INSTITUUT VOOR BODEMVRUCHTBAARHEID

NOTA 210

CALCULATION OF UPTAKE OF NUTRIENTS AND WATER BY A ROOT SYSTEM

door

P. DE WILLIGEN

1990

I N S T I T U U T   V O O R   B O D E M V R U C H T B A A R H E I D

NOTA 210

CALCULATION OF UPTAKE OF NUTRIENTS AND WATER BY A ROOT SYSTEM

door

P. DE WILLIGEN

1990

Instituut voor Bodemvruchtbaarheid, Oosterweg 92, Postbus 30003, 9750 RA Haren (Gr.)

CONTENTS

1. INTRODUCTION

Microscopic models on nutrient and water transport in the soil and on uptake by the roots deal with gradients in the immediate vicinity of a single root, i.e. with transport distances of a few millimeters or less. Most of such models only consider transport in a horizontal direction, transport being described in a fundamental way using Fick's and Darcy's laws.

In macroscopic models (like crop growth models), on the other hand, the scale is of the order of 10 cm or more and interest is focused on uptake behaviour of a root system rather than that of a single root. Usually transport in vertical direction is considered only, and fluxes of water are calculated approximately, except where interest is in dynamics of water flow per se.

Ideally, the description of uptake by a root system in a macroscopic model should be based on the description of the uptake of a microscopic model. More often than not, however, the macroscopic uptake models (implicitly) lump effects of physical and physiological properties of soil and roots into one or two parameters, of which it is often difficult to comprehend how they quantatatively relate to the microscopic properties they supposedly embrace.

The first goal of this report is to discuss how results of uptake models on a microscopic scale can lead to a description of uptake on a macroscopic scale consistent with the former. A second goal is to present a documentation on the computer programs pertaining to the models discussed.

2. MICROSCOPIC MODEL OF UPTAKE OF NUTRIENTS AND WATER

2.1. Basic assumptions

For the microscopic model we start by considering a vertical cylindrical root. The root is situated within a cylinder of soil. Over the outer boundary of this cylinder no transport will take place. Such a root can be thought of as belonging to a set of uniformly distributed roots all equally active and of the same length.

The uptake rate is assumed to be constant and independent of the nutrient concentration (water content or matric potential in case of water uptake) as long as the latter exceeds a certain limiting value. This approach is different from that of other workers in this field (Nye and Tinker, 1977; Barber, 1984) who put most emphasis on the uptake-determining role of the concentration.

Whenever the concentration (water content, matric potential) at the root surface reaches its limiting value the uptake rate equals the rate of arrival of the nutrient (water) at the root surface, i.e. the concentration (water content) there will be maintained at the limiting value. In case of the important nutrients N and K the limiting value can be safely set at zero.

2.2 Nutrients

Analytical solutions

The system described in the foregoing paragraph can be formulated mathematically by a partial differential equation, describing transport by massflow and diffusion in a hollow cylinder, together with initial and boundary conditions. The boundary condition at the inner cylinder (the root) is either that of constant uptake or of zero-concentration.

De Willigen and van Noordwijk (1987) presented an analytical solution when uptake is constant and the initial concentration is uniform. It reads in dimensionless units (Table 1 explains the symbols):

TABLE 1. List of symbols.

Symbol	Name	Dimension	Dimensionless symbol
R_0	root radius	cm	–
D	diffusion coefficient	cm^2/day	–
T	time	day	$t = DT/R_0$
T_c	period of	day	t_c
$T_{c,max}$	maximum T_c	day	$t_{c,max}$
R	radial coordinate	cm	$r = R/R_0$
R_1	radius soil cylinder	cm	$\rho = R_1/R_0$
H	root length	cm	$\eta = H/R_0$
S	available amount of nutrient	mg/cm^3	$s = S/S_i$
S_i	initial value of S	mg/cm^3	
A	uptake rate	$mg/(cm^2.day)$	$\omega = -\rho^2\theta\beta/(2\phi\eta)$
ϕ	supply/demand parameter	–	$\phi = DS_i/(AR_0)$
C_i	initial nutrient concentration	mg/ml	–
C	nutrient concentration	mg/ml	$c = C/C_i$
θ_i	initial watercontent	ml/cm^3	–
θ	water content	ml/cm^3	–
K_a	adsorption constant	ml/cm^3	–
$\theta\beta$	buffer capacity	ml/cm^3	$\beta=(K_a+\theta)/\theta$
U_1	flow of water over the root surface	cm/day	–
U_2	flow of water from bulk soil to root surface		–
E	transpiration rate	cm/day	–
V	flux of water	cm/day	$2\nu = RV/D$
P	pressure head	cm	
D_w	diffusivity	cm^2/day	
K	conductivity	cm/day	
Φ	matric flux potential	cm^2/day	
Φ_{rs}	matric flux potential at root surface	cm^2/day	

$$c = \frac{(\rho^2-1)(\nu+1)r^{(2\nu)}}{\rho^{2\nu+2}-1} + \tag{1a}$$

$$\omega\left[\frac{2(\nu+1)r^{2\nu}}{\rho^{2\nu+2}-1}\frac{t}{\theta\beta} + \frac{r^{2\nu}(r^2-\rho^2)}{\rho^{2\nu+2}-1} + \frac{\rho^2(\rho^{2\nu}-r^{2\nu})}{2\nu(\rho^{2\nu+2}-1)} + \frac{\rho^2(\rho^{2\nu}-1)r^{2\nu}(\nu+1)}{2\nu(\rho^{2\nu+2}-1)^2} + \frac{r^{2\nu}(\nu+1)(1-\rho^{2\nu+4})}{(2\nu+4)(\rho^{2\nu+2}-1)^2}\right] + \tag{1b}$$

$$(\omega-2\nu)r^{\nu}\pi\sum_{n=1}^{\infty}\frac{J_{\nu+1}(\alpha_n)J_{\nu+1}(\rho\alpha_n)}{\alpha_n}F_{\nu}(r,\alpha_n)e^{\frac{-\alpha_n^2 t}{\theta\beta}} + \tag{1c}$$

$$\frac{2\nu r^{\nu}\pi}{\rho^{\nu+1}}\sum_{n=1}^{\infty}\frac{J_{\nu+1}^2(\alpha_n)}{\alpha_n}F_{\nu}(r,\alpha_n)e^{\frac{-\alpha_n^2 t}{\theta\beta}} \tag{1d}$$

with:

$$F_{\nu}(r,\alpha_n) = \frac{Y_{\nu+1}(\rho\alpha_n)J_{\nu}(r\alpha_n) - Y_{\nu}(r\alpha_n)J_{\nu+1}(\rho\alpha_n)}{J_{\nu+1}^2(\alpha_n) - J_{\nu+1}^2(\rho\alpha_n)}$$

and α_n is the n-th root of:

$$Y_{\nu+1}(\alpha)J_{\nu+1}(\rho\alpha) - J_{\nu+1}(\alpha)Y_{\nu+1}(\rho\alpha) = 0$$

Where $J_{\nu}(x)$, $Y_{\nu}(x)$ denote the Bessel function of the first kind, order ν and argument x and the modified Bessel function of the first kind order ν and argument x respectively. Eventually the series part (1c-d) will vanish and a steady-rate situation, where concentration is a linear function of time, will develop given by (1a-1b). The uptake potential of the system can be characterized by a characteristic time constant t_u, called the period of unconstrained uptake, denoting the period during which the concentration at the root surface exceeds zero. If the concentration at the root surface reaches its limiting zero-value in the steady-rate situation, at time $t = t_u$, the concentration profile can be given by:

$$c = \omega\left[\frac{r^{2\nu+2}}{2(\rho^{2\nu+2}-1)} - \frac{r^{2\nu}(\nu+\rho^{2\nu+2})}{2\nu(\rho^{2\nu+2}-1)} + \frac{\rho^{2\nu+2}}{2\nu(\rho^{2\nu+2}-1)}\right] \tag{2}$$

If after $t = t_u$ the concentration at the root surface is constant (zero), i.e. the root behaves as a zero-sink, the concentration distribution for $t > t_u$ is given by (De Willigen and Van Noordwijk in prep.):

$$c = \frac{-2\pi\omega(\nu+1)\}}{\rho^{2\nu+2}-1} \sum_{n=0}^{\infty} \frac{J_\nu(\alpha_n)J_{\nu+1}(\alpha_n)}{J_\nu^2(\alpha_n) - J_{\nu+1}^2(\rho\alpha_n)} \frac{r^\nu C_\nu(r\alpha_n)}{\alpha_n^2} e^{\frac{-\alpha_n^2(t-t_u)}{\theta\beta}} , \tag{3}$$

where

$$C_\nu(r\alpha) = Y_\nu(r\alpha)J_{\nu+1}(\rho\alpha) - J_\nu(r\alpha)Y_{\nu+1}(\rho\alpha)$$

and α_n now is the n-th root of $C_\nu(\alpha) = 0$.
From (3) the uptake rate can be derived as the gradient of the concentration c at the root surface (where r=1):

$$\omega_t = \left(\frac{\partial c}{\partial r}\right)_{r=1} = \frac{-4(\nu+1)\omega}{\rho^{2\nu+2}-1} \sum_{n=0}^{\infty} \frac{J_{\nu+1}^2(\rho\alpha)}{\alpha^2\left(J_\nu^2(\alpha) - J_{\nu+1}^2(\rho\alpha)\right)} e^{\frac{-\alpha_n^2(t-t_u)}{\theta\beta}} \tag{4}$$

From (4) the time-integrated uptake after the period of unconstrained uptake is found as:

$$\int_{t_u}^{t} \omega_t dt = \frac{-4(\nu+1)\omega\theta\beta}{\rho^{2\nu+2}-1} \sum_{n=0}^{\infty} \frac{J_{\nu+1}^2(\rho\alpha)}{\alpha^4\left(J_\nu^2(\alpha) - J_{\nu+1}^2(\rho\alpha)\right)} \{1 - e^{\frac{-\alpha_n^2(t-t_u)}{\theta\beta}}\} \tag{5}$$

It now appears (De Willigen and Van Noordwijk 1987) that (4) can be approximated very well by a much simpler equation viz. a steady-rate equation like (2), but with time-dependent uptake rate ω_t, which is given by:

$$\omega_t = \omega\, e^{-\frac{t}{\theta\beta G(\rho,\nu)}} , \tag{6}$$

A consequence of the assumption of the concentration being given by (2) with ω_t instead of ω is that the former is proportional to the average concentration:

$$\omega_t = - \frac{(\rho^2-1)\overline{c}}{2G(\rho,\nu)} , \tag{7}$$

where the function $G(\rho,\nu)$ is given by:

$$G(\rho,\nu) = \frac{1}{2(\nu+1)}\left(\frac{1-\rho^2}{2} + \frac{\rho^2(\rho^{2\nu}-1)}{2\nu} + \frac{\rho^2(\rho^{2\nu}-1)(\nu+1)}{2\nu(\rho^{2\nu+2}-1)} + \frac{(1-\rho^{2\nu+4})(\nu+1)}{(2\nu+4)(\rho^{2\nu+2}-1)}\right)$$

and c denotes the average concentration in the soil solution.
The above solutions (i.e. (3), (4), (5) and the approximation (6)) pertain to the situation where a steady-rate concentration profile (2) has developed before the concentration at the root surface reaches zero. When the concentration at the root surface drops to zero before the steady-rate situation has been reached -thus before the terms (1c) and (1d) can be neglected - it is assumed that the root behaves as a zero-sink from the beginning. The exact solution for the concentration is now given by (De Willigen and Van Noordwijk, unpublished):

$$c = -\pi \sum_{n=0}^{\infty} \frac{2\nu J_\nu^2(\alpha_n) - \alpha\rho^{\nu+1} J_\nu(\alpha_n) J_{\nu+1}(\alpha_n)}{J_\nu^2(\alpha_n) - J_{\nu+1}^2(\rho\alpha_n)} \; \frac{r^\nu C_\nu(r\alpha_n)}{\rho^{\nu+1}\alpha_n} \; e^{\frac{-\alpha_n^2 t}{\theta\beta}} , \qquad (8)$$

and that for uptake by :

$$\omega_t = \left(\frac{\partial c}{\partial r}\right)_{r=1} = -2 \sum_{n=0}^{\infty} \frac{2\nu J_\nu(\alpha) J_{\nu+1}(\rho\alpha) - \alpha\rho^{\nu+1} J_{\nu+1}^2(\rho\alpha)}{\alpha\left(J_\nu^2(\alpha) - J_{\nu+1}^2(\rho\alpha)\right)} \; e^{\frac{-\alpha_n^2 t}{\theta\beta}} \qquad (9)$$

whereas the time-integrated uptake now is given by:

$$\int_0^t \omega_t \, dt = -2\theta\beta \sum_{n=0}^{\infty} \frac{2\nu J_\nu(\alpha) J_{\nu+1}(\rho\alpha) - \alpha\rho^{\nu+1} J_{\nu+1}^2(\rho\alpha)}{\alpha^2\left(J_\nu^2(\alpha) - J_{\nu+1}^2(\rho\alpha)\right)} \left\{1 - e^{\frac{-\alpha_n^2 t}{\theta\beta}}\right\} \qquad (10)$$

Here it also appeared that the the complicated expression for uptake (9) can be very well approximated by an equation like (7).

Figure 1 shows the cumulative uptake in the zero-sink situation for the exact solutions and the approximation. Details about the calculation procedure are found in Appendix I. The approximation is shown to be quite good, so it will be used in formulating uptake in the macroscopic model to be discussed in section 3.1.

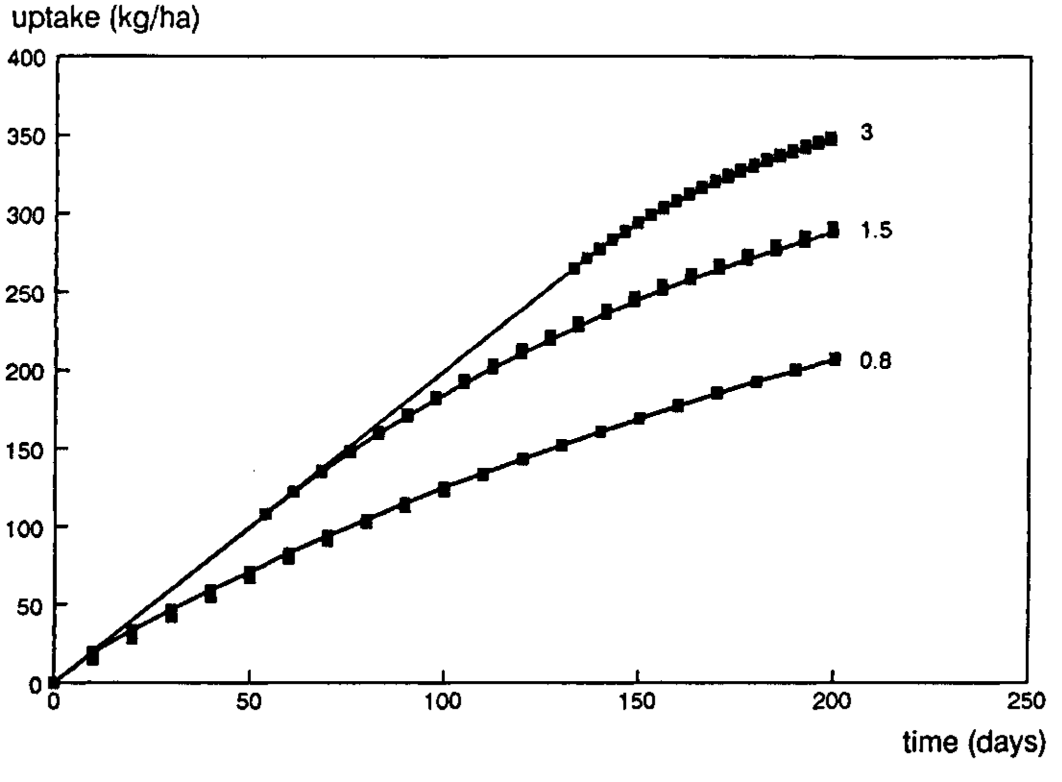


Figure 1. Cumulative uptake calculated with the exact solution (6) and the steady-rate approximation. Parameters: available amount 400 kg/ha, required uptake rate 2 kg/(ha.day), root length 20 cm, root radius 0.02 cm, transpiration 0.5 cm/day, diffusion coefficient 0.028 cm^2/day, adsorption constant 20 ml/cm^3. The lines give the exact solution; the points the steady-rate approximation. The figures at the curves denote the root length density in cm/cm^3.

2.3. Water

Mathematical formulation

The partial differential equation describing transport of water in the soil (Richards equation) is strongly nonlinear, which makes the finding of analytical solutions generally very difficult. We had to resort to numerical techniques to obtain a solution (see Appendix II).

As shown earlier (De Willigen and Van Noordwijk, 1987) the numerical solution can be very well approximated by a steady-rate approximation similar to that given in (2), be it that it should be written in terms of the matric flux potential, which is defined as:

$$\Phi = \int_{\theta}^{\theta_{sat}} D_w d\theta = \int_{P}^{0} K dP , \qquad (11)$$

and that the limit for $\nu \longrightarrow 0$ of (2) should be taken. The steady-rate solution in terms of the matric flux potential reads:

$$\Phi = \Phi_{rs} + \frac{\rho^2}{2\eta} E R_0 \left[\frac{r^2-1}{2(\rho^2-1)} - \frac{\rho^2 \ln r}{\rho^2-1} \right] \tag{12}$$

where Φ_{rs} is the matric flux potential at the root surface. The flow of water towards the root is then given as:

$$U_2 = \frac{H}{R_1^2} (\Phi_{rs} - \Phi) \frac{\rho^2-1}{G(\rho,0)} \tag{13}$$

where $G(\rho,0)$ is the function $G(\rho,\nu)$ given earlier for $\nu \longrightarrow 0$.

Figure 2 shows the time course of water uptake in case of constant water uptake until P_{rs}, the pressure head at the root surface, equals 0.5 MPa, whereas thereafter P_{rs} is kept constant at 0.5 MPa, calculated with the numerical model and with the steady-rate approximation (13). The approximation is shown to be quite satisfactory.

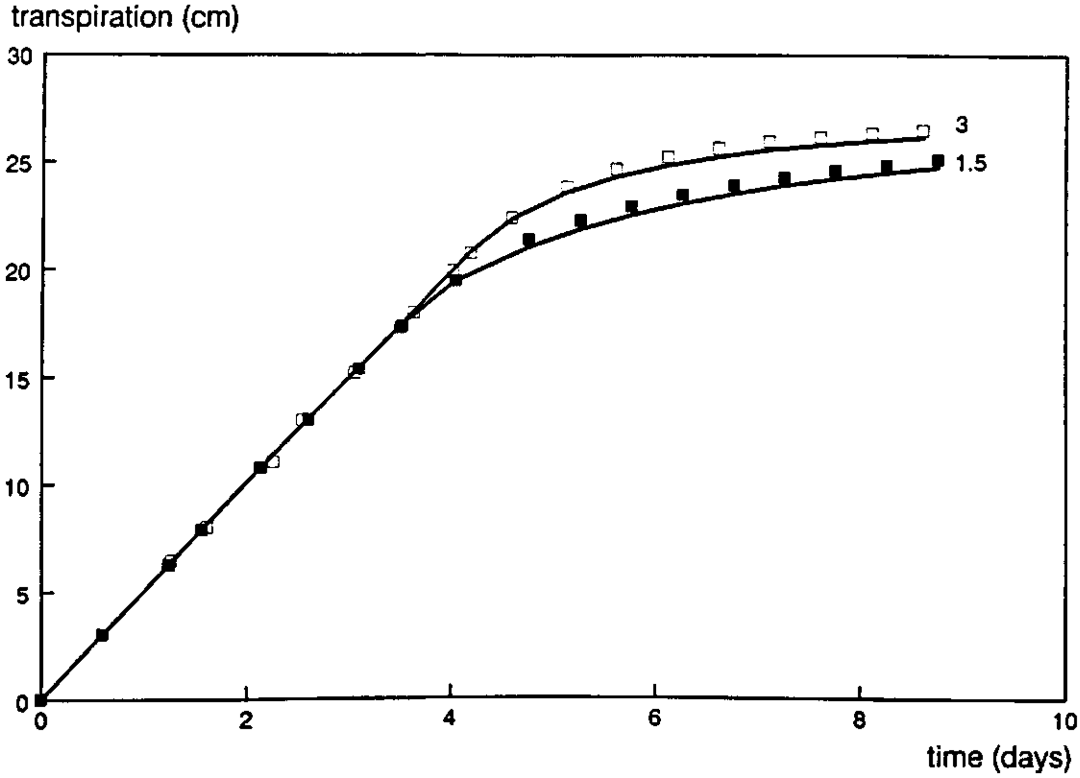


Figure 2. Cumulative transpiration calculated with the numerical model (lines) and with the steady-rate approximation (points). Transpiration was set constant at 0.5 cm/day, until the pressure head at the root surface reaches 5000 cm. Thereafter the pressure head is kept constant at 5000 cm. The figures at the curves denote the root length density in cm/cm^3.

