On the sink term of root water extraction[1]

Xuejun Dong
Texas A&M AgriLife Research - Uvalde
Uvalde, TX 78801

e-mail: xuejun.dong@ag.tamu.edu

This supplementary file documents detailed information of the water uptake term as used in the extended Fortran program "ALS.f". It is not needed in order to run the programs of this Method paper, but some interesting details may be useful for those who are curious about how the water extraction term is specified in the model. To correctly apportion the potential transpiration (in mm/day) to different soil layers, we need to consider variations in (1) soil water potential, (2) plant root distribution and (3) plant rooting depth.

1. *Soil water reduction function.* Feddes et al. (1978) proposed to use a soil water reduction function to estimate actual water extraction by plant roots and this approach has been adopted by numerous authors. The reduction function $\alpha(h)$ may be defined as

$$\alpha(h) = \begin{cases} 0 & h < h_4 \text{ or } h \geq h_1 \\ \frac{h-h_4}{h_3-h_4} & h_4 \leq h < h_3 \\ 1 & h_3 \leq h < h_2 \\ \frac{h-h_1}{h_2-h_1} & h_2 \leq h < h_1, \end{cases} \tag{1}$$

where h_1 is the pressure head at anaerobiosis point (m), h_4 is the pressure head at permanent wilting point (m) and h_2 and h_3 mark the range of water potential for optimal root extraction. According to Li et al. (1999); Mathur and Rao (1999) and Yang et al. (2009), the values of h_1, h_2 and h_4 can be chosen as -0.1m, -0.25m and -150m, respectively. The value for h_3 (m) is estimated as a function of potential transpiration T_p (mm/day):

$$h_3 = \begin{cases} -11 & T_p < 1 \\ 1.5T_p - 12.5 & 1 \leq T_p \leq 5 \\ -5 & T_p > 5. \end{cases} \tag{2}$$

The graphic form of function $\alpha(h)$ can be found in Fig. 3 of Dong et al. (2010), which can be generated using a Minitab macro shown in Appendix 1 1.1[2].

2. *Root distribution function.* With a soil water reduction model (such as Equation 1), and the assumption that plant roots distribute evenly within the rooting depth $Z_r(t)$ at a given time, Feddes et al. (1978) proposed a linear extraction model

$$S(h, z, t) = \alpha(h)\, T_p(t)/Z_r(t), \tag{3}$$

[1] Created December 30, 2015 for **F1000 Research**

[2] A brief instruction of how to use a Minitab macro is provided in the beginning of this macro.

in which $T_p(t)$ is potential transpiration in mm/day, and the root extraction term $S(h,z,t)$ (1/day) is a function of soil water potential, soil vertical position and time. It refers to the volume of soil water extracted per unit soil volume per unit time and can be used as the sink term in Richards equation. However, numerous studies have shown that the distribution of plant roots in the soil profile is not uniform, but with more roots existing on the upper soil layers and less roots in the lower soil layers (Dahlman and Kucera, 1965; Coupland and Johnson, 1965). So, numerous authors developed linear or nonlinear root distribution functions to model the soil water extraction by plant roots (Prasad, 1988; Ojha and Rai, 1996; Mathur and Rao, 1999; Wu et al., 1999; Li et al., 2001; Yadav et al., 2009a). The general form is

$$S(h,z,t) = \alpha(h)\, g(z,t)\, T_p(t), \tag{4}$$

where $g(z,t)$ is the root distribution function that varies with depth and time, and other variables are the same as defined in Equation 3. This function $g(z,t)$ should have the unit of L^{-1} so that it can scale the potential transpiration $T_p(t)$ across soil profile and that the sink term $S(h,z,t)$ shall have the unit of T^{-1}. Of the various models known to me, I feel the one by Ojha and Rai (1996) is easier to use and can be adjusted to simulate different shapes of root vertical distributions, as has been shown by several authors (Prasad, 1988; Wu et al., 1999; Li et al., 2001; Ojha et al., 2009; Yadav et al., 2009a). With rooting depth $Z_r(t)$ given, the model only has one shape parameter β:

