V2(I) = CON * ((H2(I) - H2(I-1))/DZ) +CON
         CO(I-1) =-CON
50    CONTINUE
      V2(1)=V2(2)
      IF(ALP.EQ.1.0) V2(1)=UTOP
C     Y1=SIGN(1.0,V0(1))
C     Y2=SIGN(1.0,V2(1))
C     WRITE(6,51) Y1,Y2
51    FORMAT(50X,2F5.2)
C     IF(Y1.NE.Y2) V0(1)=V2(1)
      DELFLU = (-V2(1) - V0(1) + V2(NZ1) + V0(NZ1)) * DT(TT) / 2.0
```

```
      EMB = ABS(DELMO - DELFLU)
      REMB = ( EMB/ABS(DELFLU))*100
      RETURN
      END
      SUBROUTINE CONDI(Z,T,UT,UB,TIM,F,V,U)
C
C TEMPERATURE DISTRIBUTION AND TRANSIENT BOUNDARY CONDITIONS
C AS A FUNCTION OF TIME:
C
C     DEBUG UNIT(9),INIT(SCALF,HYS1)
      REAL Z(220),T(220),UT,UB,TIM,F(220),V(220),XX
      COMMON AAA,JJJ,NZ1,ZBOT,B3,B4,B5,A6,A7,J3,A8,A9,H1,H2,SC,TS,IFLAG
C     GO TO 600
      XX=100000.
      IF(TIM.GT.(7200.+XX)) GO TO 100
      IF(TIM.LE.7200.) GO TO 200
      IF(TIM.GT.XX) GO TO 300
100   A1 =(2*3.141*(TIM-7200))/86400
      UT=0.0000025+0.0000025*SIN(A1)
C     UT=0.000004
C     IFLAG=1
      GO TO 400
200   UT= -3./3600
C     IF(IFLAG.EQ.1) UT=0.0000025+SIN((2*3.14*(TIM-4000))/86400)
      GO TO 400
300   UT =(-1./3600)*(TIM-XX)/3600.
      IF(TIM.GT.(XX+3600)) UT=-2./3600
400   IF(TIM.GT.7200.) UB = U
      IF(TIM.LE.7200.) UB = -61.5
      WW = 7.272E-05
      TA = 30.0
      A0 = 15.0
      DD = 22.6
      DO 500 I=1,NZ1
          A2 = Z(I)/DD
          A3 = (EXP(A2))*SIN(WW*TIM + A2)
          T(I) = TA + A0*A3
C       T(I) = 20.0
500   CONTINUE
      GO TO 700
C600  UT = -13.69/3600.
600   UT=-30.0
      UB = -61.5
C     UT=5.0/3600
C     UB=-30.0
      DO 650 I=1,NZ1
      T(I)=20.0
650   CONTINUE
700   CALL CORTEM(T,Z,F,V)
      RETURN
      END
//*GO
```

```
//GO.FT01F001 DD DSNAME=AYL59JH.OUTDAT.CNTL,UNIT=DISK,
// DISP=(NEW,CATLG),SPACE=(TRK,(5,5),RLSE),LABEL=RETPD=3,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//*GO.FT09F001 DD DSN=AYL59JH.DUMMY.DATA,UNIT=DISK,
//* DISP=(NEW,CATLG),SPACE=(TRK,(5,5),RLSE),LABEL=RETPD=3,
//* DCB=(RECFM=FB,LRECL=132,BLKSIZE=1320)
//GO.FT03F001 DD DSN=AYL59JH.WAFLOW.LIB(DATIN),DISP=SHR,LABEL=(,,,IN)
//GO.SYSIN DD *
//*
//
```