3. DESCRIPTION OF UPTAKE IN A MACROSCOPIC MODEL

3.1. Nutrients

The findings discussed in 2.2 can be used to describe uptake in a macroscopic model, i.e. a model which deals with average values of concentration and root density in horizontal layers. We first consider a single layer with uniformly distributed roots. The potential uptake rate from this layer is given by (7), i.e. the uptake rate in the zero-sink situation. As long as this potential uptake rate exceeds the required uptake rate the actual uptake rate equals the required uptake rate. When the potential uptake rate is lower than the required uptake rate, actual uptake rate equals potential uptake rate. The actual uptake rate is thus the minimum of the required and the potential uptake rate. The total uptake then simply is the sum of the uptake of the individual roots.

When different horizontal layers in the root zone are distinguished, each with its own root density and nutrient concentration the situation is more complicated. It is conceivable, as indeed is often found, that roots in favourable position can compensate for roots in less favourable position (De Jager, 1985). It is assumed therefore that the uptake rate of roots in a certain layer depends on the uptake potential of roots in other layers.

Uptake now is calculated in an iterative way. First (step 1) the nutrient demand is divided by the total root length to obtain the required uptake per unit root length. Multiplying this by the root length in a given layer yields the required uptake from that layer. If the potential uptake rate exceeds the required uptake rate, uptake from this layer equals the required uptake. If the potential uptake rate is less than the required uptake rate, uptake from this layer equals the potential uptake rate. For convenience, those layers where actual uptake equals potential uptake will be indicated as layers of category 1.

The total uptake by the root system is the sum of the uptake rates of the individual layers. If the uptake in each layer can proceed at the required rate, total uptake equals nitrogen demand and no iteration is required. If total uptake is less than nitrogen demand, it is checked whether uptake from those layers where the concentration was sufficiently high to meet the original demand (for that particular layer) can be raised to increase total uptake, possibly enough to meet the total demand.

This is achieved as follows. In step 2, first the difference between demand and total uptake, as calculated in step 1, is divided by the total root length of those layers (category 2) that were able to satisfy the required uptake rate of step 1. This yields an additional uptake rate. The required uptake rate for layers of category 2, in step 2, now equals the required uptake rate of step 1, augmented with the additional uptake rate. It now is examined if the layer can satisfy the required uptake, i.e. if its potential uptake rate still exceeds the new value of the required uptake rate. If not, uptake rate from this layer equals potential uptake rate. If all layers of category 2 can satisfy the required uptake of step 2, total uptake equals demand and the iteration ends. If none of the layers of category 2 can satisfy the required uptake of step 2, i.e. if in all layers of category 1 and 2 zero-sink uptake occurs, the iteration also ends. If only a part of the layers of category 2 can satisfy the required uptake of step 2, iteration proceeds to step 3, etc. Appendix III describes the computerprogram of this procedure in detail.

This calculation procedure implies that roots growing under favourable conditions will compensate as much as possible for roots growing under less favourable conditions. It is thus assumed that information about the necessary behavior, as far as uptake is concerned, is instantaneously available throughout the complete root system.

The procedure described above can also be applied when roots within a horizontal layer are not distributed uniformly. The layer then is divided over a number of compartments within which root distribution can be assumed uniform.

3.2 Water

As in 3.1 we start with considering a single layer of soil containing vertical and uniformly distributed roots. The flow of water over the root surface is supposed to be linearly related to the difference between the plant water potential and the pressure head of the soil water at the root surface (De Willigen and Van Noordwijk, 1987):

$$F_w = k_1 (P_{rs}-P_p) + k_2 \tag{14}$$

where F_w is the flux of water in $cm^3/(cm\ root.day)$, P_{rs} is the pressure head in the soil at the root surface in cm, P_p is the plant water potential in cm, and k_1 in cm/day and k_2 in $cm^3/(cm\ root.day)$ are coefficients related to the root conductivity.

If the thickness of the soil layer is Δx cm and the root length density L_{rv} cm/cm^3 , the uptake U_1 in cm^3 water per cm^2 soil surface per day is given by:

$$U_1 = L_{rv}\ \Delta x\ F_w = q(P_{rs} - P_p) + v \tag{15}$$

where

$$q = L_{rv}\,\Delta x\,k_1, \text{ and } v = L_{rv}\,\Delta x\,k_2$$

We assume that no water can accumulate at the root surface, so that U_1 equals the supply of water at the root surface. The flow from the bulk soil in the layer with matric flux potential Φ towards the rootsurface where the matric flux potential is Φ_{rs} is given by (cf (13)):

$$U_2 = s(\Phi_{rs} - \Phi) \tag{16}$$

where

$$s = \frac{\Delta x}{R_1^2}\,\frac{(\rho^2 - 1)}{G(\rho,0)}$$

whereas there exists a functional relationship between Φ and P (see definition of Φ in (11)).

If the required uptake is given as E_{act} (the transpiration rate in cm/day) the right hand side of (16) should equal E_{act}, and the resulting equation can simply be solved for Φ_{rs}. Using the inverse of (11) then P_{rs} is calculated. Finally because:

$$U_1 = U_2 = E_{act}$$

the plant water potential can be computed from (15) as:

$$P_p = P_{rs} - (E_{act} - v)/q \tag{17}$$

The situation is in fact more complicated because the actual transpiration rate is a function of the plant water potential:

$$E_{act} = f(P_p)\,E_{pot} \tag{18}$$

where $f(P_p)$ is a factor by which the potential transpiration E_{pot} has to be reduced. As $f(P_p)$ is a non-linear function of P_p, E_{act} and P_{rs} have to be found by iteration.

When the root system is distributed over n different layers, but roots can be assumed to be distributed uniformly within a layer, for each layer equations like (15) and (16) can be formulated. Equating these equations one finds for the i-th layer:

$$q_i(P_{rs,i} - P_p) + v_i = s_i(\Phi_{rs,i} - \Phi_i) \tag{19}$$

It is assumed that the plant water potential P_p is identical everywhere in the root system. In total then one obtains n equations of the form of (19), and n+1 unknowns i.e. $P_{rs,1} - P_{rs,n}$ (where again (11) can be used to calculate Φ from P or vice-versa) and P_p. When the uptake rate E_{act} is known an additional equation states that the sum of the flows over the rootsurface in each layer from the root zone equals E_{act}:

$$\sum_{i=1}^{n} q_i(P_{rs,i} - P_p) + v_i = E_{act} \tag{20}$$

When one takes into account the relation between plant water potential and uptake rate equation (18) has to be used. Because of the non-linearity of (11) and (18) the solution has to be found by iteration. Appendix IV gives the details.

4. PROGRAMS

Calculations were performed with the computerprograms shown in appendices I-IV. All programs are written in Fortran. The programs consist of a number of subprograms (Fortran subroutines and functions), each dealing with a specific part of the calculations. The programs are organized and documented according to recommendations of Van Kraalingen and Rappoldt (1989).

5. EXAMPLES

5.1 Water uptake

As an example results of Sharp and Davies (1985) will be used. They performed an experiment on water uptake and root growth of maize plants. Individual plants were grown in soil columns of 80 cm height. After a preliminary growth period of two weeks the plants were subjected to two treatments. Half of the plants were watered daily, where care was taken that the soil water was replenished up to field capacity, the other half did not get any water at all for the 18 day experimental period. Every three days some columns were harvested and soil water content, root length distribution, total dry matter and its distribution were measured, as well as, prior to harvesting, the leaf water potential. Soil water depletion rates were calculated from the changes in soil water content between successive sampling days.

No data on diffusivity as a function of water content are given, and the information about the moisture retention curve is rather incomplete. Nevertheless, the data can be used to illustrate the performance of the model. To this end the model was run with data on hydraulic properties of

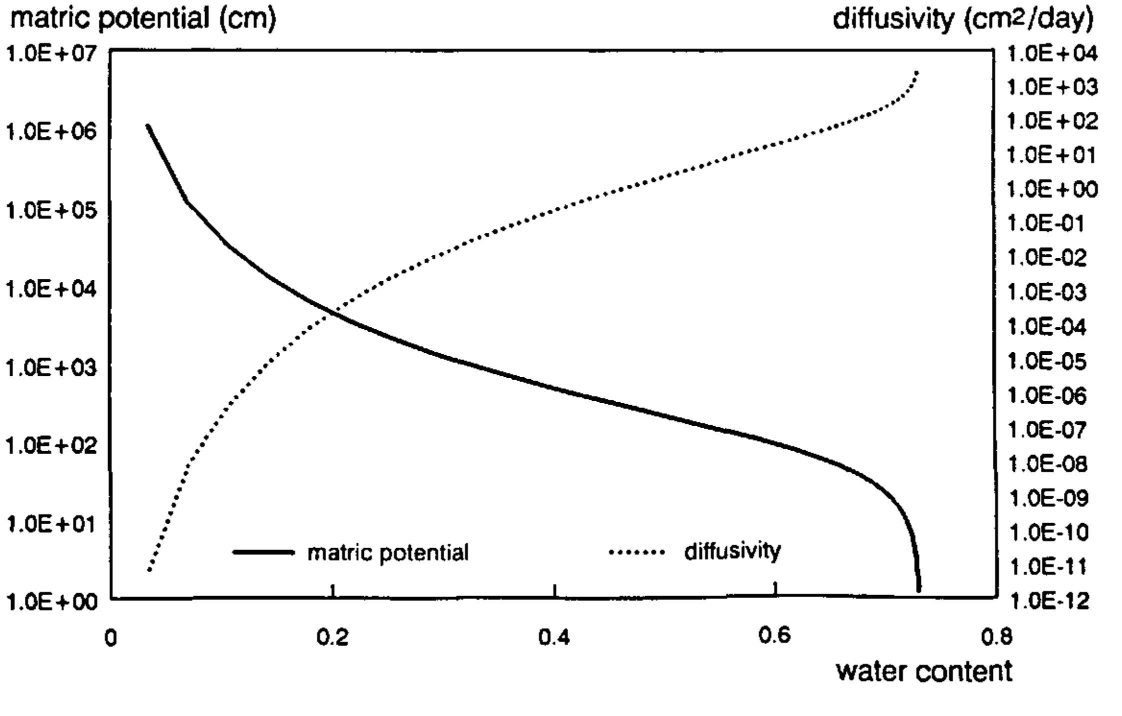


Figure 3. Relation between matric potential, diffusivity and water content for a peat soil (no B16 in the Staring Series (Wösten, 1987).

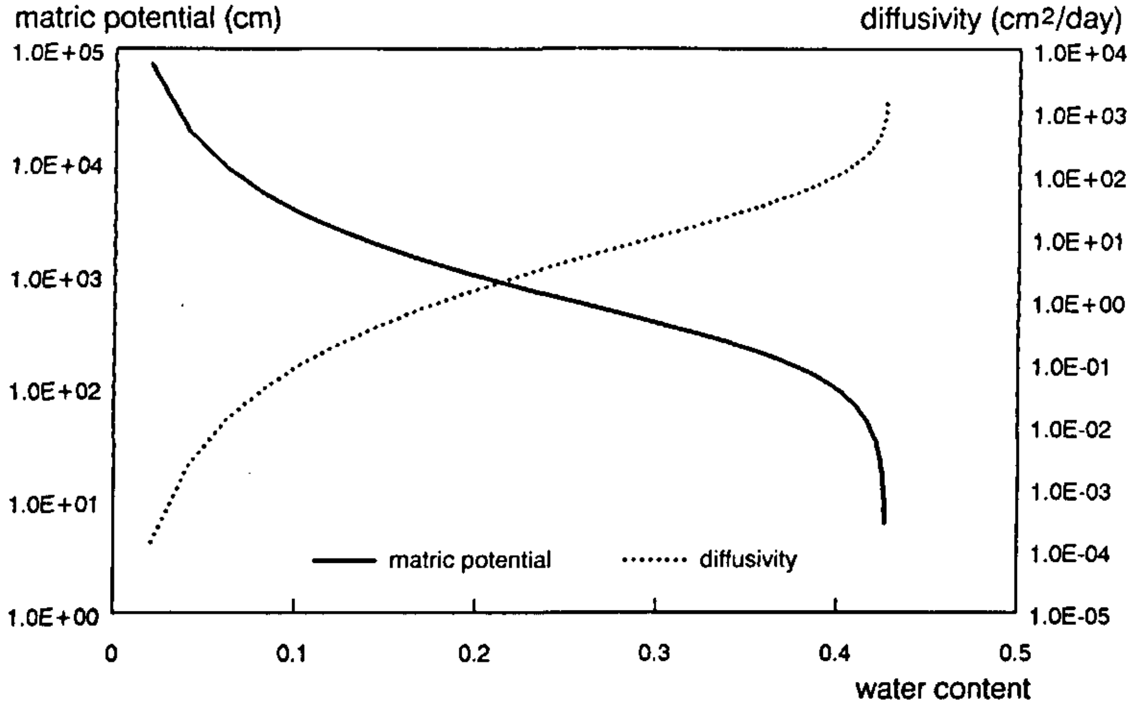


Figure 4. Relation between matric potential, diffusivity and water content for a sandy loam (Lovl2c data from Boekhold, 1987).

two soils: one a sandy loam (Lovl2c; Boekhold, 1987) the other a peaty soil (B16; Wösten, 1987). Figures 3 and 4 depict the relation between water content, pressure head and diffusivity of these two soils.

Figures 5 and 6 give the results for day 15 for the two treatments. In the treatment where the soil was kept at field capacity (figure 5) distribution of uptake with depth was similar to the root density distribution. In the calculations it was assumed that soil water was in equilibrium with a groundwater table at a depth of 80 cm, the length of the soil tube. According to the calculations at every depth in the transport chain: bulk soil ---> root surface ---> root, the highest resistance was experienced by the transport over the root wall. The overall resistance was therefore lowest where root density was highest, which results in uptake being more or less proportional to root density. Because of the dominance of the root resistance the results for the two soils were practical identical, differences in calculated uptake being less than 0.01%.

The results for the unwatered treatment are given in figure 6. The pressure head at every depth was estimated from the reported water content. Now the soil resistance is, except for the deepest layer, higher than the root resistance, so distribution of uptake is quite different from root density distribution. Moreover, the results for the two soils now clearly

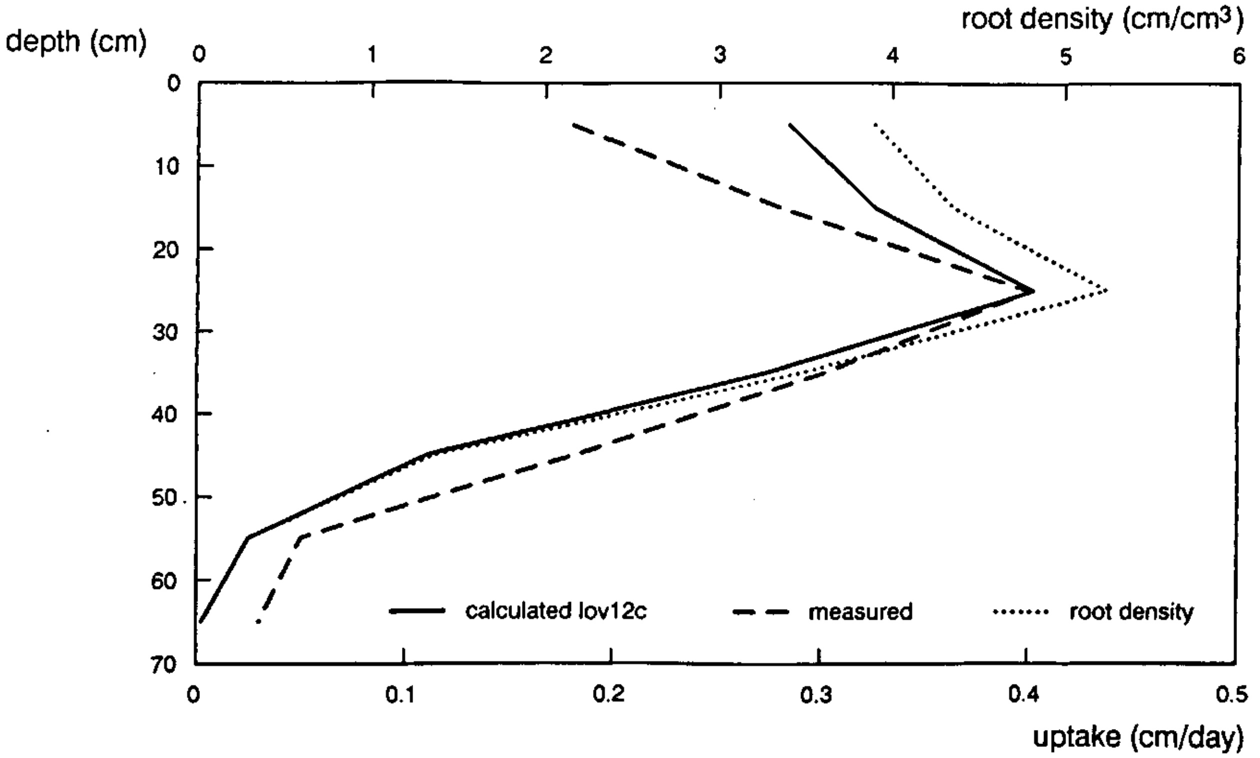


Figure 5. Measured and calculated depth-distribution of water uptake and root-density of a maize-plant in a soil kept at field capacity. Data from Sharp and Davies (1985). For the calculations the hydraulic properties of Lov12c (figure 4) have been used.