$$g(z,t) = \frac{\beta+1}{Z_r(t)}\big[1 - \frac{z}{Z_r(t)}\big]^{\beta}, \tag{5}$$

where β is a parameter and $Z_r(t)$ is the rooting depth at time t. Because $g(z,t)$ is just used to scale the potential transpiration, the integration within the rooted soil depth should be unity such that (Lai and Katul, 2000; Liu et al., 2005)

$$\int_0^{Z_r(t)} g(z,t) = 1, \tag{6}$$

which was shown in Dong et al. (2010). As a matter of fact, Equation 5, if normalized by the root depth at time t, $Z_r(t)$, is identical to a generalized root distribution function proposed by Zuo et al. (2013), which takes the form of

$$NRLD(Z_{rel}) = a(1 - Z_{rel})^{a-1}, \tag{7}$$

where Z_{rel} is relative root depth and a is a shape parameter.

The behavior of Equation 5 is shown in Figure 1 with a rooting depth of 2m and potential transpiration T_p of 3 mm/day (vertical axis is $g(z,t)T_p$). It has been shown that, with $\beta = 2$, the model can best describe data of wheat root distribution (Ojha et al., 2009). Note that, with $\beta = 0$, the model describes a constant maximum uptake (corresponding to constant root distribution of Feddes et al. (1978)). With $\beta = 1$ the model describes a linear root distribution of Prasad (1988). For other β values, the model has similar behaviors as several other nonlinear models (Wu

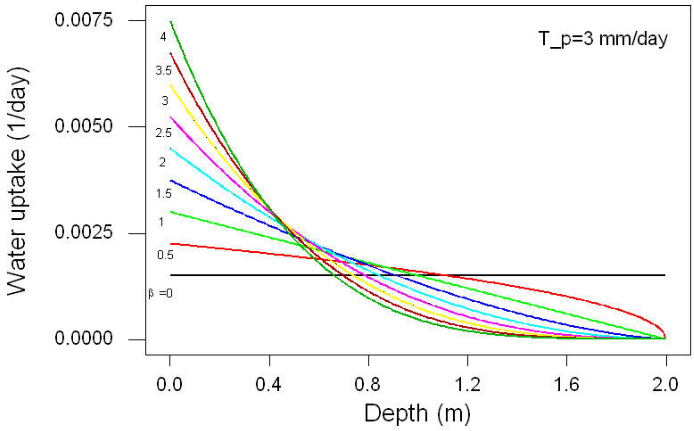

Figure 1: *A graphic display of the nonlinear root distribution function of Ojha and Rai (1996) with various values of β, assuming potential transpiration T_p is 3 mm/day.*

et al., 1999; Li et al., 2001; Yadav et al., 2009a). The MINITAB code for generating Figure 1 is listed in section 1.2.

The behavior of the soil water extraction term $S(h, z, t)$ in Equation is shown in Figure 2 and the corresponding MINITAB code is shown in section 1.3.

In the Richards equation with a sink term, the unit of $S(z, t)$ is T^{-1}. However, in actual numerical implementation (assuming 1-D situation), soil water extraction occurs at each of the segments Δz. For nonlinear root distribution models, such as the one by Ojha et al. (2009), the prescribed potential transpiration may not be accurately apportioned into each Δz of the segments if we use Equation 8 of Ojha and Rai (1996). This is shown in Figure 3, where we used the first method for computing S_{max} (see section 1.3). We can see that the sum of the total water extraction from all of the 100 segments is not equal to the applied T_p of 4.000 mm/day. This might have some impact on the computed water budget of the soil profile, but it appears that Ojha et al. (2009) used this approach in their paper.