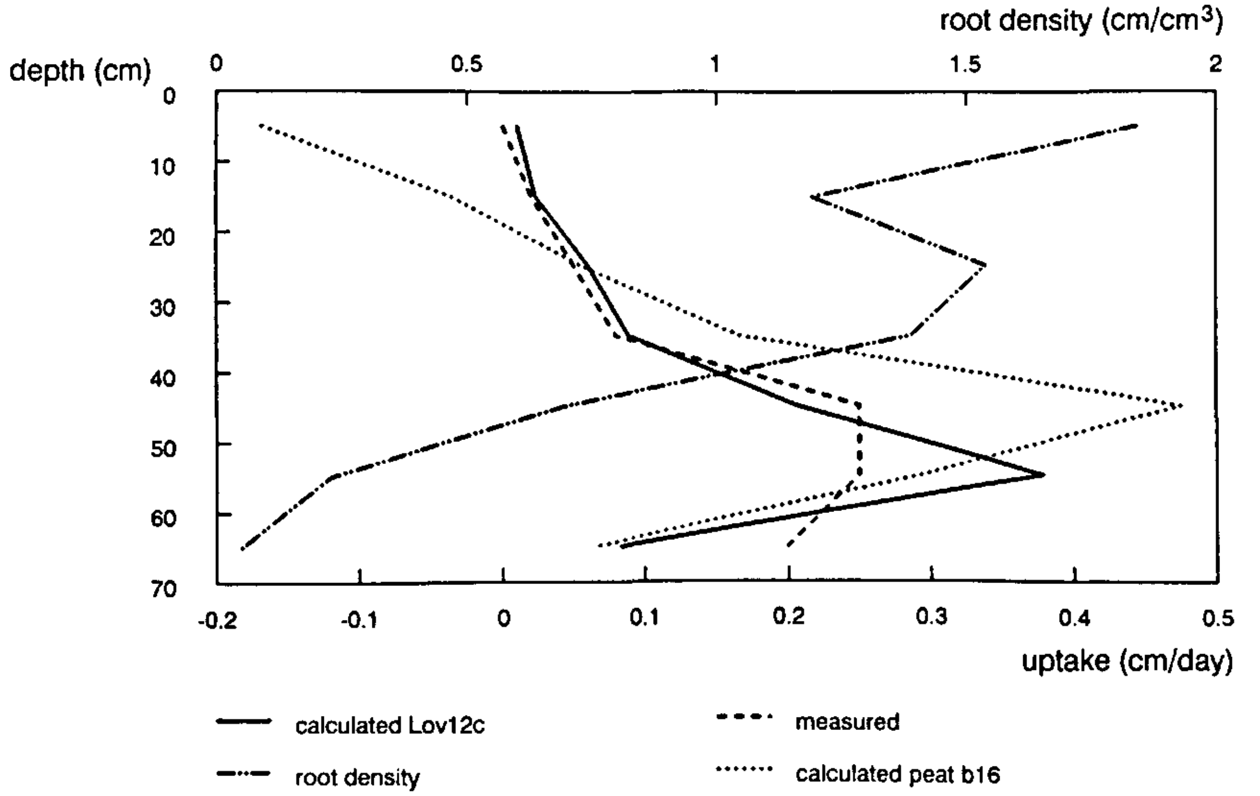


Figure 6. As figure 5, but in a soil which has not been watered during 15 days. For the calculations the hydraulic properties of Lov12c (figure 4) and B16 (figure 3) have been used.

differ, those for Lovl2c being more or less similar to the measured results. The results of the peat B16 differ considerably from the measurements. On this soil in the upper two layers negative water uptake occurs according to the calculations, i.e. water flows out of the root to the soil. This may be an artefact due to the assumption of uniform plant water potential, but on the other hand such a phenomenon is often established experimentally (e.g. Baker and Van Bavel, 1986).

5.2. Nutrient uptake

To illustrate the possibilities of the nutrient uptake model, the water uptake pattern and water content distribution as found by Sharp and Davies (1985) discussed in 5.1 were used.

Four situations were considered: the two treatments of water application mentioned in section 5.1 with the corresponding root length distributions, and two different distributions of mineral nitrogen in the soil. In one of the chosen distributions every soil layer contains 1 kg/ha, except the upper 10 cm where mineral nitrogen amounts to 20 kg/ha, in the other this amount is found in the layer 40-50 cm. The required uptake rate of the maize plants was estimated from the transpiration rate assuming a transpiration coefficient of 350 kg/kg and an assumed nitrogen content of 2.5%. For the adequately watered plants the required uptake calculated in this fashion amounted to 8.8 kg/(ha.day), for the unwatered plants to 6 kg/(ha.day).

The effective diffusion coefficient of ions in soil is strongly influenced by the water content. It can be calculated as:

$$D = D_0 f_l \theta \qquad (21)$$

where D_0 is the diffusion coefficient of the ion in free water, and f_l an impedance factor. The latter is a function of θ as well. Figure 7 gives the diffusion coeffient as a function of the water content using the relation between impedance factor and water content as established by Barraclough and Tinker (1981) for sandy and sandy loam soils. This relation has been used in the model calculations.

Figures 8 and 9 show the results. For the adequately watered plants the distribution of nitrogen has no effect, whereas in the dry situation concentration of nitrogen in deeper moister layers is much more favourable, the realized uptake rates being 1.5 and 4.5 kg/(ha.day) respectively.

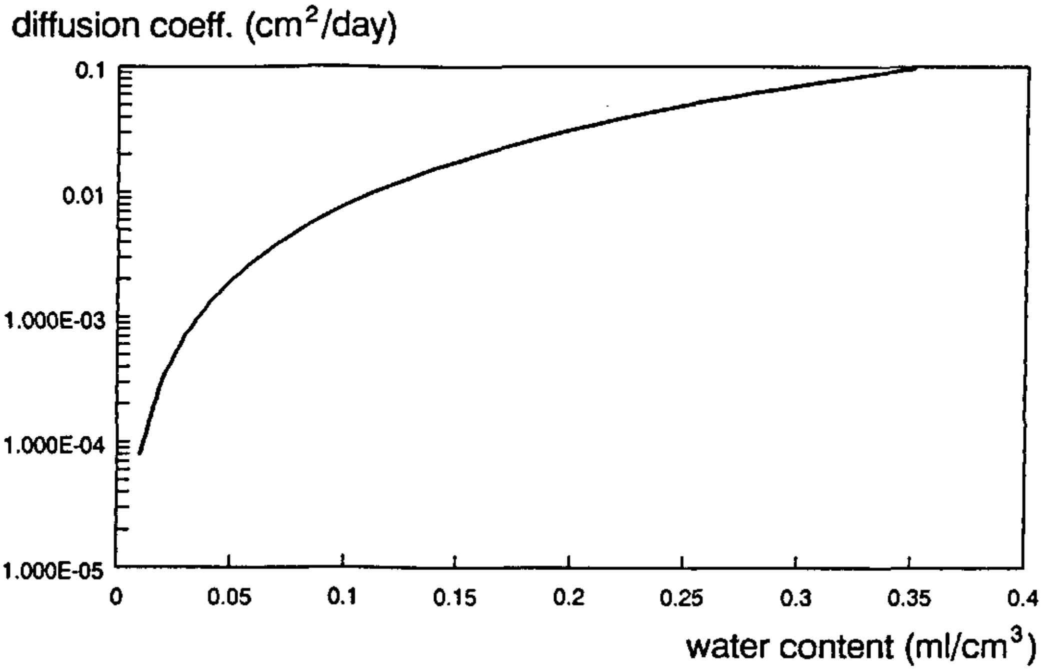


Figure 7. Diffusion coefficient of a nutrient, the diffusion constant of which in water is 0.5 cm^2/day, as a function of water content for sandy soils. After Barraclough and Tinker (1981).

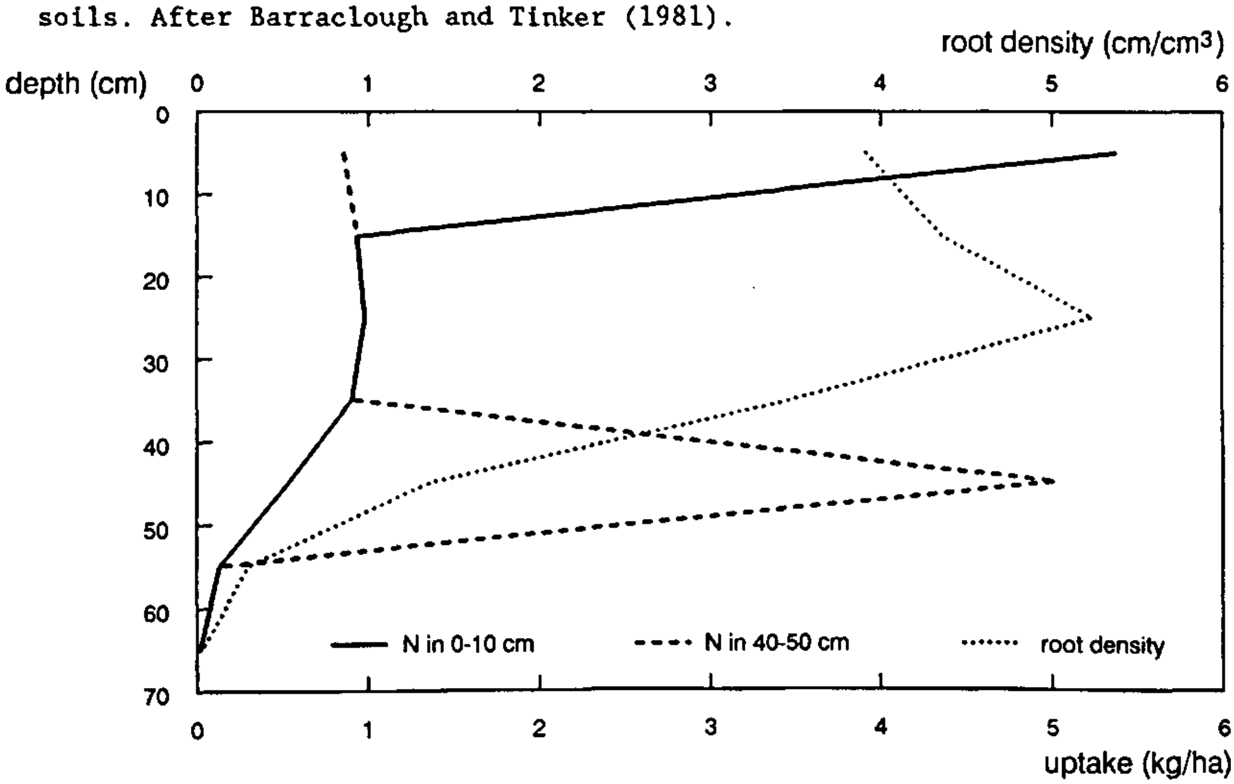


Figure 8. Distribution of nitrate-N uptake and root length density with depth as calculated with the model UPTA in a soil at field capacity. Water-uptake distribution as in figure 5 (measurements). Diffusion coefficient calculated from watercontent as shown in figure 7. All soil layers contain 1 kg of nitrate-N per ha, except for the layer 0-10 cm (continuous line) or 40-50 cm (interrupted line). These layers contain 20 kg nitrate-N per ha. Nitrogen demand: 8.8 kg/(ha.day).

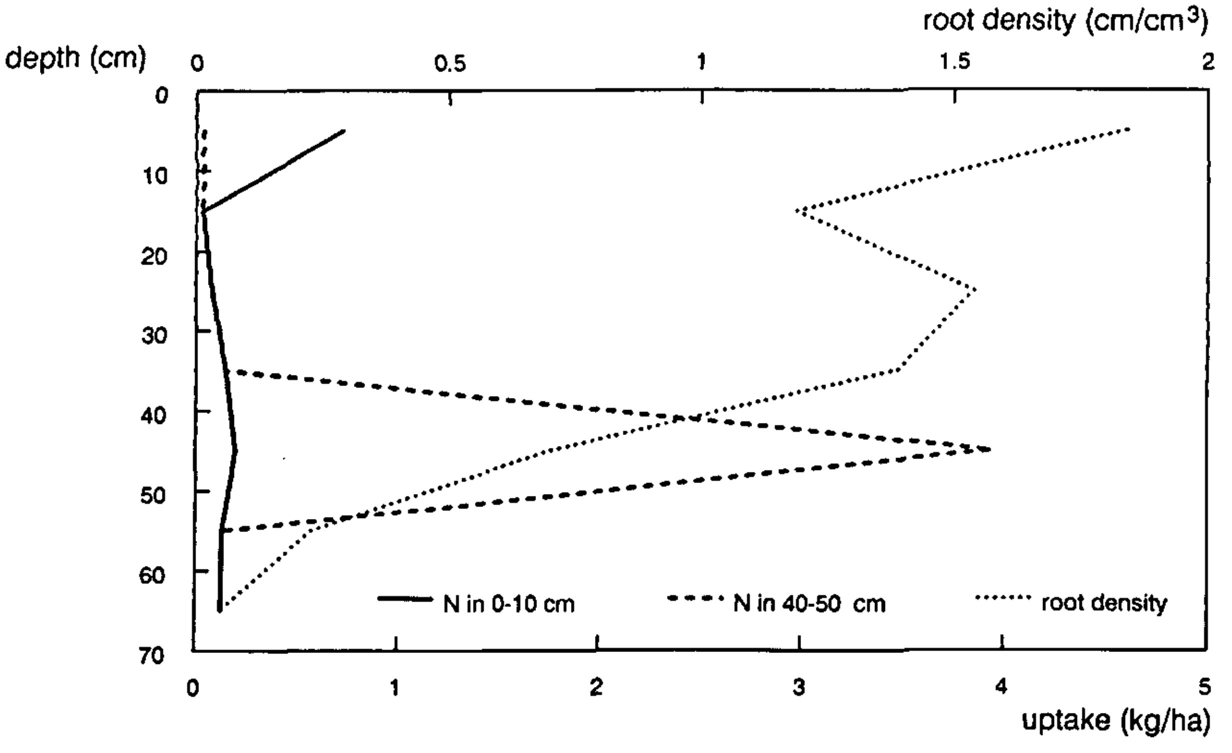


Figure 9. As fig 8 but in a soil which has not been watered during 15 days. Water-uptake distribution as in figure 6 (measurements). Nitrogen demand: 6 kg/(ha.day).

6. LITERATURE

Baker J M and LC H M Van Bavel, 1986. Resistance of plant roots to water loss. Agron J 78: 641-644.

Barber S A, 1984. Soil nutrient bioavailability: a mechanistic approach. New York, John Wiley, 398 pp.

Barraclough P B and P B Tinker, 1981. The determination of ionic diffusion coefficients in field soils. I. Diffusion coefficients in sieved soils in relation to water content and bulk density. J Soil Sci 32: 225-236.

Boekhold A E 1987. Waterbeweging en stoffentransport in de onverzadigde zone van de bodem: Invloed van bodemfysische eigeschappen op watergehalte profielen en tijd-diepte curven, Inst. Bodemvruchtbaarheid, Nota 171, 35 pp.

De Jager A, 1985. Response of plants to a localized nutrient supply. PhD Thesis, University of Utrecht, 137 pp.

De Willigen P and M Van Noordwijk, 1987. Roots, plant production and nutrient use efficiency. PhD Thesis, Agricultural University Wageningen, 280 pp.

Forsythe G E, M A Malcolm and C B Moler, 1977. Computer methods for mathematical computations. Prentice-Hall, Inc. Englewood Cliffs, 260 pp.

Lindelof T., 1981. Cern Computer Centre Program Library Manual, CERN Geneva, Switzerland.

Nye P H and P B Tinker, 1977. Solute movement in the soil-root system. Sudies in ecology, vol. 4. Oxford, Blackwell Scientific Publications, 342 pp.

Patankar S V, 1980. Numerical heat transfer and fluid flow. Hemisphere Publishing Corporation, New York etc., 183 pp.

Sharp R E and W J Davies, 1985. Root growth and water uptake by maize plants in drying soil. J Expt Bot 36: 1441-1456.

Ten Berge H F M , 1986. Heat and water transfer at the bare soil surface. PhD Thesis, Agricultural University Wageningen, 214 pp.

Van Genuchten M Th, 1980. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. Soil Sci. Soc. Am. J. 44: 892-898.

Van Kraalingen D W G and C Rappoldt, 1989. Subprograms in simulation models. Simulation Report CABO-TT nr 18, 54 pp.

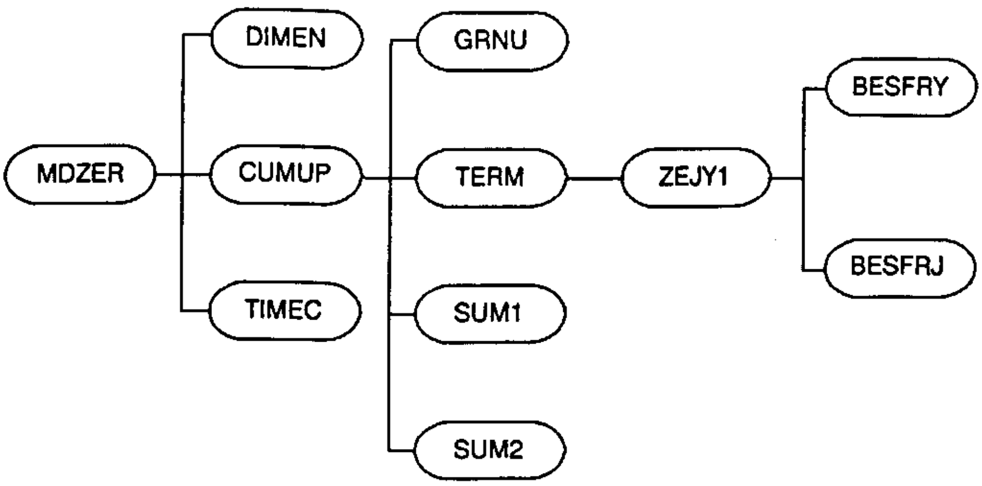
Wösten J H M 1987. Beschrijving van de waterretentie- en doorlatendheidskarakteristieken uit de Staringreeks met analytische functies. STIBOKA Rapport 2019, 53 pp.

APPENDIX I. Calculation of the nutrient uptake by a single root in a zero-sink situation.

The relation between the subroutines is given in figure AI-1.

The subroutine DIMEN calculates dimensionless parameters ρ, η, ϕ. These are used in TIMEC, which calculates the period of unconstrained uptake both in dimensionless and absolutes units.

Relational diagram of subroutines in MDZER



The core of the program is formed by the subroutine CUMUP which calculates the uptake and cumulative uptake after the period of unconstrained uptake, both exact and approximately. When the period of unconstrained uptake is less than zero, indicating that the concentration at the root surface drops to zero before a steady-rate situation has been developed, the calculations pertain to (9) and (10), otherwise to (4) and (5). CUMUP calls TERM, which calculates the terms of the summation series of (4), (5), (9), and (10), thereby using subroutine ZEJY1 which calculates the zero's α_n of the crossproduct given in the context of (3).

In the program the uptake rate (UP), the cumulative uptake (TUP), and their approximations (UPAP resp. TUPAP) are calculated in non-dimensionless form. In case of zero-sink uptake from the beginning the uptake rate in kg/(ha.day) is calculated as:

$$UP = 100 \frac{2HDC_i}{R_1^2} \omega_t$$

and a similar expression is used to calculate TUP. The factor 100 converts the units from mg/cm^2 to kg/ha.