An accurate apportioning of T_p to the soil segments during numerical implementation can be done if we use Equation 10 of Ojha and Rai (1996), which already takes into consideration the nonlinear change in root distribution function with soil depth by integrating the function over the distance of Δz:

$$W(t) = \frac{\tau Z_r(t)}{\beta + 1}\left[\left(1 - \frac{z_1}{Z_r(t)}\right)^{\beta+1} - \left(1 - \frac{z_2}{Z_r(t)}\right)^{\beta+1}\right], \tag{8}$$

where $\tau = T_p(t)(\beta + 1)/Z_r(t)$, with $T_p(t)$ the potential transpiration at time t and $Z_r(t)$ the rooting depth at time t. $W(t)$ is the amount of water extracted from the soil segment between depth z_2 and z_1, with $\Delta z = z_2 - z_1$, and $z_1 \leq z_2 \leq Z_r(t)$ at time t. The result using Equation 8 is shown in Figure 4. We can see that this time the summed transpiration from all layers (S_max) equals the applied p_t perfectly, while the actual transpiration under drier soil ($h = -30m$) is reduced to

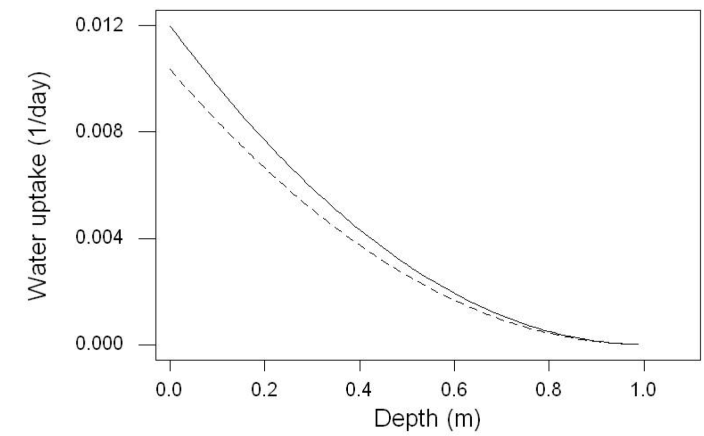

Figure 2: *An illustration of the nonlinear root water extraction function of Ojha and Rai (1996),* $S(h, z, t)$*, assuming potential transpiration* T_p *is 4 mm/day and* $\beta = 2$*. The solid line is for the maximum water uptake (1/day) and the dotted line is for the actual water uptake given a uniform soil water potential of -30 m. This is the result of the second method as discussed in the Minitab macro in Section 1.3.*

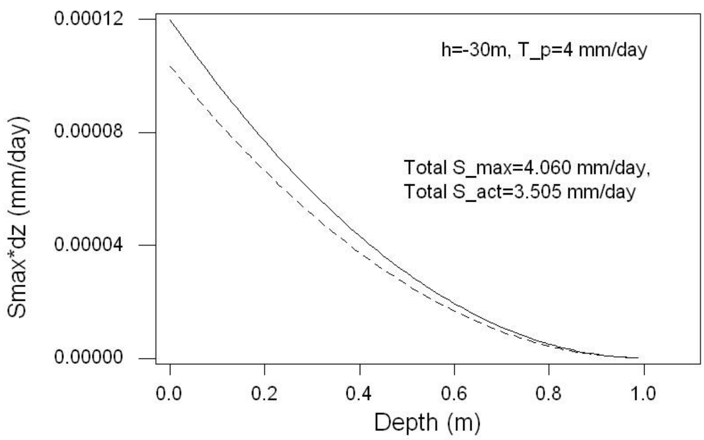

Figure 3: *Computing the root water extraction rate (mm/day) from each* Δz *segment for a soil profile evenly divided into 100 segments. The applied potential transpiration and other parameters are the same as in Figure 2. This is the result using the first method as discussed in the Minitab macro.*

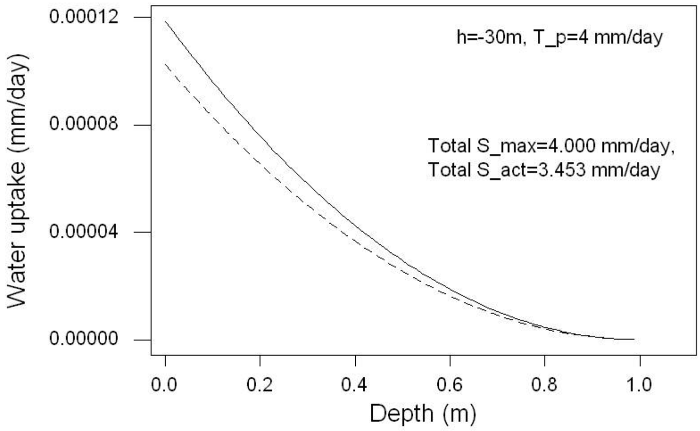

Figure 4: *Computing the root water extraction rate (mm/day) from each Δz segment for a soil profile evenly divided into 100 segments using the third method of section 1.3. The applied potential transpiration and other parameters are the same as in Figure 2, except that the third method was used in calculating the amount of water extraction.*

3.453 mm/day. As shown in Ojha et al. (2009), Equation 8 can be used to predict soil water depletion due to plant water use under various conditions, provided soil water potential is available.

3. *Seasonal change of rooting depth.* It is possible to relate the dynamic change of rooting depth with accumulated temperature (Yang et al., 2009). Also, several authors have used equations to describe rooting depth as a function of time during a growing season (Mathur and Rao, 1999; Yadav and Mathur, 2008; Yadav et al., 2009a). I feel this second approach is simpler to use. Yadav and Mathur (2008) used an equation by Gardner et al. (1985):

$$Z_r(t) = Z_m \left[\frac{1 + a_g}{a_g \left\{ 1 + a_g \, e^{\left(-b_g \frac{D_1}{D_2} \right)} \right\}} - \frac{1}{a_g} \right], \tag{9}$$

where Z_m is the maximum rooting depth, D_1 and D_2 are days after planting and days required for the maturity of the plant, and a_g and b_g are fitting parameters. The shape of Equation 9 is shown in Figure 5.

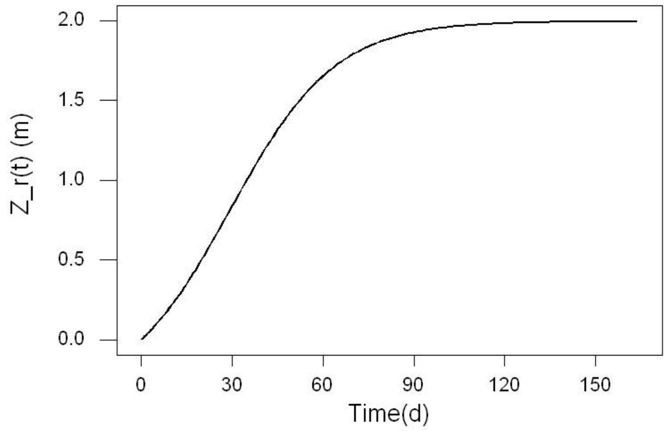

Figure 5: *Dynamic rooting depth growth as a function of time, with* $a_g = 5.372$ *and* $b_g = 6.105$. *Assume* D_1 *is set on April 15 and* D_2 *on July 15.*

References

Coupland, R. T., Johnson, R. E., 1965. Rooting characteristcs of native grassland species in Saskatchewan. J Ecol 53, 475–507.

Dahlman, R. C., Kucera, C. L., 1965. Root productivity and turnover in native prairie. Ecology 46, 84–89.

Dong, X., Patton, B. D., Nyren, A. C., Nyren, P. E., Prunty, L. D., 2010. Quantifying root water extraction by rangeland plants through soil water modeling. Plant Soil 335, 181–198.

Feddes, R. A., Kowalik, P. J., Zaradny, H., 1978. Simulation of field water use and crop yield. Tech. rep., Center for Agricultural Publishing and Documentation, Wageningen, The Netherlands.

Gardner, F. P., Pearce, R. B., Mitchell, R. L., 1985. Physiology of Crop Plants. The Iowa State University Press, Ames, Iowa.

Kirkland, M. R., Hills, R. G., Wierenga, P. J., 1992. Algorithms for solving Richards' equation for variably saturated soils. Wat Resources Res 28, 2049–2058.

Lai, C. T., Katul, G., 2000. The dynamic role of root water uptake in coupling potential to actual transpiration. Wat Resources Res 23, 427–439.