If the period of unconstrained uptake is larger than zero, UP is calculated using:

$$UP/UPR = \omega_t/\omega$$

where UPR is the uptake rate (in kg/ha) in the period of unconstrained uptake. The cumulative uptake TUP is the sum of the total uptake in the period of unconstrained uptake and the integral of UP in time.

```
*-------------------------------------------------------------------------*
* Program MDZER.FOR                                                       *
* Purpose: Calculation of  diffusion and massflow  of a nutrient          *
* to a root. Zero-sink solution. Replenishment of water at outer boundary *
*  FORMAL PARAMETERS:  (I-input,O-output,C-control,IN-init,T-time)        *
*  name    type meaning                                    units   class  *
*  ----    ---- -------                                    -----   -----  *
*  RO      R4   radius root                                cm        I    *
*  DO      R4   diffusion coefficient in free water        cm2/day   I    *
*  DIF     R4   diffusion coefficient in soil              cm2/day   O    *
*  CF1     R4   coefficients for relation between                         *
*  CF2     R4   impedance and water content                               *
*  TFAC    R4   ratio between real and dimensionless time  day       O    *
*                                                                         *
*  SUBROUTINES called:                                                    *
*     DIMEN: calculates dimensionless variables                           *
*     TIMEC: calculates time constants                                    *
*     CUMUP: calculates uptake rates                                      *
*                                                                         *
* File usage:                                                             *
*     MDZER.INP: contains input-data                                      *
*     MDZER.OUT: detailed output on uptake, cumulative uptake, parameters *
*                                                                         *
*-------------------------------------------------------------------------*

      REAL LRV(20),TIME(50),UP(50)
      REAL UPAP(50),TUP(50),TUPAP(50)
      INTEGER NC(20)
      REAL NU
*
*------------------- Input data
      DATA TOL,DO/1.E-5,0.5/

      OPEN (FILE='MDZER.OUT',UNIT=45)
      OPEN (FILE='MDZER.INP',UNIT=40)
      OPEN (FILE='MASZER.LOG',UNIT=46)

      CALL MOFILP(40)
      READ (40,*) NLRV
      CALL MOFILP(40)
      READ (40,*) (LRV(I),I=1,NLRV)
      CALL MOFILP(40)
      READ (40,*) RO
      CALL MOFILP(40)
      READ(40,*) HSC
      CALL MOFILP(40)
      READ (40,*) AK
      CALL MOFILP(40)
      READ (40,*) TRANSP
      CALL MOFILP(40)
      READ (40,*) PAVAM
```

```
      CALL MOFILP(40)
      READ (40,*) UPR
      CALL MOFILP(40)
      READ (40,*) WC
      CALL MOFILP(40)
      READ(40,*) CF1,CF2,WCLOW
*
      PI=4.*ATAN(1.)

*------------------ calculation of diffusion coefficient as a function of
*------------------ water content, FIMP = impedance factor
      IF(WC.GE.WCLOW)THEN
         FIMP=CF1*WC+CF2
      ELSE
         FIMP=(CF1*WCLOW+CF2)*WCLOW*WC
         ENDIF
      DIF=D0*WC*FIMP
      TFAC=R0**2/DIF

      WRITE (45,100)
      WRITE(45,101) TRANSP,PAVAM,UPR,AK,WC,DIF

10    CONTINUE

*------------------ DO 25 over root densities
      DO 25 IL=1,NLRV
         CALL DIMEN(IL,LRV,R0,TRANSP,HSC,DIF,PAVAM,AK,WC,UPR,
     1                   R1,RHO,NU,CII,CT,AAA,BUF,PHI,ETA,COF)
         TNU=2.*NU

         CALL TIMEC(RHO,ETA,PHI,BUF,TNU,TFAC,TAUM,TAUD,TAUNU,FDD,FDNU,
     1                   CUD,CUNU)

         CALL CUMUP(20,RHO,TFAC,TAUM,TAUNU,UPR,BUF,TNU,
     1                   COF,R1,HSC,CII,DIF,TIME,UP,UPAP,TUP,TUPAP)

         WRITE(45,102) (TIME(I),UP(I),UPAP(I),TUP(I),
     $                   TUPAP(I),I=1,21)

25    CONTINUE

100   FORMAT(/1X,'PARAMETERVALUES:')
101   FORMAT(1X,'TRANSP  ='F7.2,1X,'CM/DAY'/
     1         1X,'PAVAM  ='F7.1,1X,'KG/HA'/
     2         1X,'UPR    ='F7.1,1X,'KG/(HA.DAY)'/
     3         1X,'AK     ='F7.1/1X,'WATERCONTENT = 'F8.2/
     4         1X,'DIF. COEFF = '1PE15.5)
102      FORMAT(
     1      /12X,'TIME',13X,'UP',11X,'UPAP',12X,'TUP',10X,'TUPAP'
     2      /(1X,1P5E15.5))
      STOP
      END
```

```
*------------------------------------------------------------------------*
* Subroutine TIMEC                                                       *
* Purpose: Calculation time constants, period of unconstrained uptake,   *
*          maximum uptake period and related parameters                  *
*                                                                        *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)       *
*  name    type meaning                                   units  class   *
*  ----    ---- -------                                   -----  -----   *
*  RHO     R4   Dimensionless radius soil cylinder          -      I     *
*  ETA     R4   Dimensionless root length                   -      I     *
*  PHI     R4   Dimensionless supply parameter              -      I     *
*  NU      R4   Dimensionless flux of water                 -      I     *
*  BUF     R4   Buffer capacity                             -      I     *
*  TAUM    R4   Dimensionless maximum period of unconstrained -    O     *
*               uptake                                                   *
*  TAUD    R4   Dimensionless period of unconstrained uptake  -    O     *
*               transport by diffusion only                              *
*  TAUNU   R4   Dimensionless period of unconstrained uptake             *
*               transport by diffusion and massflow           -    O     *
*  FDD     R4   Fractional depletion by unconstrained uptake  -    O     *
*               transport by diffusion only                              *
*  FDNU    R4   Fractional depletion by unconstrained uptake  -    O     *
*               transport by diffusion and massflow                      *
*  CUD     R4   Fraction left at end of period of unconstrained -  O     *
*               uptake, transport by diffusion only                      *
*  CUNU    R4   Fraction left at end of period of unconstrained -  O     *
*               uptake, transport by diffusion and massflow              *
*                                                                        *
*  Subroutines called : none                                             *
*  Functions called :                                                    *
*     GRNU: calculates G-function                                        *
*                                                                        *
*  File usage: time constants written to MDZER.OUT                       *
*------------------------------------------------------------------------*

      SUBROUTINE TIMEC(RHO,ETA,PHI,BUF,TNU,TFAC,TAUM,TAUD,TAUNU,FDD,
     1                 FDNU,CUD,CUNU)

*------------------------------------------------------------------------*
*------------------------------- Calculation period of unconstrained
*------------------------------- uptake TAUNU, maximum TAUM
      RHO2=RHO**2
      TAUM=(RHO2-1.)/RHO2*ETA*PHI
      TAUD=TAUM-BUF*GRNU(0.,RHO)
      TAUNU=TAUM-BUF*GRNU(TNU,RHO)
      FDNU=TAUNU/TAUM
      FDD=TAUD/TAUM
      CUNU=1.-FDNU
      CUD=1.-FDD

*
      TM=TFAC*TAUM
```

```
      TD=TFAC*TAUD
      TN=TFAC*TAUNU
      WRITE(45,206) TM,TAUM,TD,TAUD,TN,TAUNU
206   FORMAT(/15X,'TIME-CONSTANTS'/31X,'DAYS',11X,'DIML'
     1         /1X,'MAXIMUM:'12X,1P2E15.5/1X,'DIFFUSION :'9X,1P2E15.5
     2         /1X,'DIF. AND MASS FLOW :'1P2E15.5)

      RETURN
      END
```

```
*------------------------------------------------------------------------*
* Subroutine DIMEN                                                       *
* Purpose: calculation of dimensionless parameters                      *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)       *
*  name   type meaning                                 units     class   *
*  ----   ---- -------                                 -----     -----   *
*  LRV    R4   Root length density                     cm/cm3     I      *
*  RO     R4   Radius root                             cm         I      *
*  TRANSP R4   transpiration rate                      cm/day     I      *
*  HSC    R4   Root length                             cm         I      *
*  DIF    R4   Diffusion coefficient soil              cm2/day    I      *
*  PAVAM  R4   Potential available amount of nutrient  kg/ha      I      *
*  AK     R4   Adsorption constant nutrient            -          I      *
*  WC     R4   Water content                           -          I      *
*  UPR    R4   Uptake rate nutrient                    kg/(ha.day) I     *
*  R1     R4   Radius soil cylinder                    cm         O      *
*  RHO    R4   Dimensionless radius soil cylinder      -          O      *
*  NU     R4   Dimensionless flux of water             -          O      *
*  CII    R4   Initial concentration nutrient          mg/ml      O      *
*  CT     R4   Bulk density of nutrient                mg/cm3     O      *
*  AAA    R4   Uptake rate                             mg/(cm2.day) O    *
*  BUF    R4   Buffer capacity                         -          O      *
*  PHI    R4   Dimensionless supply parameter          -          O      *
*  ETA    R4   Dimensionless root length               -          O      *
*                                                                        *
*  Subroutines/Functions called:none                                     *
*                                                                        *
*  File usage: dimensionless parameters written to MDZER.OUT             *
*------------------------------------------------------------------------*

      SUBROUTINE DIMEN(IL,LRV,RO,TRANSP,HSC,DIF,PAVAM,AK,WC,UPR,
     1                 R1,RHO,NU,CII,CT,AAA,BUF,PHI,ETA,COF)
      DIMENSION LRV(30)
      REAL LRV,NU

      PI=4.*ATAN(1.)
      R1=1./SQRT(PI*LRV(IL))
      RHO=R1/RO
      RHO2=RHO**2
      TNU=-R1**2*TRANSP/(2.*DIF*HSC*(RHO2-1.))*(RHO2-1.)
      NU=TNU/2.
      BUF=WC+AK
      CII=0.01*PAVAM/(HSC*BUF)
      CT=CII*BUF
      AAA=UPR/100.
      PHI=DIF*CT/(AAA*RO)
      ETA=HSC/RO
      COF=4.*(NU+1.)/(RHO**(TNU+2.)-1.)

      WRITE(45,102) LRV(IL)
      WRITE(45,103) R1,RHO,RHO2,PHI,ETA,TNU
      RETURN
```

```
102   FORMAT(//'**************  LRV = '1PE15.5,2X,'**************'/
    1        1X,'COMBINED AND DIMENSIONLESS PARAMETERS')
103   FORMAT(1X,'R1    ='1PE15.5,1X,'CM'/1X,'RHO  ='1PE15.5,
    1        1X,'RHO2  ='1PE15.5/1X
    2        'PHI  ='1PE15.5/1X,'ETA  ='1PE15.5,1X/1X
    3        'TNU ='1PE15.5)

      END
```

```
*------------------------------------------------------------------------*
* Subroutine TERM                                                        *
* Purpose: calculates N terms form the series part of zero-sink solution *
*          concerning uptake of nutrients. The terms TE1 and TE2 pertain *
*          to steady-rate initial condition, TE3 and TE4 to constant     *
*          initial condition                                             *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)       *
*  name    type meaning                                   units  class   *
*  ----    ---- -------                                   -----  -----   *
*  RHO     R4   dimensionless root density, i.e. the radius   -     I    *
*               of the soil cylinder R1 over the root radius             *
*               R0                                                       *
*  FL      R4   Dimensionless flux of water                   -     I    *
*  TE1     R4   Term of solution for integrated uptake        -     O    *
*  TE2     R4   Term of solution for uptake rate              -     O    *
*  TE3     R4   Term of solution for integrated uptake        -     O    *
*  TE4     R4   Term of solution for uptake rate              -     O    *
*  N       I4   Number of terms                               -     I    *
*  AL      R4   Zero's from crossproduct of Bessel-functions: -     O    *
*               Y(FL,AL)J(FL+1,AL) - J(FL,AL)Y(FL+1,AL) = 0              *
*                                                                        *
*  Subroutines called:                                                   *
*     ZEJY1 : calculates zero's from crossproduct of Bessel-functions:   *
*               Y(FL,AL)J(FL+1,AL) - J(FL,AL)Y(FL+1,AL) = 0              *
*     BESFRJ : Bessel function of first kind and fractional order        *
*                                                                        *
*  Functions called: none                                                *
*                                                                        *
*  File usage: none                                                      *
*------------------------------------------------------------------------*
      SUBROUTINE TERM(RHO,FL,TE1,TE2,TE3,TE4,N,AL)
      REAL TE1(N),TE2(N),TE3(N),TE4(N),AL(N),BJ(2)
      REAL JNA,JN1RA

      DO 5 I=1,N
*------------------ Calculation of AL(I)
         CALL ZEJY1(AL(I),I,RHO,FL,20.,1.E-5,6)
         WRITE(46,*) 'I, AL ',I,AL(I)
*------------------
*------------------ Calculation of Bessel function J, with order FL
*------------------ and argument AL(I)
         CALL BESFRJ(AL(I),FL,6,BJ)
         JNA=BJ(1)

*------------------ Calculation of Bessel function J, with order FL+1
*------------------ and argument AL(I)*RHO
         RA=RHO*AL(I)
         CALL BESFRJ(RA,FL,6,BJ)
         JN1RA=BJ(2)

         TE1(I)=JN1RA**2/(AL(I)**4*(JNA**2-JN1RA**2))
         TE2(I)=TE1(I)*AL(I)**2
```

```
      TE3(I)=(2*FL*JNA*JN1RA-AL(I)*RHO**(FL+1)*JN1RA**2)/
     1           (AL(I)**3*RHO**(FL+1)*(JNA**2-JN1RA**2))
      TE4(I)=TE3(I)*AL(I)**2
    5 CONTINUE
      RETURN
      END
```

```
*---------------------------------------------------------------------------*
* Function SUM1                                                             *
* Purpose: Calculates summation series of solution of integrated            *
*          uptake rate, steady-rate initial condition                       *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)          *
*  name    type meaning                                       units  class  *
*  ----    ---- -------                                       -----  -----  *
*  T       R4   Dimensionless time                              -      I    *
*  AL      R4   Zero's from crossproduct of Bessel-functions:   -      I    *
*  BUF     R4   Dimensionless buffer-capacity                   -      I    *
*  TE1     R4   Terms from summation series                     -      I    *
*  N       R4   Number of terms                                 -      I    *
*                                                                           *
*  Subroutines called: none                                                 *
*  Functions called: none                                                   *
*  File usage: none                                                         *
*---------------------------------------------------------------------------*

      FUNCTION SUM1(T,TE1,AL,BUF,N)
      REAL TE1(N),AL(N)

      SUM1=0.
      DO 5 I=1,N
         TERM=TE1(I)*(1.-EXP(-AL(I)**2*T/BUF))
         SUM1=SUM1+TERM
         IF(ABS(TERM/SUM1).LT.1.E-5)GO TO 6
5     CONTINUE

6     RETURN
      END
```

```
*-----------------------------------------------------------------------------*
* Function SUM2                                                               *
* Purpose: Calculates summation series of solution of uptake rate, steady-*
*          rate initial condition                                             *
*  FORMAL PARAMETERS:  (I-input,O-output,C-control,IN-init,T-time)            *
*  name    type meaning                                       units  class    *
*  ----    ---- -------                                       -----  -----    *
*  T       R4   Dimensionless time                              -      I      *
*  AL      R4   Zero's from crossproduct of Bessel-functions:   -      I      *
*  BUF     R4   Dimensionless buffer-capacity                   -      I      *
*  TE2     R4   Terms from summation series                     -      I      *
*  N       R4   Number of terms                                 -      I      *
*                                                                             *
*  Subroutines called: none                                                   *
*  Functions called: none                                                     *
*  File usage: none                                                           *
*-----------------------------------------------------------------------------*

      FUNCTION SUM2(T,TE2,AL,BUF,N)
      REAL TE2(N),AL(N)

      SUM2=0.
      DO 5 I=1,N
         TERM=TE2(I)*EXP(-AL(I)**2*T/BUF)
         SUM2=SUM2+TERM
         IF(ABS(TERM/SUM2).LT.1.E-5)GO TO 6
5     CONTINUE

6     RETURN
      END
```

```
*-----------------------------------------------------------------------*
* Subroutine CUMUP                                                      *
* Purpose: calculates uptake rate and cumulative uptake in zero-sink    *
*          situation, and their steady-rate approximations as a function *
*          of time.                                                     *
*                                                                       *
*  FORMAL PARAMETERS:  (I-input,O-output,C-control,IN-init,T-time)      *
*  name    type meaning                                  units   class  *
*  ----    ---- -------                                  -----   -----  *
*  NI      I4   Number of time-steps                       -       I    *
*  TFAC    R4   Ratio between real and dimensionless time  day     I    *
*  NU      R4   Dimensionless flux of water                -       I    *
*  BUF     R4   Buffer capacity                            -       I    *
*  TAUM    R4   Dimensionless maximum period of unconstrained -    I    *
*               uptake                                                  *
*  TAUNU   R4   Dimension less period of unconstrained uptake  -   I    *
*  UPR     R4   Required uptake rate nutrient         kg/(ha.day) I     *
*  AL      R4   Zero's from crossproduct of Bessel-functions            *
*  TIME    R4   Time                                       days    0    *
*  UP      R4   Uptake rate in zero-sink situation    kg/(ha.day) 0     *
*  UPAP    R4   Uptake rate in zero-sink situation,   kg/(ha.day) 0     *
*               steady-rate approximation                               *
*  TUP     R4   Cumulative uptake                          kg/ha   0    *
*  TUPAP   R4   Cumulative uptake steady-rate approximation kg/ha  0    *
*                                                                       *
*  Subroutines called:                                                  *
*        TERM: calculates terms of summation series                     *
*                                                                       *
*  Functions called:                                                    *
*     SUM1,SUM2: produce values of summation series                     *
*        GRNU : Calculates G-function                                   *
*                                                                       *
*  File usage: none                                                     *
*-----------------------------------------------------------------------*

      SUBROUTINE CUMUP(NI,RHO,TFAC,TAUM,TAUNU,UPR,BUF,TNU,
     1                   COF,R1,HSC,CII,DIF,TIME,UP,UPAP,TUP,TUPAP)
      DIMENSION TIME(30),UP(30),UPAP(30),TUP(30),TUPAP(30)
      DIMENSION TE1(50),TE2(50),TE3(50),TE4(50),AL(50)
      REAL NU

         NU-TNU/2
         CALL TERM(RHO,NU,TE1,TE2,TE3,TE4,50,AL)
         WRITE(46,*) 'TERM ',(TE4(I),I-1,20)

         TAU-TAUNU
         IF(TAUNU.LT.0.)TAU-0.
         STEP-(TAUM-TAU)/NI
*----------------------------- Do 30 over time
         DO 30 I-1,NI+1
            TIM-FLOAT(I-1)*STEP
            TIME(I)-TFAC*(TAU+TIM)
```

```
          IF(TAUNU.GT.0.)THEN
             UP(I)=UPR*COF*SUM2(TIM,TE2,AL,BUF,50)
             UPAP(I)=UPR*EXP(-TIM/(BUF*GRNU(TNU,RHO)))
          ELSE
             UP(I)=-4.*HSC*DIF*CII*SUM2(TIM,TE4,AL,BUF,50)/
     1              R1**2*100.
             UPAP(I)=100.*HSC*DIF*CII/R1**2*(RHO**2-1.)
     1                *EXP(-TIM/(BUF*GRNU(TNU,RHO)))/GRNU(TNU,RHO)
          ENDIF
          IF(TIM.LE.0.)THEN
              TUP(I)=UPR*TFAC*TAU
          ELSE
             IF(TAUNU.GT.0.)THEN
                TUP(I)=UPR*TFAC*TAU+COF*BUF*UPR*TFAC
     1                 *SUM1(TIM,TE1,AL,BUF,50)
                TUPAP(I)=UPR*TFAC*BUF*GRNU(TNU,RHO)*(1.-EXP(-TIM/
     1                 (BUF*GRNU(TNU,RHO)))) + UPR*TFAC*TAU
               ELSE
                TUP(I)=UPR*TFAC*TAU-4.*100.*TFAC*HSC*DIF*CII
     1                 *BUF*SUM1(TIM,TE3,AL,BUF,50)/R1**2
                  TUPAP(I)=100.*HSC*DIF*CII/R1**2*(RHO**2-1.)*
     1                   TFAC*BUF*(1.-EXP(-TIM/(BUF*GRNU(TNU,RHO))))
             ENDIF
          ENDIF

 30         CONTINUE
      RETURN
      END
```

```
*------------------------------------------------------------------------*
* Function GRNU                                                          *
* Purpose: Calculation of G(nu,rho), (De Willigen & Van Noordwijk, 1987  *
*          page 130)                                                     *
*                                                                        *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)       *
*  name   type meaning                                      units  class *
*  ----   ---- -------                                      -----  ----- *
*  TNU    R4   2*NU, NU dimensionless flux of water           -      I   *
*  RHO    R4   Dimensionless radius soil cylinder             -      I   *
*                                                                        *
*  Subroutines called: none                                              *
*  Functions called: none                                                *
*  File usage: none                                                      *
*------------------------------------------------------------------------*

      FUNCTION GRNU(TNU,RHO)
      RHO2=RHO*RHO
      IF(TNU.EQ.0.)THEN
         G1=(1.-3.*RHO2)/4.
         G2=RHO2**2*ALOG(RHO)/(RHO2-1.)
         GRNU=(G1+G2)/2.
         RETURN
      ELSE
         GN1=(1.-RHO2)/2.
         GN2=RHO2*(RHO**TNU-1.)/TNU
         GN3=RHO2*(RHO**TNU-1.)*(TNU/2.+1.)/(TNU*
     1         (RHO**(TNU+2.)-1.))
         GN4=(1.-RHO**(TNU+4.))*(TNU/2.+1.)/((TNU+4.)*
     1         (RHO**(TNU+2.)-1.))
         GRNU=(GN1+GN2+GN3+GN4)/(TNU+2.)
      ENDIF
      RETURN
      END
```

```
*-----------------------------------------------------------------------*
* Subroutine ZEJY1                                                      *
* Purpose: calculates the n-th root of:                                 *
*          Y(ORD+1,RHO*X)*J(ORD,X)-Y(ORD,X)*J(ORD+1,RHO*X)              *
*          J,Y Bessel-functions of first kind, order ORD                *
*          Solution by Newton-raphson iteration, first estimate as given *
*          in Abramowitz & Stegun , page 374, 9.5.33.                   *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
*  name    type meaning                                    units  class *
*  ----    ---- -------                                    -----  ----- *
*  N       I4   rootnumber                                              *
*  TLIM    R4   maximum number of iterations                 -      I   *
*  EPS     R4   required accuracy                            -      I   *
*  X       R4   the value of the root                        -      O   *
*  IP      I4   Error indicator                              -      O   *
*                                                                       *
* Subroutines called:                                                   *
*     BESFRJ: calculates Bessel function J of fractional order          *
*       BESFRY: calculates Bessel function Y of fractional order        *
*  Functions called: none                                               *
*  File usage: none                                                     *
*-----------------------------------------------------------------------*

      SUBROUTINE ZEJY1(X,N,RHO,ORD,TLIM,EPS,ISIG)
      DIMENSION BJ(2),BY(2),BRJ(2),BRY(2)

      DOUBLE PRECISION JNA,JNRA,JN1A,JN1RA,YNA,YNRA,YN1A,
     $                   YN1RA,FA,FAA
*     PRINT *,'TLIM ',TLIM
      IP=0
      ORD1=ORD+1
      PI=3.141592654
      S=N
      X=(S-0.5)*PI/(RHO-1.)
      TEL=0.
1     SOL=X
      TEL=TEL+1
      IF(TEL.GT.TLIM)GO TO 5
      IF(SOL.LE.0.)GO TO 10
      RSOL=RHO*SOL
      CALL BESFRJ(SOL,ORD,ISIG,BJ)
      CALL BESFRY(SOL,ORD,ISIG,BY)
      CALL BESFRJ(RSOL,ORD,ISIG,BRJ)
      CALL BESFRY(RSOL,ORD,ISIG,BRY)
      JNA=BJ(1)
      JN1A=BJ(2)
      YNA=BY(1)
      YN1A=BY(2)
      JNRA=BRJ(1)
      JN1RA=BRJ(2)
      YNRA=BRY(1)
      YN1RA=BRY(2)
```

```
      DJN=-JN1A+ORD/SOL*JNA
      DYN=-YN1A+ORD/SOL*YN1A
      DJN1R=RHO*(JNRA-ORD1/RSOL*JN1RA)
      DYN1R=RHO*(YNRA-ORD1/RSOL*YN1RA)
      FA=JNA*YN1RA-JN1RA*YNA
*     PRINT *,'TEL,SOL,FA ',TEL,SOL,FA
      FAA=YN1RA*DJN+JNA*DYN1R-JN1RA*DYN-YNA*DJN1R
      X=SOL-FA/FAA
      IF(ABS((SOL-X)/X).GT.EPS)GO TO 1
      RETURN
5     CONTINUE
      PRINT  100
      IP=2
      RETURN
10    PRINT 101,TEL,SOL
      IP=3
      RETURN

100   FORMAT(1H0,'TOO MANY ITERATIONS IN ZEJY1')
101   FORMAT(1H0,'AFTER',F5.0,1X,'ITERATION, NEGATIVE ARG: '
     11PD15.5)

      END
```

```
*-----------------------------------------------------------------------------*
* Subroutine: BESFRJ                                                          *
* Purpose: calculates bessel-functions first kind                             *
* for argument x and (fractional) orders alph and alph+1                      *
*                                                                             *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)            *
*  name    type meaning                                       units  class    *
*  ----    ---- -------                                       -----  -----    *
*  X       R4   argument of Besselfunction                      -      I      *
*  ALPH    R4   order of Besselfunction                         -      I      *
*  ISIG    R4   required significant digits                     -      I      *
*  BJ      R4   contains the values of J(ALPH,X), J(ALPH+1,X)          O      *
*                                                                             *
* Subroutines called:                                                         *
*      BESJN from CERNLIB                                                     *
*  Functions called: none                                                     *
*  File usage: none                                                           *
*-----------------------------------------------------------------------------*

      SUBROUTINE BESFRJ(X,ALPH,ISIG,BJ)
      DIMENSION BBJ(2),BJ(1)
C
      IF(ALPH.LT.0.)GO TO 5
      NNN=ALPH
      ALP=ALPH-NNN
      CALL BESJN(X,ALP,NNN,ISIG,BBJ)
      BJ(1)=BBJ(1)
      BJ(2)=BBJ(2)
      RETURN
5     CONTINUE
      NNN=ALPH-1.
      ALP=ALPH-NNN
      CALL BESJN(X,ALP,NNN,ISIG,BBJ)
      BJ(1)=BBJ(2)
      BJ(2)=BBJ(1)
      RETURN
      END
```

```
*---------------------------------------------------------------------------*
* Subroutine BESFRY
* Purpose: calculates modified bessel-functions first kind
* for argument x and (fractional) orders alph and alph+1
*  ----   ----  ------                                      -----   -----   *
*  X      R4    argument of Besselfunction                    -        I
*  ALPH   R4    order of Besselfunction                       -        I
*  ISIG   R4    required significant digits                   -        I
*  BY     R4    contains the values of Y(ALPH,X), Y(ALPH+1,X)          O
*
*  Subroutines called:
*     BESFRJ
*  Functions called: none
*  File usage: none
*---------------------------------------------------------------------------*

      SUBROUTINE BESFRY(X,ALPH,ISIG,BY)
      DIMENSION BY(1),BBJ(2),BBMJ(2)
      PI=3.1415926535
      CO=COS(ALPH*PI)
      SI=SIN(ALPH*PI)
      ALP=-ALPH
      CALL BESFRJ(X,ALP,ISIG,BBMJ)
      CALL BESFRJ(X,ALPH,ISIG,BBJ)
      BY(1)=(BBJ(1)*CO-BBMJ(1))/SI
      BY(2)=-(2./(PI*X)-BBJ(2)*BY(1))/BBJ(1)
    1 CONTINUE
      RETURN
      END
```

APPENDIX II. Numerical solution of radial flow of water to a plant root.

Background

The program solves the following partial differential equation,

$$\frac{\partial \theta}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}D_w\frac{\partial \theta}{\partial r} \qquad \text{(AII-1)}$$

with boundary conditions:

$$r = \rho \ , \ \ \frac{\partial \theta}{\partial r} = 0 \qquad \text{(AII-2)}$$

and either:

$$r = 1, \ \left(D_w\frac{\partial \theta}{\partial r}\right) = constant = q \qquad \text{(AII-3a)}$$

or:

$$r = 1 \ \ , \theta = constant = \theta_{rs} \qquad \text{(AII-3b)}$$

The initial condition is simply :

$$t = 0, \ \theta = \text{constant} = \theta_i \qquad \text{(AII-4)}$$

The meaning of the symbols can be found in table 1 in the main text. Equation (AII-1) is Richard's equation in cylindrical coordinates.

Before discretisizing the partial differential equation (AII-1), the hollow cylinder with inner radius 1 and outer radius ρ, is transformed to a rectangle by the transformation:

$$x = \ln r \qquad \text{(AII-5)}$$

This transforms (AII-1)-(AII-3) into resp.:

$$e^{2x}\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial x}D_w\frac{\partial \theta}{\partial x} \qquad \text{(AII-6)}$$

$$x = \ln\rho \ , \ \ \frac{\partial \theta}{\partial x} = 0, \qquad \text{(AII-7)}$$

$$x = 0, \ \frac{\partial \theta}{\partial x} = q \qquad \text{(AII-8a)}$$

and

$x = 0, \ \theta = \theta_{rs}$

(AII-8b)

Equation (AII-6) is discretized by dividing the region $0 < x < \ln\rho$ into a number of control-volumes and integrating of (AII-6) over the control-colume and a finite time-step Δt (Patankar, 1980) following a complete implicit scheme. This leads to a set of n equations in the n unknown θ, where n is the number of control volumes. These equations have the form:

$$a_i\theta_i = b_i\theta_{i+1} + c_i\theta_{i-1} + d_i \qquad \text{(AII-9)}$$

where

$$b_i = \frac{\overline{D_i}}{\Delta x}, \quad c_i = \frac{\overline{D_{i-1}}}{\Delta x} \quad ,$$

$$a_i = b_i + c_i + \sinh(\Delta x)\frac{e^{2(i-1)\Delta x}}{\Delta t} \quad , \quad d_i = \sinh(\Delta x)\frac{e^{2(i-1)\Delta x}}{\Delta t}\theta_i^0$$

Δx is the thickness of the control-volumes, Δt the time-step, D_i and D_{i-1} the average diffusivities at the boundaries of control-volume i and i+1, resp i-1 and i. It is to be understood that all values of θ are the new unknown values at time $t+\Delta t$ except where the superscript ^{0} is used. The coefficients a,b,c and have special values at the boundaries, where i=1 resp i=n. For i=1 (x=0), if condition of constant flux (AII-8a) is used:

$$a_1 = \frac{(e^{\Delta x}-1)}{2\Delta t} + \frac{\overline{D_1}}{\Delta x} \quad , \quad b_1 = \frac{\overline{D_1}}{\Delta x}$$

$$c_1 = 0, \quad d_1 = -q + \frac{(e^{\Delta x}-1)}{2\Delta t}\theta_1^0 \ ,$$

whereas when for constant watercontent (AII-8b):

$$a_1 = 1, \quad b_1 = 0, \quad c_1 = 0, \quad d_1 = \theta_{rs}.$$

For i=n ($x=\ln\rho$)

$$a_n = \frac{(\rho^2 - e^{2(n-1)\Delta x - \Delta x})}{2\Delta t}, \quad b_n = 0$$

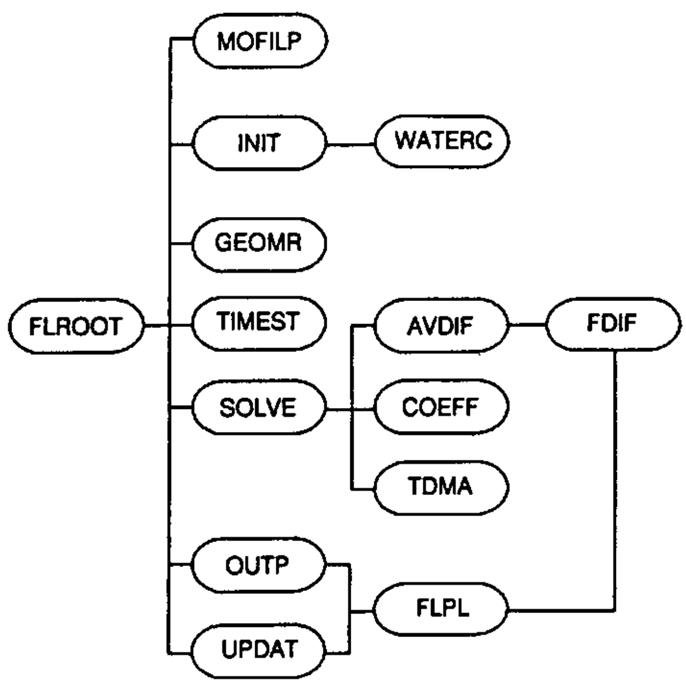
$$c_n = \frac{\overline{D_{n-1}}}{\Delta x}, \quad d_n = \frac{(\rho^2 - e^{(n-1)\Delta x - \Delta x})}{2\Delta t}\theta_n^0$$

The solution of the discretization eqations (AII-9) is found by the TDMA algorithm (Patankar, 1980).

Program

The relation between the subroutines is given in the diagram in Figure AII-1. The core of the program is formed by the subroutine SOLVE wherein the discretization equations are solved. Due to the nonlinear relation between the diffusivity and the water content the solution is found by iteration. If necessary underrelaxation can be used to avoid divergence (Patankar, 1980, Ch 4. page 67).

Relational diagram of subroutines in FLROOT



The relation between diffusivity and water content is as proposed by Van Genuchten (1980):

$$D = K_s \frac{W^l}{(\theta_{sat}-\theta_r)\alpha n}\left(W^{-\frac{1}{m}}-1\right)^{(-m)}\left[1-\left(1-W^{\frac{1}{m}}\right)^m\right]^2 \qquad \text{(AII-10)}$$

where

$$W = \frac{\theta - \theta_r}{\theta_{sat} - \theta_r}\ , \ m = 1 - \frac{1}{n}$$

This is calculated in the function FDIF, and is used in the function FLPL which calculates the integral give in (11). The numerical integration is performed by the function GAUSS (not shown) which is taken from CERNLIB Lindelof, 1981).

```
*----------------------------------------------------------------------*
*  Program FLROOT                                                      *
*  Purpose : Calculation of flow of water towards a root.              *
*          The geometry is a hollow cylinder representing the soil around*
*          the root. Boundary conditions are: constant flux, or constant *
*          water-content at the root surface (the inner cylinder) and  *
*          zero-flux at the outer cylinder. The partial differential   *
*          equation is solved by completely implicit finite difference *
*          scheme, as described in S.V. Patankar, 1980, Numerical heat *
*          transfer and fluid flow.                                    *
*          The flow is also calculated with an analytical function,    *
*          based on the steady-rate approximation (De Willigen and     *
*          Van Noordwijk, 1987, Roots, plant production and nutrient use *
*          efficiency page 149, (9.73)).                               *
*                                                                      *
*  The mutual relations between matric potential, water content,       *
*  and hydraulic conductivity are given by Van Genuchten functions.    *
*                                                                      *
*  SUBROUTINES called:                                                 *
*       MOFILP: organizes input                                        *
*       INIT: initializes watercontent                                 *
*       GEOMR: calculates geometry, thickness control-volumes etc.     *
*       SOLVE: solves the discretization equations                     *
*       UPDAT: updates watercontent, calculate totals, averages, etc.  *
*       OUTP: organizes the output                                     *
*                                                                      *
*  FILE usage:                                                         *
*        1. input from screen (format: soilname.DAT),unit 40,          *
*           contains soil and root data (I)                            *
*        2. CONTR.DAT, unit 40, contains data that control boundary    *
*           conditions, iteration, relaxation (I)                      *
*        3. soilname.PET, unit 41, main outputfile (O)                 *
*        4. soilname.PLO, unit 48, plotfile with data (O)              *
*                                                                      *
*----------------------------------------------------------------------*

      DIMENSION G(30),C(30),WCN(30),DIFDN(30)
      DIMENSION A(30),X(30),B(30),D(30),WCOL(30)
      DIMENSION USTR(30),WCSTR(30),DW(30)

      REAL KS,M,N,L
      REAL U(30),R(30),LRV
      REAL WC(30),DWCDT(30),DIFD(30)
      CHARACTER*20 NAME,INP,OUTF
      LOGICAL MODE

      COMMON /FDIF1 / WCSAT,WCR,M,N,L,ALPH,KS

      DATA FLB/1./

*-------- INITIAL --------*
```

```
      PI=ATAN(1.)*4.
      TWOPI=2.*PI

*------------------- input
      PRINT *,'SOILNAME?'
      READ(*,100) NAME

*--------------------------- File with soil and root-data is opened
      OPEN (UNIT=40,FILE=NAME)

*-------------------------- Outputfile is opened
      I = INDEX (NAME,'.')
      IF (I.EQ.0) I = INDEX (NAME,' ')
      NAME(I:I+3) = '.PET'
      OPEN (UNIT=41,FILE=NAME)

*------------------------- Plotfile is opened
      NAME(I:I+3)='.PLO'
      OPEN(UNIT=48,FILE=NAME)

*------------------------------ Reading of soil and root data
      CALL MOFILP(40)
      READ(40,100) NAME
      CALL MOFILP(40)
      READ(40,*) WCSAT,WCR,WCMIN,WCREF,PLIM,PMIN
      PRINT *,'WCSAT,WCR,WCMIN,WCREF,PLIM ',WCSAT,WCR,WCMIN,WCREF,PLIM
      CALL MOFILP(40)
      READ(40,*) KS,N,L,ALPH
      M=1.-1./N
      CALL MOFILP(40)
      READ(40,*) LRV,RO,HSC
      CALL MOFILP(40)
      READ(40,*) EVAP
      CALL MOFILP(40)
      READ(40,*) WCI
      CALL MOFILP(40)
      READ(40,*) IR
      CLOSE(40)
*------------------------- SOILNAME.DAT is closed


*------------------------ CONTR.DAT is opened
      OPEN(FILE='CONTR.DAT',UNIT=40)
*----------------------------- reading control data
      CALL MOFILP(40)
      READ (40,*) TFAC
*----------------------------------- maximum change watercontent
*----------------------------------- over time-step
      CALL MOFILP(40)
      READ (40,*) CHMAX
*----------------------------------- iteration tolerance
      CALL MOFILP(40)
```

```
      READ (40,*) EPS
*-------------------------------- relaxation factor
      CALL MOFILP(40)
      READ (40,*) RELF
*--------------------------------- CWRS < 0 , watercontent at root surface
*-------------------------------------------- constant
      CALL MOFILP(40)
      READ (40,*) CWRS
*-------------------------------------------- read waterpotential root
*-------------------------------------------- surface
      IF(CWRS.LT.0.)THEN
         CALL MOFILP(40)
         READ (40,*) PRS
         FLB=-1.
      ELSE
      ENDIF
*----------------------- Close CONTR.DAT
      CLOSE(40)

*------------------ End input
*------------------ Initialisation
      CALL INIT(IR,LRV,HSC,RO,WCI,PLIM,EVAP,FLB,PRS,
     1          R1,SUMWI,RHO,ETA,WCLIM,QT,RAT,WC,WCAPP)

*----------------------- Reading time data from screen
      TMAX=SUMWI/(PI*R1**2*EVAP)
      PRINT *,'MAXIMUM EXTRACTION TIME IN DAYS ',TMAX
      PRINT *,'GIVE FINTIM IN DAYS'
      READ *,FINTIM
      PRINT *,'PRINT-INTERVAL IN DAYS'
      READ *,PRDEL

*------------------ Geometry
      CALL GEOMR(IR,RHO,X,R,DX,DX2)

      PRDELT=PRDEL*RAT
      IP=0
      T=0
      WRITE(41,*) 'FINTIM,DELT,DT ',FINTIM,DELT,DT

      DELT=.1
      FIT=0
      TRPAR=1.
      WCAPP=WCI

*-------- DYNAMIC --------*

10    CONTINUE


      IF(WC(1).LE.WCLIM.AND.FLB.GT.0.)THEN
         FLB=-1.
```

```
         WRITE(41,*) 'TIME = ',TIME,'   WC(1).LE.WCLIM'
         PRINT 101,TIME,DELT
         PRINT 102,(WC(K),K=1,IR)
         CALL OUTP(IR,R,R0,HSC,R1,T,TIME,DELT,WCN,WC,DWCDT,DIFD,
     1             BAL,RUPW,RUPWSR,CUPW,CUPWSR,WCAV,WCAPP)
      ENDIF

*------------------ Check if time exceeds fintim
      IF(TIME.GE.FINTIM)GO TO 20

*------------------ Check if output should be written
      IF(TIME.GE.(IP*PRDEL))THEN
         IP=IP+1
         CALL OUTP(IR,R,R0,HSC,R1,T,TIME,DELT,WCN,WC,DWCDT,DIFD,
     1             BAL,RUPW,RUPWSR,CUPW,CUPWSR,WCAV,WCAPP)
      ENDIF

      CALL SOLVE(IR,WC,WCN,DELT,DX,RHO,QT,FLB,RELF,EPS,WCLIM,
     1           DIFD)

      CALL UPDAT(IR,WC,WCN,WCLIM,R,R0,R1,HSC,DELT,T,
     1      DWCDT,RUPW,RUPWSR,CUPWSR,CUPW,BAL,WCAV,WCAPP)

      TIME=T*(R0*R0)

      CALL TIMEST(DELT,FLB,TFAC,CHMAX,WC,DWCDT,PRDELT)

      GO TO 10

20    CONTINUE
      WRITE(41,*) ' TIME .GE. FINTIM '
      CALL OUTP(IR,R,R0,HSC,R1,T,TIME,DELT,WCN,WC,DWCDT,DIFD,
     1          BAL,RUPW,RUPWSR,CUPW,CUPWSR,WCAV,WCAPP)
      STOP

100   FORMAT(A20)
101   FORMAT(1H0,'TIME ='1PE15.5,1X,'DELT ='1PE15.5)
102   FORMAT(1H0,/13X,'WC'/(1PE15.5))

      END
```

```
*-----------------------------------------------------------------------------------*
*  Subroutine INIT                                                                   *
*  Purpose: Initialisation of watercontent, calculation of parameters                *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)                   *
*  name     type meaning                                         units     class     *
*  ----     ---- -------                                         -----     -----     *
*  LRV      R4   Root length density                             cm/cm3    I         *
*  HSC      R4   Root lenght                                     cm        I         *
*  RO       R4   Root radius                                     cm        I         *
*  WCI      R4   Initial water content                           -         I         *
*  PLIM     R4   Limiting value matric potential                 cm        I         *
*  EVAP     R4   Required transpiration                          cm/day    I         *
*  FLB      R4   Parameter that controls boundary condition                I         *
*                at root surface. FLB>0 constant flux,                               *
*                FLB<0 constant watercontent                                         *
*  PRS      R4   Matric potential at root surface                cm        I         *
*  R1       R4   Radius soil cylinder                            cm        O         *
*  RHO      R4   Dimensionless radius soil cylinder              -         O         *
*  SUMWI    R4   Initial total amount of water in soil           cm3       O         *
*  ETA      R4   Dimensionless root lenght                       -         O         *
*  WCLIM    R4   limiting value water content                    -         O         *
*  QT       R4   Required uptake rate of water by root           cm2/day   O         *
*  WC       R4   Water content                                   -         O         *
*  WCAPP    R4   Water content calculated by steady-rate         -         O         *
*                approximation                                                       *
*                                                                                    *
*  Subroutines called: none                                                          *
*                                                                                    *
*  Functions called:                                                                 *
*     WATERC: calculates water content as a function of matric potential             *
*                                                                                    *
*  File usage: soilname.PET                                                          *
*-----------------------------------------------------------------------------------*

      SUBROUTINE INIT(IR,LRV,HSC,RO,WCI,PLIM,EVAP,FLB,PRS,
     1                R1,SUMWI,RHO,ETA,WCLIM,QT,RAT,WC,WCAPP)
      DIMENSION WC(30)
      REAL M,N,L,KS,LRV
      COMMON /FDIF1 / WCSAT,WCR,M,N,L,ALPH,KS

      PI=4.*ATAN(1.)
      R1=1./SQRT(PI*LRV)
      SUMWI=PI*(R1*R1-RO*RO)*HSC*WCI
      RHO=R1/RO
      ETA=HSC/RO
      WCLIM=WATERC(PLIM,WCSAT,WCR,N,ALPH)
      QT=RHO**2/(2.*ETA)*EVAP*RO
      RAT=1./(RO*RO)

*---------------- initialisation OF WATER CONTENT
      DO 30 K=2,IR+1
         WC(K)=WCI
```

```
30        CONTINUE
          IF(FLB.GT.0.)THEN
             WC(1)=WCI
          ELSE
             WC(1)=WATERC(PRS,WCSAT,WCR,N,ALPH)
          ENDIF
          WCAPP=WCI

*----------------- initial output
          WRITE(41,100) LRV,EVAP,RHO,ALPH,QT,RAT
          PRINT     100,LRV,EVAP,RHO,ALPH,QT,RAT
          PRINT 101,SUMWI
          WRITE(41,101) SUMWI
          RETURN

100       FORMAT(1X,//,1X,'ROOT-DENSITY CM-2:'F5.1/
     1          1X,'EVAPOTRANSPIRATION CM/DAY :'F6.2/
     2          1X,'RHO ;'1PE15.5/
     3          1X,'ALPH :'1PE15.5/1X,
     4          1X,'DIMENSIONLESS UPTAKE QT:'1PE15.5/
     5          1X,'RATIO DIMENSIONLESS TIME AND REAL TIME:'1PE15.5)
101       FORMAT(1X,'SUMWI='1PE15.5)

          END
```

```
*-----------------------------------------------------------------------*
* Subroutine: SOLVE                                                     *
* Purpose: construction of coefficient-matrix of discretisized equations *
*          and solving the equations by TDMA                            *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
*  name    type meaning                                   units  class  *
*  ----    ---- -------                                   -----  -----  *
*  WC      R4   Water content                             -       I     *
*  WCN     R4   New values of water content               -       I,O   *
*  DELT    R4   Time step in scaled units                 day/cm2 I     *
*  DX      R4   Thickness control-volume                  -       I     *
*  RHO     R4   Dimensionless radius soil cylinder        -       I     *
*  QT      R4   Uptake rate of water by root              cm2/day I     *
*  FLB     R4   Parameter that controls boundary condition        I     *
*               at root surface. FLB>0 constant flux,                   *
*               FLB<0 constant watercontent                             *
*  RELF    R4   Relaxation factor                         -       I     *
*  EPS     R4   Convergence criterion for iteration       -       I     *
*  DIFD    R4   Average diffusivity at face control volume -      O     *
*  WCLIM   R4   limiting value water content              -       O     *
*                                                                       *
*  Subroutines called:                                                  *
*     COEFF: calculates coefficients discretization equations          *
*     TDMA : solves discrtization equations                            *
*                                                                       *
*  Functions called:                                                    *
*     AVDIF: calculates average diffusivity at interface               *
*               two control-volumes                                     *
*  File usage: none                                                     *
*                                                                       *
*-----------------------------------------------------------------------*

      SUBROUTINE SOLVE(IR,WC,WCN,DELT,DX,RHO,QT,FLB,RELF,EPS,WCLIM,
     1                 DIFD)
      DIMENSION WC(30),WCN(30)
      DIMENSION WCOL(30),DIFD(30),A(30),B(30),C(30),D(30),DIFDN(30)

10     CONTINUE
*-------------------- average diffusivities are calculated
*-------------------- old values of WC stored in WCOL
      DO 20 K=1,IR-1
         DIFD(K)=AVDIF(WC(K),WC(K+1),.FALSE.)
         WCOL(K)=WC(K)
20    CONTINUE
      WCOL(IR)=WC(IR)

*-------------------- tentative new values of WC (WCN) calculated
*------------------------- no relaxation
      CALL COEFF(IR,A,B,C,D,WC,DIFD,1.,DX,RHO,DELT,QT,FLB,WC(1),
     1 WCOL,RELF,0)
      CALL TDMA(IR,A,B,C,D,WCN)
```

```
*------------------------ iterative search for better new values of WC
        DO 50 IT=1,10
           IF(WCN(1).LT.0.)GO TO 60
           DO 30 K=1,IR-1
              TRPAR=1.
              IF(WCN(K).LT.WCLIM)THEN
                 TRPAR=-1.
              ENDIF
              DIFDN(K)=AVDIF(WCN(K),WCN(K+1),.FALSE.)
30         CONTINUE
*---------------------------- criterion of convergence is relative
*---------------------------- difference between consecutive values of
*---------------------------- diffusivity at first or second gridpoint
           DIFCRT=(DIFDN(1)-DIFD(1))/DIFD(1)
           IF(FLB.LT.0.)DIFCRT=(DIFDN(2)-DIFD(2))/DIFD(2)
           IF(ABS(DIFCRT).LT.EPS)THEN
            GO TO 70
           ELSE
            CALL COEFF(IR,A,B,C,D,WC,DIFDN,1.,DX,RHO,DELT,QT,FLB,
     1                 WC(1),WCOL,RELF,IT)
            CALL TDMA(IR,A,B,C,D,WCN)
            DO 40 K=1,IR
                 TRPAR=1.
                 IF(WCN(K).LT.WCLIM)THEN
                    TRPAR=-1.
                 ENDIF
                 DIFD(K)=DIFDN(K)
                 WCOL(K)=WCN(K)
40          CONTINUE
           ENDIF
50      CONTINUE

60      DELT=DELT/2.
        GO TO 10

*------------------------ end of iteration
70      CONTINUE

        RETURN
        END
```

```
*-----------------------------------------------------------------------*
* Function FLPL                                                         *
* Purpose: Calculation of matric flux potential as integral of diffusivity*
*          over watercontent                                            *
*                                                                       *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)     *
*  name   type meaning                                  units   class  *
*  ----   ---- -------                                  -----   -----  *
*  WC     R4   water content                            ml/cm3  I      *
*  WCSAT  R4   Watercontent at saturation               ml/cm3  I      *
*  N      R4   parameter from conductivity-watercontent   -     I      *
*              relation                                                 *
*  ALPH   R4   parameter                                cm-1    I      *
*  WCR    R4   residual water content                   ml/cm3  I      *
*  KS     R4   saturated conductivity                   cm/day  I      *
*                                                                       *
* Subroutines called: none                                              *
*                                                                       *
* Functions called :                                                    *
*        GAUSS: a function from CERNLIB which performs numerical        *
*               integration according to Gauss' method.                 *
*        FDIF: calculates diffusivity as a function of water content.   *
*                                                                       *
* File usage: none                                                      *
*                                                                       *
*-----------------------------------------------------------------------*

      REAL FUNCTION FLPL(WC)
      REAL KS,M,N,L
      COMMON /FDIF1/ WCSAT,WCR,M,N,L,ALPH,KS
      EXTERNAL FDIF

      FLPL=GAUSS(FDIF,WC,0.99*WCSAT,.001)

      RETURN
      END
```

```
*-----------------------------------------------------------------------*
* FUNCTION WATERC                                                       *
* Purpose : Calculation of watercontent by hydraulic head Van Genuchten *
*           function.                                                   *
*  FORMAL PARAMETERS:  (I-input,O-output,C-control,IN-init,T-time)      *
*  name   type meaning                                   units  class   *
*  ----   ---- -------                                   -----  -----   *
*  PRES   R4   hydraulic head                            cm       I     *
*  WCSAT  R4   Watercontent at saturation                ml/cm3   I     *
*  N      R4   parameter from conductivity-watercontent   -       I     *
*              relation                                                 *
*  ALPH   R4   parameter                                 cm-1     I     *
*  WCR    R4   residual water content                    ml/cm3   I     *
*  WATERC R4   watercontent                              ml/cm3   O     *
*                                                                       *
* Subroutines/functions called: none.                                   *
*                                                                       *
* File usage: none                                                      *
*                                                                       *
*-----------------------------------------------------------------------*

      FUNCTION WATERC(PRES,WCSAT,WCR,N,ALPH)
      REAL M,N
      M=1.-1./N
      IF(PRES.LE.0.)THEN
         TRPAR=-1
      ENDIF
      HELP=(1.+(ALPH*PRES)**N)
      IF(HELP.LE.0.)THEN
         TRPAR=-1
      ELSE
      ENDIF
      WATERC=(WCSAT-WCR)*(1.+(ALPH*PRES)**N)**(-M)+WCR
      RETURN
      END
```

```
*-----------------------------------------------------------------------*
* Function FDIF                                                         *
* Purpose: Calculation of diffusivity from watercontent                 *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
*  name   type meaning                                   units  class   *
*  ----   ---- -------                                   -----  -----   *
*  WCSAT  R4   Watercontent at saturation                ml/cm3  I      *
*  N      R4   parameter from conductivity-watercontent    -     I      *
*              relation                                                 *
*  ALPH   R4   parameter                                 cm-1    I      *
*  WCR    R4   residual water content                    ml/cm3  I      *
*  FDIF   R4   Diffusivity                               cm2/day        *
*                                                                       *
*  SUBROUTINES called: none                                             *
*  FUNCTIONS called: none                                               *
*  FILE usage:none                                                      *
*-----------------------------------------------------------------------*

      FUNCTION FDIF(WC)
      REAL KS,M,N,L
      COMMON /FDIF1/ WCSAT,WCR,M,N,L,ALPH,KS

      W=(WC-WCR)/(WCSAT-WCR)
      M=1.-1./N
      TRPAR=1.
      IF(W.LT.0.)THEN
         TRPAR=-1.
      ENDIF
      WL=W**(-1./M)
      TRPAR=1
      IF(WL.LE.1.)THEN
         TRPAR=-1
      ENDIF
      FDIF=KS*W**L/((WCSAT-WCR)*ALPH*N)*(WL-1.)**(-M)*
     1     (1.-(1.-W**(1./M))**M)**2
      RETURN
      END
      FUNCTION FG(R)
      FG=0.5*((1-3.*R**2)/4.+R**4*ALOG(R)/(R**2-1))
      RETURN
      END
```

```
*------------------------------------------------------------------------*
* Subroutine TIMEST                                                      *
* Purpose: calculation of time step DELT                                 *
*                                                                        *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)       *
*  name   type meaning                                   units    class  *
*  ----   ---- -------                                   -----    -----  *
*  DELT   R4   timestep in scaled units                  day/cm2  I,O    *
*  TFAC   R4   multiplying factor, if possible DELT is            I      *
*              increased by TFAC                                         *
*  CHMAX  R4   permitted maximum relative change in        -      I      *
*              watercontent                                              *
*  WC     R4   watercontent                                       I      *
*  DWCDT  R4   rate of change of WC                      cm2/day  I      *
*  PRDELT R4   print-interval in scaled units            day/cm2  I      *
*                                                                        *
* Subroutines/Functions called: none                                     *
* FILE usage: none                                                       *
*------------------------------------------------------------------------*

      SUBROUTINE TIMEST(DELT,FLB,TFAC,CHMAX,WC,DWCDT,PRDELT)
      DIMENSION WC(30),DWCDT(30)

      IF(FLB.GT.0.)THEN
         DCH=CHMAX*WC(1)/ABS(DWCDT(1))
      ELSE
         DCH=CHMAX*WC(2)/ABS(DWCDT(2))
      ENDIF
      DELT2=DELT*TFAC
      DELT=AMIN1(DCH,PRDELT,DELT2)

      RETURN
      END
```

```
*-----------------------------------------------------------------------------*
* Subroutine: UPDAT                                                           *
* Purpose: Updating of watercontent, time, total amount of water in soil,     *
*          decrease of total water,  transpiration by steady-rate             *
*          approximation                                                      *
*  FORMAL PARAMETERS:  (I-input,O-output,C-control,IN-init,T-time)            *
*  name    type meaning                                     units    class    *
*  ----    ---- -------                                     -----    -----    *
*  R       R4   Dimensionless radial coordinate               -       I       *
*  RO      R4   Radius root                                   cm      I       *
*  HSC     R4   Root length                                   cm      I       *
*  R1      R4   Radius soil cylinder                          cm      I       *
*  WC      R4   Water content                                 -       I,O     *
*  WCN     R4   New values of water content                   -       I       *
*  DWCDT   R4   Rate of change of water content             cm2/day I        *
*  DELT    R4   timestep in scaled units                    day/cm2 I        *
*  RHO     R4   Dimensionless radius soil cylinder            -       I       *
*  WCLIM   R4   water content at root surface                 -       I       *
*  RUPW    R4   Rate of uptake of water                      cm/day O        *
*  CUPW    R4   Cumulative water uptake                       cm      O       *
*  R            As RUPW and CUPW but calculated with         cm/day  O       *
*  CUPWSR R4    steady-rate approximation                     cm      O       *
*  BAL     R4   Balance of water in soil                      cm      O       *
*  WCAPP   R4   Average water content calculated by steady-   -       O       *
*               rate approximation                                            *
*  WCAV    R4   Average water content                         -       O       *
*                                                                             *
*  Subroutines called: none.                                                  *
*                                                                             *
*  Functions called:                                                          *
*     FLPL: calculates matric flux potential as a function of                 *
*           water content                                                     *
*                                                                             *
*-----------------------------------------------------------------------------*

      SUBROUTINE UPDAT(IR,WC,WCN,WCLIM,R,RO,R1,HSC,DELT,T,
     1                 DWCDT,RUPW,RUPWSR,CUPWSR,CUPW,BAL,WCAV,WCAPP)

      DIMENSION WC(30),WCN(30),DWCDT(30),R(30),USTR(30)
      LOGICAL INIT
      DATA INIT/.FALSE./

      IF(.NOT.INIT) THEN
         RHO=R1/RO
         ETA=HSC/RO
         PI=4.*ATAN(1.)
         INIT=.TRUE.
         SUMWI=(R1*R1-RO*RO)*HSC*WC(2)/R1**2
      ENDIF
```

```
*------------------- calculation of DWC/DT
      DO 10 K=1,IR
         DWCDT(K)=(WCN(K)-WC(K))/DELT
10    CONTINUE

*------------------- SUMW total water in soil cylinder cm
*------------------- RUPW rate of decrease of water cm/day
      SUMW=0.
      RUPW=0.
      DO 20 K=1,IR-1
         SUMW=SUMW+
     1         (R(K+1)-R(K))*(R(K+1)*WC(K+1)+R(K)*WC(K))/2.
         RUPW=RUPW+
     1          (R(K+1)-R(K))*(R(K+1)*DWCDT(K+1)+R(K)*DWCDT(K))/2.
20    CONTINUE
      SUMW=2.*HSC*RO**2*SUMW/R1**2
      RUPW=-2.*HSC*RUPW/R1**2
      BAL=SUMW+CUPW-SUMWI

*------------------- WCAV average water content
      WCAV=SUMW*R1**2/(HSC*(R1**2-RO**2))
      USTR(1)=FLPL(WC(1))
      UAVAPP=FLPL(WCAPP)
*------------------- RUPWSR water uptake according to steady-rate
*------------------------- approximation cm/day
      RUPWSR=2.*HSC/R1**2*(USTR(1)-UAVAPP)*(RHO**2-1.)/(2.*FG(RHO))

*-------------------
      IF(WCN(1).LE.WCLIM.AND. WC(1).GT.WCLIM)THEN
         DELT=AMIN1(DELT,(WCLIM-WC(1))/DWCDT(1))
         TRPAR=-1.
      ENDIF

*------------------- Updating of WC
      DO 30 K=1,IR
         WC(K)=WC(K)+DWCDT(K)*DELT
         TPAR=1.
         IF(WC(K).LT.WCLIM)THEN
            TRPAR=-1.
         ENDIF
30    CONTINUE

      T=T+DELT

      CUPW=CUPW+RUPW*DELT*RO**2
      CUPWSR=CUPWSR+RUPWSR*DELT*RO**2
      WCAPP=WCAPP-RUPWSR*DELT*RO**2*R1**2/(HSC*(R1**2-RO**2))
      RETURN
      END
```

```
*---------------------------------------------------------------------------*
* Subroutine GEOMR                                                          *
* Purpose: calculation of position of gridpoints both in transformed and   *
*          untransformed coordinates                                        *
* Note: Gridpoints equidistant in transformed region, distances between    *
* gridpoints geometrically increasing in untransformed cylindrical region.*
*                                                                           *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)         *
*  name    type meaning                                     units  class   *
*  ----    ---- -------                                     -----  -----   *
*  IR      I4   Number of gridpoints                          -      I     *
*  RHO     R4   Dimensionless radius soil cylinder            -      I     *
*  R       R4   Radial coordinate gridpoint                   -      O     *
*  X       R4   Coordinate of gridpoint in transformed region: -     O     *
*               X = Ln(R)                                                   *
*  DX      R4   Distance between gridpoints in transformed region          *
*                                                                           *
*  Subroutine/Finstions called: none                                        *
*                                                                           *
*  File usage: Coordinates gridpoints wriiten to file soilname.PET          *
*                                                                           *
*---------------------------------------------------------------------------*

      SUBROUTINE GEOMR(IR,RHO,X,R,DX,DX2)
      DIMENSION X(30),R(30)

      DX=ALOG(RHO)/(IR-1)
      DX2=DX*DX
      R(1)=1.
      X(1)=0.
      DO 10 K=2,IR
         X(K)=X(K-1)+DX
         R(K)=EXP(X(K))
10    CONTINUE

      WRITE(41,100) IR,DR
      PRINT 100,IR,DR

      WRITE(41,101) (R(K),K=1,IR)
      PRINT 101,(R(K),K=1,IR)

      RETURN

100   FORMAT(1H0,I3,1X,'POINTS IN R-DIRECTION'/
     3/1X,'DR='1PE15.5)
101   FORMAT(1H0,'R='1P5E15.5)

      END
```

```
*-------------------------------------------------------------------------*
* Function: AVDIF                                                         *
* Purpose: calculates harmonic or arithmetic average of diffusivities,    *
*          with equal weights                                             *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)        *
*  name    type meaning                                    units  class  *
*  ----    ---- -------                                    -----  -----  *
*  X1,X2   R4   dummies, representing watercontent           -      I    *
*  MODE    L    If .true. harmonic average, else arithmetic               *
*                                                                         *
*  Subroutines called: none                                               *
*                                                                         *
*  Functions called:                                                      *
*     FDIF: calculates diffusivity as a function of water content         *
*                                                                         *
*  File usage: none                                                       *
*                                                                         *
*-------------------------------------------------------------------------*

      FUNCTION AVDIF(X1,X2,MODE)
      LOGICAL MODE
      D1=FDIF(X1)
      D2=FDIF(X2)
      IF(MODE) THEN
         AVDIF=2*D1*D2/(D1+D2)
      ELSE
           AVDIF=(D1+D2)/2.
      ENDIF
      RETURN
      END
```

```
*-----------------------------------------------------------------------------------*
* Subroutine COEFF                                                                  *
* Purpose: calculation of coefficients of the discretization equations.            *
*          These are of the form:                                                   *
*          A(I)*T(I)=B(I)*T(I+1) + C(I)*T(I-1) + D(I), where T(I) is the            *
*          value of the dependent variable at gridpoint I.                         *
*                                                                                   *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)                  *
*  name      type meaning                                      units  class         *
*  ----      ---- -------                                      -----  -----         *
*   N        I4   Number of gridpoints                           -      I           *
*  A,B,C,D   R4   Coefficients                                          O           *
*   T        R4   Dummy denoting the dependent variable                             *
*  FM        R4   If FM=1 scheme completely implicit, FM=0              I           *
*                 scheme completely explicit                                        *
*  DX        R4   distance between gridpoints                    -      I           *
*  RHO       R4   dimensionless radius soil cylinder             -      I           *
*  DT        R4   timestep                                              I           *
*  Q         R4   uptake rate                                                       *
*  FLB       R4   Parameter that controls boundary condition            I           *
*                 at root surface. FLB>0 constant flux,                             *
*                 FLB<0 constant watercontent                                       *
*  WCLIM     R4   water content at root surface                   -     I           *
*  TO        R4   Values of T at preceding time-step                    I           *
*  RELF      R4   Relaxation factor                               -     I           *
*  IT        R4   Controls relaxation, if IT >1 relaxation        -     I           *
*                                                                                   *
*  Subroutines/functions called: none                                               *
*                                                                                   *
*  File usage: none                                                                 *
*                                                                                   *
*-----------------------------------------------------------------------------------*

      SUBROUTINE COEFF(N,A,B,C,D,T,DIF,FM,DX,RHO,DT,Q,FLB,WCLIM,
     1                            TO,RELF,IT)
      DIMENSION A(30),B(30),C(30),D(30),T(30),TO(30),DIF(30)
      EMFM=1.-FM
      BETA=(EXP(DX)-EXP(-DX))/2.

      DO 5 I=2,N-1
         AE=DIF(I)/DX
         AW=DIF(I-1)/DX
         AOP=BETA*EXP(2.*(FLOAT(I)-1.)*DX)/DT
         AI=(FM*AE+FM*AW+AOP)

         B(I)=FM*AE
         C(I)=FM*AW
         DI=AE*EMFM*T(I+1) + AW*EMFM*T(I-1) +
     1                  (AOP-EMFM*AE-EMFM*AW)*T(I)
```

```
         IF(IT.LE.1)THEN
            A(I)=AI
            D(I)=DI
         ELSE
            A(I)=AI/RELF
            D(I)=DI+(1.-RELF)*A(I)*TO(I)
         ENDIF

5     CONTINUE

      IF(FLB.GT.0.)THEN
         A1=((EXP(DX)-1.)/(2.*DT)+DIF(1)*FM/DX)
         B(1)=DIF(1)*FM/DX
         C(1)=0.
         D1=DIF(1)*EMFM*(T(2)-T(1))/DX - Q +
     1                   (EXP(DX)-1.)/(2.*DT)*T(1)

         IF(IT.LE.1)THEN
            A(1)=A1
            D(1)=D1
         ELSE
            A(1)=A1/RELF
            D(1)=D1+(1.-RELF)*A(1)*TO(1)
         ENDIF

      ELSE
         A(1)=1.
         B(1)=0.
         C(1)=0.
         D(1)=WCLIM
      ENDIF

      AN=((RHO**2-EXP(2.*FLOAT(N-1)*DX-DX))/(2.*DT)+
     1            DIF(N-1)*FM/DX)
      B(N)=0.
      C(N)=DIF(N-1)*FM/DX
      DN=-DIF(N-1)*EMFM*(T(N)-T(N-1))/DX +
     1   (RHO**2-EXP(2.*FLOAT(N-1)*DX-DX))/(2.*DT)*T(N)

      IF(IT.LE.1)THEN
          A(N)=AN
          D(N)=DN
      ELSE
          A(N)=AN/RELF
          D(N)=DN+(1.-RELF)*A(N)*TO(N)
      ENDIF

      RETURN
      END
```

```
*------------------------------------------------------------------------*
* Subroutine TDMA                                                        *
* Purpose: solves the discretization equations                           *
*                                                                        *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)       *
*  name    type meaning                                   units  class   *
*  ----    ---- -------                                   -----  -----   *
*   N       I4  Number of gridpoints                        -      I     *
*  A,B,C,D R4   Coefficients                                       I     *
*   TE      R4  Dummy denoting the dependent variable              O     *
*                                                                        *
*  Subroutines/Functions called: none                                    *
*                                                                        *
*  File usage: none                                                      *
*                                                                        *
*------------------------------------------------------------------------*

      SUBROUTINE TDMA(N,A,B,C,D,TE)
      DIMENSION A(30),B(30),C(30),D(30),TE(30)
      REAL P(50),Q(50)
      P(1)=B(1)/A(1)
      Q(1)=D(1)/A(1)
      DO 5 I=2,N
         DENOM=A(I)-C(I)*P(I-1)
         P(I)=B(I)/DENOM
         Q(I)=(D(I)+C(I)*Q(I-1))/DENOM
5     CONTINUE
      TE(N)=Q(N)
      DO 10 I=N-1,1,-1
         TE(I)=P(I)*TE(I+1)+Q(I)
10    CONTINUE
      RETURN
      END
```

```
*----------------------------------------------------------------------*
* Subroutine OUTP                                                      *
* Purpose: organizes output                                            *
*                                                                      *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)     *
*  name    type meaning                                units    class  *
*  ----    ---- -------                                -----    -----  *
*  IR      I4   Number of gridpoints                    -        I     *
*  R       R4   Dimensionless radial coordinate         -        I     *
*  RO      R4   Radius root                             cm       I     *
*  HSC     R4   Root length                             cm       I     *
*  R1      R4   Radius soil cylinder                    cm       I     *
*  T       R4   Time in scaled units                    day/cm2  I     *
*  TIME    R4   Time in absolute units                  day      I     *
*  DELT    R4   Time step in scaled units               day/cm2  I     *
*  WC      R4   Water content                           -        I     *
*  WCN     R4   New values of water content             -        I     *
*  DWCDT   R4   Rate of change of water content         cm2/day  I     *
*  BAL     R4   Balance of water in soil                cm       I     *
*  RUPW    R4   Rate of uptake of water                 cm/day   I     *
*  CUPW    R4   Cumulative water uptake                 cm       I     *
*  RUPWSR  R4   As RUPW and CUPW but calculated with    cm/day   I     *
*  CUPWSR  R4   steady-rate approximation               cm       I     *
*  WCAV    R4   Average water content                   -        I     *
*  WCAPP   R4   Average water content calculated by steady-  -   I     *
*               rate approximation                                     *
*                                                                      *
*  Subroutines called: none                                            *
*                                                                      *
*  Functions called:                                                   *
*     FLPL: calculates matric flux potential as a function of          *
*             water content                                            *
*                                                                      *
*  File usage:                                                         *
*     detailed output written to unit 41, soilname.PET                 *
*       Plot-output written to soilname.PLO                            *
*                                                                      *
*----------------------------------------------------------------------*

      SUBROUTINE OUTP(IR,R,RO,HSC,R1,T,TIME,DELT,WCN,WC,DWCDT,DIFD,
     1                 BAL,RUPW,RUPWSR,CUPW,CUPWSR,WCAV,WCAPP)

      DIMENSION WCN(30),WC(30),DWCDT(30),DIFD(30)
      DIMENSION USTR(30),WCSTR(30),U(30),R(30)
      REAL KS,M,N,L

      RHO=R1/RO
      ETA=HSC/RO

*------------------ U(I) matric flux potential
      DO 5 I=1,IR
5     U(I)=FLPL(WC(I))
```

```
      USUM=0.
      DO 10 K=1,IR-1
         USUM=USUM+
     1            (R(K+1)-R(K))*(R(K+1)*U(K+1)+R(K)*U(K))/2.
10    CONTINUE
      QT=RHO**2/(2.*ETA)*RUPW*RO

      UAVAPP=FLPL(WCAPP)
      USUM=2.*HSC*RO**2*USUM/R1**2
      UAV=USUM*R1**2/(HSC*(R1**2-RO**2))
      USTR(1)=FLPL(WC(1))

*------------------- USTR(I) matric flux potential calculated with
*------------------- steady-rate approximation
      DO 20 K=2,IR
         USTR(K)=USTR(1)+QT*
     1                        ((R(K)**2-1.)/(2.*(RHO**2-1.))-
     2                        RHO**2*ALOG(R(K))/(RHO**2-1.))
20    CONTINUE

      IF(TIME.EQ.0.)THEN
         WRITE(48,100)
      ENDIF

      WRITE(41,101) T,TIME,DELT,BAL,WCAV,WCAPP
      PRINT 101,T,TIME,DELT,SUMW,BAL,WCAV,WCAPP
      WRITE(41,102) (R(K),WC(K),DWCDT(K),WCN(K),
     1                 DIFD(K),K=1,IR)
      WRITE(41,103) (R(K),U(K),USTR(K),K=1,IR)
      WRITE(41,*)
      WRITE(41,*) 'UAV = ',UAV,' UAVAPP = ',UAVAPP
      WRITE (41,*) 'CUPW = ',CUPW,' CUPWSR = ',CUPWSR
      WRITE(48,104) TIME,RUPW,RUPWSR,CUPW,CUPWSR
      RETURN

100   FORMAT(1X,11X,'TIME',10X,'RUPW',10X,'RUPWSR',5X,'CUPW',
     1                 4X,'CUPWSR')
101   FORMAT(//10X,'T='1PE15.5,2X,'REAL TIME='1PE15.5,
     1         1X,'DELT='1PE15.5/
     2         1X,'BAL='1PE15.5,
     3        /1X,'WCAV ='1PE15.5,1X,'WCAPP ='1PE15.5)
102   FORMAT(/
     1         1X,14X,'R',13X,'WC',10X,'DWCDT',12X,'WCN'
     2        ,11X,'DIFD'/
     3        (1X,1P5E15.5))
103   FORMAT(/1X,14X,'R',14X,'U',11X,'USTR'/
     1                 (1X,1P3E15.5))
104   FORMAT(1X,1P3E15.5,2F10.2)
      END
```

Appendix III. Calculation of uptake of a nutrient by a root system.

This calculation is performed by the subroutine UPTA. As input it needs the average root radius, the nitrogen demand of the crop, and the distribution with depth of root length density, water content, water uptake, concentration of the nutrient.

In the DO-loop 10 the potential uptake PUPT in mg/(cm^3 soil. day), i.e. uptake in the zero-sink situation, is calculated.
It is calculated from (7) as:

$$\pi L_{rv} D \frac{(\rho^2-1)}{G(\rho,\nu)} \bar{C} \ ,$$

where L_{rv} is the root length density in cm/cm^3.

Prior to the actual iteration FLAG(I), a marker which indicates if uptake from a layer can proceed as required, is in DO-loop 15 set equal to 0, for all I. The uptake from each soil layer, as described in the main text section 3.1, is performed in the DO-loop 30. The DO 30 starts with calculation of the required uptake per cm root.

In the DO 40, nested within DO 30, the actual uptake is set equal to the required uptake, if the latter is less than the potential uptake, if not actual uptake is set equal to potential uptake and FLAG for this layer is set equal to 1. Next it is investigated if the uptake does not exceed a maximum value imposed upon it by on the one hand the amount present in the layer, on the other hand the maximum flux over the root. Finally the N-requirement is updated by subtracting from it the calculated uptake if FLAG = 1, because in that case the roots take up at maximum rate. Also the total root length is updated by subtracting the root length in the layer if FLAG =1.

In the last part of DO 30 it is investigated if the iteration can be ended, either because total uptake rate equals N-demand, or all roots take up at maximum rate.

```
*------------------------------------------------------------------------------*
* Subroutine UPTA                                                              *
* Purpose: calculation uptake of linearly adsorbed nutrient by a root          *
*          system. Roots are uniformly distributed within a layer, but         *
*          vertically root distribution is non-uniform.                        *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)             *
*  name   type meaning                                      units   class      *
*  ----   ---- -------                                      -----   -----      *
*  RO     R4   radius of the root                           cm        I        *
*  LRV    R4   Root density                                 cm/cm3    I        *
*  DIF    R4   diffusion coefficient                        cm2/day   I        *
*  WCACT  R4   water content                                ml/cm3    I        *
*  THCKN  R4   thickness layers                             cm        I        *
*  IN     R4   number of rooted layers                      -         I        *
*  CONC   R4   concentration of nutrient in soil solution   mg/ml     I        *
*  TRR    R4   water uptake by the roots from a layer       cm/day    I        *
*  NDEM   R4   nitrogen demand                          kg/(ha.day)   I        *
*  NUPTR  R4   realized total uptake rate               kg/(ha.day)   O        *
*  FMAX   R4   maximum flux through root surface                               *
*  DELT   R4   timestep                                     day       T        *
*  NUPTR  R4   realized total uptake rate               kg/(ha.day)   O        *
*  UP     R4   Uptake from soil layer                   mg/(cm3.day)  O        *
*                                                    or kg/(ha.day)            *
*  PUP    R4   potential uptake from soil layer         kg/(ha.day)   O        *
*                                                                              *
*  Functions called:                                                           *
*     GRNU: calculates G-function                                              *
*------------------------------------------------------------------------------*

      SUBROUTINE UPTA(RO,LRV,DIF,
     1      WCACT,THCKN,IN,CONC,
     2      TRR,NDEM,FMAX,DELT,NUPTR,UP,PUP)


      IMPLICIT REAL (A-H,J-Z)
      IMPLICIT INTEGER (I)
      DIMENSION WCACT(10),THCKN(10),TRR(10),NCON(10)
      DIMENSION NU(10),LRV(10),DIF(10), SAMPD(10)
      DIMENSION BUF(10),CONC(10),FLAG(10),UP(10)
      DIMENSION MXUPT(10),G(10),RHO(10),PUP(10),PUPT(10)

*------     Specific surface root (root surface per unit root length)
      PI=4.*ATAN(1.)
      SPSUR=2.*PI*RO

*------      Calculation of total root length in cm/(cm2 soil surface)
      TOTRL=0.
      DO 5 I=1,IN
         TOTRL=TOTRL+LRV(I)*THCKN(I)
5     CONTINUE
      TOTRU=TOTRL
```

```
*------     Calculation dimensionless massflow (NU), radius soil cylinder (RHO),
*                    potential uptake PUPT
      DO 10 I=1,IN
         NU(I)=-TRR(I)*0.5/(2.*PI*THCKN(I)*DIF(I)*LRV(I))
         RDE=LRV(I)
         RHO(I)=1./(RO*SQRT(PI*RDE))
         TNU=2.*NU(I)
         G(I)=GRNU(TNU,RHO(I))
         PUPT(I)=LRV(I)*PI*(RHO(I)**2-1.)*CONC(I)/G(I)*DIF(I)
10    CONTINUE

*------ Putting iteration flag to zero
      DO 15 I=1,IN
         FLAG(I)=0.
15    CONTINUE

      IF(TOTRU.EQ.0.)GO TO 20
*------- NREQ required uptake in mg/(cm2.day)
      NREQ=NDEM/100.
      TUP=0.
      DO 30 IJK=1,100
*---------- RUPTM required uptake per cm root mg/(cm.day)
         RUPTM=NREQ/TOTRU
            DO 40 I=1,IN
               IF(FLAG(I).GT.0.)GO TO 40
               IF(PUPT(I).GT.(LRV(I)*RUPTM))THEN
                  UP(I)=RUPTM*LRV(I)
               ELSE
                  UP(I)=PUPT(I)
                  FLAG(I)=1.
               ENDIF

*------------------ MXUPT is least upper bound on uptake , maximum uptake
*------------------ is bounded by the amount of nutrient present, and the
*------------------ maximum flux through the root surface.
               MXUPT(I)=AMIN1(WCACT(I)*CONC(I)/DELT,FMAX*LRV(I)
     1                             *SPSUR)
               IF(UP(I).GT.MXUPT(I))THEN
                  UP(I)=MXUPT(I)
                  FLAG(I)=1.
               ENDIF

               IF(CONC(I).LE.0.)THEN
                  UP(I)=0.
                  FLAG(I)=1.
               ENDIF

*------------------ Remaining part of N-demand and total root length
*------------------ are calculated
               NREQ=NREQ-THCKN(I)*UP(I)*FLAG(I)
               TOTRU=TOTRU-LRV(I)*THCKN(I)*FLAG(I)
40          CONTINUE
```

```
         TUP=0.
         DO 45 I=1,IN
            TUP=TUP+THCKN(I)*UP(I)
45       CONTINUE
         CRIT=ABS((NDEM-100.*TUP)/NDEM)
         IF(CRIT.LE.1.E-5.OR.TOTRU.LE.1.E-4)GO TO 20
30    CONTINUE
20    CONTINUE

*---------------- conversion of mg/(cm3.day) to kg/(ha.day)
      NUPTR=0.
      DO 50 I=1,IN
         UP(I)=100.*THCKN(I)*UP(I)
         PUP(I)=100.*THCKN(I)*PUPT(I)
         NUPTR=NUPTR+UP(I)
50    CONTINUE
      RETURN
      END
```

APPENDIX IV. Calculation of water uptake by a root system.

The relation between the subroutines is shown in figure AIV-1. The calculation is performed by the routine TRANSP. As input it needs information about the number of (rooted) layers, their thickness and water content, the distribution with depth of root length density, the potential transpiration, the root conductivity, and the hydraulic properties of the layers. The latter include a pair of parametervalues for the calculation of the matric flux potential Φ. This calculation is based on a empirical relation between Φ and watercontent as found by Ten Berge (1986):

$$\Phi = \frac{Ax}{x+B} \qquad \text{(AIV-1)}$$

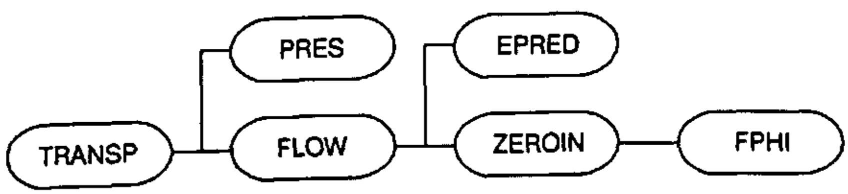
where $x = 1 - \theta/\theta_{sat}$.
Instead of (AIV-1) of course other approximations to (8) can be used, or eventually interpolation in a table.

In TRANSP equation (15) from section 3.2 is solved for plant water potential P_p. This is done by iteration by the method of bisection during the course of which (15) written in the form:

$$H(P_p) = F_1 - E_{act} \qquad \text{(AIV-2)}$$

where F_1, which stands for the left hand side of (15), has to be evaluated for different values of P_p. To do this however one must also solve the equations (14). This is achieved, again by iteration, in the subroutine FLOW, in doing so FLOW calls the function ZEROIN which finds the zero of a function in an interval. ZEROIN is given in Forsythe et al. (1977). The function EPRED calculates the actual transpiration as a function of plant water potential and potential transpiration.

Relational diagram of subroutines in TRANSP



```
*------------------------------------------------------------------------*
*  Subroutine: TRANSP                                                    *
*  Purpose : calculation of transpiration rate, water-uptake rate, and   *
*            plant water-potential, as function of soil water potential  *
*  FORMAL PARAMETERS:  (I-input,O-output,C-control,IN-init,T-time)       *
*  name    type meaning                                   units   class *
*  ----    ---- -------                                   -----   ----- *
*  DAY     R4   daynumber                                   -        I   *
*  INLAY   I4   number of soil layers                       -        I   *
*  WC      R4   watercontent                                ml/cm3   I   *
*  DX      R4   thickness layers                            cm       I   *
*  EPOT    R4   Potential transpiration                     cm/day   I   *
*  WCR     R4   residual watercontent                       ml/cm3   I   *
*  WCSAT   R4   saturated watercontent                      ml/cm3   I   *
*  ALPH    R4   parameter from Van Genuchten equation       cm-1     I   *
*  K1      R4   root conductivity for watertransport        cm/day   I   *
*  N       R4     "       "        "         "              -        I   *
*  A       R4   parameter from Ten Berge equation           cm2/day  I   *
*  B       R4     "       "        "         "              -        I   *
*  NRT     R4   number of rooted layers                     -        I   *
*  RD      R4   root-density in layer                       cm/cm3   I   *
*  RO      R4   root radius                                 cm       I   *
*  REDPOT  R4   indicates if reduction on potential         -        I   *
*              transpiration due to plant water potential                *
*              should be taken into account.                             *
*  EACT    R4   actual transpiration                        cm/day   O   *
*  UPW     R4   uptake rate of water                        cm/day   O   *
*                                                                        *
*                                                                        *
* Local variables:                                                       *
*  name    type meaning                                     units        *
*  K2      R4   -K1*DPO                                     cm2/day      *
*  DPO     R4   Intercept of waterflow-pressure curve for   cm           *
*               a root (value is 0.5MPa= 500 cm)                         *
*  RSS     R4   Resistance in soil, driving force is difference day-1    *
*               in hydraulic head                                        *
*  RSR     R4   Resistance in root                          day-1        *
*  PP      R4   Plant water potential                       cm           *
*                                                                        *
* Subroutines called:                                                    *
*       FLOW: solves the steady-rate equation for the matric flux        *
*       potential at the root surface                                    *
*                                                                        *
* Functions called:                                                      *
*  PRES: Calculates the matric potential as a function of water          *
*              content                                                   *
*                                                                        *
* File usage: Unit 89                                                    *
*------------------------------------------------------------------------*

      SUBROUTINE TRANSP(DAY,INLAY,WC,DX,EPOT,WCR,WCSAT,ALPH,N,A,
     $     B,REDPOT,NRT,RD,RO,K1,EACT,UPW)
```

```
      REAL WC(INLAY),DX(INLAY)
      REAL WCSAT(INLAY),WCR(INLAY),ALPH(INLAY),N(INLAY)
      REAL A(INLAY),B(INLAY)

      REAL RSS(10),RSR(10),RUPW(10),PHIMAX(10)
      REAL P(10),S(10),UPW(10),R1(10),RHO(10)
      REAL PHI(10),PHIRZ(10),UPRL(10),RD(10)
      REAL V(10),PRZ(10),WCRZ(10),Q(10),K1,K2,LP

      LOGICAL INITIAL

      DATA INITIAL/.TRUE./
      DATA PLIM/1.6E4/
      DATA DPO/500./

*------------------ initialisation
      IF(.NOT.INITIAL)GO TO 8
      PI=4.*ATAN(1.)

      DO 5 I=1,INLAY
      PHIMAX(I)=A(I)/(1.+B(I))
5     CONTINUE

      INITIAL=.FALSE.
8     CONTINUE
*------------------

*------------------ calculation of root (Q(I),V(I),SV,SQ) - and soil (S(I),SS)
*------------------ resistance parameters, radius (R1(I)) of soil cylinder, and
*------------------ matric potential P(I)
      K2=-K1*DPO
      SV=0.
      SQ=0.
      SS=0.
      DO 10 I=1,NRT
         R1(I)=1./SQRT(PI*RD(I))
         RHO(I)=R1(I)/R0
         S(I)=DX(I)/R1(I)**2*(RHO(I)**2-1.)/G(RHO(I))
         SS=SS+S(I)
         Q(I)=RD(I)*K1*DX(I)
         V(I)=RD(I)*K2*DX(I)
         P(I)=PRES(WC(I),WCSAT(I),WCR(I),N(I),ALPH(I))
         SV=SV+V(I)
         SQ=SQ+Q(I)
10    CONTINUE
*------------------

*------------------ first estimate of plant water potential (PP) in cm
      PP=500.
      DO 15 I=1,NRT
         IF(PP.GT.P(I))PP=P(I)
15    CONTINUE
```

```
*-------------------
*------------------- two values of plant potential (PP1 resp. PP2) are
*------------------- sought, for which (F1-EACT) has opposite sign
      PP1=1.
      PP2=1.
      DO 20 ITEL=1,20
         CALL FLOW(INLAY,EPOT,WC,P,A,B,Q,V,S,PP,N,
     $                 PHIMAX,WCSAT,WCR,ALPH,
     $                 PHIRZ,EACT,F1,F2,REDPOT)
         IF(EACT.EQ.0.0.AND.ABS(F1-EACT).LE.0.001)GO TO 40
         IF((F1-EACT).LT.0.)THEN
            PP1=PP
            PP=3.*PP
         ELSE
            PP2=PP
            GO TO 25
         ENDIF
20    CONTINUE
25    CONTINUE
*------------------ done

*------------------ Now (F1-EACT)=0 is solved for PP, by method of bisection
      DO 30 IK =1,30
         PP=(PP1+PP2)/2.
         CALL FLOW(INLAY,EPOT,WC,P,A,B,Q,V,S,PP,N,
     $                 PHIMAX,WCSAT,WCR,ALPH,
     $                 PHIRZ,EACT,F1,F2,REDPOT)
         DENOM=EACT
         IF(EACT.EQ.0.)DENOM=1.
         IF(ABS(F1-EACT)/DENOM.LT..001)GO TO 40

         IF((F1-EACT).LT.0.)THEN
            PP1=PP
         ELSE
            PP2=PP
         ENDIF

30    CONTINUE
40    CONTINUE
*------------------ done
      SQP=0.
      DO 60 IX=1,NRT
         X=1.-WC(IX)/WCSAT(IX)
         PHI(IX)=A(IX)*X/(B(IX)+X)
         WCRZ(IX)=WCSAT(IX)+PHIRZ(IX)*B(IX)*WCSAT(IX)/(PHIRZ(IX)
     1              -A(IX))
         PRZ(IX)=PRES(WCRZ(IX),WCSAT(IX),WCR(IX),N(IX),ALPH(IX))
         SQP=SQP+Q(IX)*PRZ(IX)
         UPW(IX)=(Q(IX)*(PP-PRZ(IX))+V(IX))
         UPRL(IX)=UPW(IX)/(RD(IX)*DX(IX))
         RSS(IX)=ABS(P(IX)-PRZ(IX))/(UPW(IX))
```

```
          RSR(IX)=ABS(PP-PRZ(IX))/(UPW(IX))

          IF(EACT.NE.0.)THEN
             RUPW(IX)=UPW(IX)/(EACT)
          ELSE
             RUPW(IX)=0.
          ENDIF
60      CONTINUE

        WRITE (89,*)
        WRITE (89,*)
        WRITE (89,*) 'TIME = ',DAY
        WRITE(89,100) PP,EACT,EPOT,F1
100     FORMAT(1H0,'PP = '1PE15.5,/1X,'EACT ='1PE15.5/1X,
     $  'EPOT ='1PE15.5/1X,'F1 ='1PE15.5)
        WRITE(89,101) (I,WC(I),P(I),PHIRZ(I),PHI(I),UPW(I),I=1,NRT)
101     FORMAT(/1X,4X,'I',10X,'WC',11X,'P',8X,'PHIRZ',9X,'PHI',
     $         9X,'UPW'/
     $         (1X,I5,5E12.5))
        WRITE(89,102) (I,UPRL(I),RUPW(I),RSS(I),RSR(I),I=1,NRT)
102     FORMAT(/1X,4X,'I',11X,'UPRL',11X,'RUPW',12X,'RSS',12X,'RSR'/
     $       (1X,I5,4E15.5))

        END
```

```
*-----------------------------------------------------------------------*
* Subroutine FLOW                                                       *
* Purpose: finds the value of the matric flux potential at the root wall,*
*          for which the flow through the rhizosphere equals the flow over
*          the root wall, i. e. for each soil layer it solves:          *
*         Q(I)*(PP-PRZ(i))+V(I)-S(I)*(PHIRZ(I)-PHI(I)), or              *
*          FPHI= Q(I)*(PP-PRZ(i))+V(I)-S(I)*(PHIRZ(I)-PHI(I))=0         *
*                                                                       *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)      *
*  name    type meaning                                    units  class *
*  ----    ---- -------                                    -----  ----- *
* INLAY    I4   number of soil layers                      -        I   *
* WC       R4   watercontent in bulk soil                  ml/cm3   I   *
* DX       R4   thickness layers                           cm       I   *
* EPOT     R4   Potential transpiration                    cm/day   I   *
* A        R4   parameter from Ten Berge equation          cm2/day  I   *
* B        R4     "         "    "    "                    -        I   *
* P        R4   matric potential                           cm       I   *
* S        R4   parameter expressing the local resistance to cm-1   I   *
*               waterflow in the soil, based on steady-rate             *
*               profile in matric flux potential                        *
* Q        R4   Resistance to waterflow in the root        cm/day   I   *
* PP       R4   Plant water potential                      cm       I   *
* WCR      R4   residual watercontent                      ml/cm3   I   *
* WCSAT    R4   saturated watercontent                     ml/cm3   I   *
* ALPH     R4   parameter from Van Genuchten equation      cm-1     I   *
* REDPOT   R4   indicates if reduction on potential        -        I   *
*               transpiration due to plant water potential              *
*               should be taken into account.                           *
* PHIRZ    R4   matric flux potential at root surface      cm2/day  O   *
* EACT     R4   actual transpiration                       cm/day   O   *
* F1       R4   total water flow over surface root system  cm/day   O   *
* F2       R4   total water flow vrom bulk soil to roots   cm/day   O   *
*                                                                       *
* Subroutines called: none                                              *
* Functions called:                                                     *
*       EPRED: calculates reduction potential transpiration             *
*       FPHI: evaluates the function:                                   *
*            Q(I)*(PP-PRZ(i))+V(I)-S(I)*(PHIRZ(I)-PHI(I))               *
*       ZEROIN : finds zero of a function in a given interval           *
*                                                                       *
* File usage: none                                                      *
*                                                                       *
*-----------------------------------------------------------------------*
      SUBROUTINE FLOW(INLAY,EPOT,WC,P,A,B,Q,V,S,PP,N,
     $                  PHIMAX,WCSAT,WCR,ALPH,
     $                  PHIRZ,EACT,F1,F2,REDPOT)
      REAL Q(INLAY),V(INLAY),S(INLAY),ALPH(INLAY)
      REAL WC(INLAY),P(INLAY),PHIRZ(INLAY)
      REAL N(INLAY),PHIMAX(INLAY)
      REAL WCSAT(INLAY),WCR(INLAY),A(INLAY),B(INLAY)
      REAL PHI(10)
```

```
      COMMON /PH/ AA,BB,QQ,VV,PPHI,PPP,SS,WCS,WR,FN,ALP
      EXTERNAL FPHI

      SQP=0.
      F1=0.
      F2=0.

*------------------- potential transpiration EPOT is corrected for plant
*------------------- potential effects if REDPOT>0
      EACT=EPOT
      IF(REDPOT.GT.0.)EACT=EPRED(PP,EPOT)

*------------------- for each layer the function FPHI is solved by
*------------------- ZEROIN  to find, for given value of plant waterpotential,
*------------------- the matric flux potential in the rhizosphere (PHIRZ)
*------------------- for which FPHI = 0.
*------------------- FPHI = 0 implies that watertransport through the
*------------------- rhizosphere equals water transport over the root wall.

      DO 10 IX=1,INLAY
         X=1.-WC(IX)/WCSAT(IX)
         PHI(IX)=A(IX)*X/(B(IX)+X)
         P(IX)=PRES(WC(IX),WCSAT(IX),WCR(IX),N(IX),ALPH(IX))

*---------------------------- First two values of PHIRZ are sought for
*---------------------------- which FPHI>0, resp. <0
         PHIRZ1=PHI(IX)-0.5
         IF(PHIRZ1.LE.0.)THEN
            PHIRZ1=.01
         ELSE
         ENDIF
         WCRZ=WCSAT(IX)+PHIRZ1*B(IX)*WCSAT(IX)/(PHIRZ1-A(IX))
         PRZ=PRES(WCRZ,WCSAT(IX),WCR(IX),N(IX),ALPH(IX))
         AA=A(IX)
         BB=B(IX)
         QQ=Q(IX)
         VV=V(IX)
         PPHI=PHI(IX)
         PPP=PP
         SS=S(IX)
         WCS=WCSAT(IX)
         WR=WCR(IX)
         FN=N(IX)
         ALP=ALPH(IX)
         FPHI1=FPHI(PHIRZ1)

         PHIRZ2=PHI(IX)+0.2
         IF(PHIRZ2.GE.PHIMAX(IX))THEN
            PHIRZ2=PHIMAX(IX)
         ENDIF
         WCRZ=WCSAT(IX)+PHIRZ2*B(IX)*WCSAT(IX)/(PHIRZ2-A(IX))
         PRZ=PRES(WCRZ,WCSAT(IX),WCR(IX),N(IX),ALPH(IX))
```

```
      FPHI2=FPHI(PHIRZ2)
      IF(FPHI2.GT.0.)THEN
         PHIRZ2=PHIMAX(IX)
      ENDIF
      FPHI2=FPHI(PHIRZ2)
      IF((FPHI1*FPHI2).GT.0.)THEN
         PRINT *,'FPHI1.2 SAME SIGN'
         PRINT *,'PHIRZ1,PHIRZ2 ',PHIRZ1,PHIRZ2
         PRINT *,'PHI(IX) ',PHI(IX)
         PRINT *,'FPHI1,FPHI2 ',FPHI1,FPHI2
         PRINT *,'PHIMAX ',PHIMAX(IX)
9        PRINT *,'GIVE PHIRZS'
         READ *,PHIRZS
         WCRZ=WCSAT(IX)+PHIRZS*B(IX)*WCSAT(IX)/(PHIRZS-A(IX))
         PRZ=PRES(WCRZ,WCSAT(IX),WCR(IX),N(IX),ALPH(IX))
         FPHIRS=FPHI(PHIRZS)
         PRINT *,'FPHIRZS ',FPHIRS
         PRINT *,'ANOTHER GO? >0'
         READ *,ANGO
         IF(ANGO.GT.0.)GO TO 9
         PRINT *,'GIVE PHIRZ1,PHIRZ2 '
         READ *,PHIRZ1,PHIRZ2
      ELSE
      ENDIF
*---------------------------- Done, two values for which FPHI has
*---------------------------- opposite sign are PHIRZ1, and PHIRZ2

*------------------- Now FPHI=0 is solved for PHIRZ
      PHIRZ(IX)=ZEROIN(PHIRZ1,PHIRZ2,FPHI,.00001)
      FPHIRZ=FPHI(PHIRZ(IX))
      WCRZ=WCSAT(IX)+PHIRZ(IX)*B(IX)*WCSAT(IX)/(PHIRZ(IX)-A(IX))
      PRZ=PRES(WCRZ,WCSAT(IX),WCR(IX),N(IX),ALPH(IX))

*------------------- F1 = Total flow over root wall
      F1=F1+Q(IX)*(PP-PRZ)+V(IX)
*------------------- F2 = Total flow through rhizosphere
      F2=F2+S(IX)*(PHIRZ(IX)-PHI(IX))

10    CONTINUE
      RETURN
      END
```

```
*---------------------------------------------------------------------------*
* Function EPRED                                                            *
* Purpose: calculation of actual transpiration by reduction of potential   *
*          transpiration due to plant water potential PP                    *
*          For PP < 5000 cm, no reduction,                                  *
*          for 5000<PP<16000 reduction linearly interpolated between       *
*          (5000,1) and (16000,0.1)                                         *
*  FORMAL PARAMETERS:  (I=input,O=output,C=control,IN=init,T=time)          *
*  name    type meaning                                    units   class   *
*  ----    ---- -------                                    -----   -----   *
*  PP      R4   Plant water potential                      cm        I     *
*  EPOT    R4   Potential transpiration                    cm/day    I     *
*  EPRED   R4   Reduced transpiration                      cm/day    O     *
*---------------------------------------------------------------------------*

      FUNCTION EPRED(PP,EPOT)
      DATA P1/5000./
      DATA P2/16000./

      RED=1.
      IF(PP.GT.P1)THEN
         RED=(0.1-1.)/(P2-P1)*(PP-P1)+1.
      ELSE
      ENDIF
      IF(RED.LT.0.)RED=1.E-4
      EPRED=RED*EPOT
      RETURN
      END

*---------------------------------------------------------------------------*
* Function FPHI                                                             *
* Purpose :  FPHI evaluates the function:                                   *
*          FPHI= Q(I)*(PP-PRZ(i))+V(I)-S(I)*(PHIRZ(I)-PHI(I))               *
*          the zero of which is sought.                                     *
* Only input the matric flux potential at the root surface PHIRZ,          *
* other variables and parameters input via COMMON, as required by          *
* calling function ZEROIN.                                                  *
*---------------------------------------------------------------------------*

      FUNCTION FPHI(PHIRZ)
      COMMON /PH/ AA,BB,QQ,VV,PPHI,PPP,SS,WCS,WR,FN,ALP
      WCP=WCS+PHIRZ*BB*WCS/(PHIRZ-AA)
      PRZ=PRES(WCP,WCS,WR,FN,ALP)
      FPHI=QQ*PPP-QQ*PRZ+VV-SS*(PHIRZ-PPHI)
      RETURN
      END


      FUNCTION G(R)
      G=0.5*((1.-3.*R**2)/4. + R**4*ALOG(R)/(R**2-1.))
      RETURN
      END
```