Li, K. Y., Boisvert, J. B., De Jong, R., 1999. An exponential root-water-uptake model. Can J Soil Sci 79, 333–343.

Li, K. Y., De Jong, R., Boisvert, J. B., 2001. An exponential root water uptate model with water stress compensation. J Hydrol 252, 189–204.

Liu, S., Graham, W. D., Jacobs, J. M., 2005. Daily potential evapotranspiration and diurnal climate forcings: influence on the numerical modelling of soil water dynamics and evapotranspiration. J Hydrol 309, 39–52.

Lorenz, R. J., Rogler, G. A., 1967. Grazing and fertilization affect root development of range grasses. J Range Manage 20, 129–132.

Mathur, S., Rao, S., 1999. Modeling water uptake by plant roots. J Irrig Drain Eng 125, 159–165.

Ojha, C. S. P., Hari Prasad, K. S., Shankar, V., Madramootoo, C. A., 2009. Evaluation of a nonlinear root-water uptake model. J Irrig Drain Eng 135, 303–312.

Ojha, C. S. P., Rai, A. K., 1996. Nonlinear root water uptake model. J Irrig Drain Eng 122, 198–201.

Prasad, R., 1988. A linear root water uptake model. J Hydrol 99, 297–306.

Wu, J., Zhang, R., Gui, S., 1999. Modeling soil water movement with water uptake by roots. Plant Soil 215, 7–17.

Yadav, B. K., Mathur, S., 2008. Modeling soil water uptake by plants using nonlinear dynamic root density distribution function. J Irrig Drain Eng 134, 430–436.

Yadav, B. K., Mathur, S., Siebel, M. A., 2009a. Soil moisture dynamics modeling considering the root compensation mechanism for water uptake by plants. J Hydrol Eng 14, 913–922.

Yang, D., Zhang, T., Zhang, K., Greenwood, D. J., Hammond, J. P., White, P. J., 2009. An easily implemented agro-hydrological procedure with dynamic root simulation for water transfer in the crop-soil system: validation and application. J Hydrol 370, 177–190.

Zuo, Q., Zhang, R., Shi, J. 2013. Characterization of root length density distribution of wheat sing a genearlized funciton. In: Timlin, D., Ahuja, L. R. (Eds.), Advances in Agricultural Sysmtes Modeling. Vol. 4. Enhancing understanding and chracteritization of soil-root growth interactions, ASA, CSSA, SSSA, Madison, WI, U.S.A., pp. 93–117.

1 Appendices

1.1 MINITAB macro to plot soil water reduction function

```
gmacro
alphah

####################################################################
#                How to use a Minitab macro
####################################################################
# (1) Assume a recent version of Minitab software is installed.
# A 30-day free trial version is downloadable from www.minitab.com.
# Save the macro file in Minitab's macro folder, which can be accessed at
# ...Program files\Minitab\Minitab 17\English\Macros\
#
# (2) Start Minitab and Click on "session window" icon, which is the
# window to accept user commands.
# In this session window, click on "Editor/Enable Commands",
# so the prompt "MTB>" is displayed.
#
# (3) Assume this macro is given a name "alphah.mac",
# we enter one line of command
#
# %'alphah'
#
# followed by Return. This will evoke the macro.
#####################################################################


# The purpose of this macro is to simulate the reduction
# function for root water extraction considering
# different potential transpiration (mm day-1).

# c33(scratch), c2-c7: array variables for the reduction
#  function
# c1    array variable of soil matric potential (m)
# c30   array variable to store different potential
# transpiration values (mm day-1)

# k10: scratch variable to store the length of P_t array
# k11: scratch variable to store the length of h array
#
# The following h values are based on
#  Yang:J2009 and Mathur:J1999, and Li:J1999.
# h1:  =-0.1, potential at anaerobiosis point (m)
# h2:  =-0.25,starting h for optimal water extraction (m)
# h3:  variable as a function of potential transpiration
```

```
#  with h3=1.5*p_t-12.5, for 1 <= p_t <= 5,
#  h3=-5, for p_t>5, and h3=-11, for p_t<1,
# where h3 is in m and p_t in mm day-1
# h4:  =-150, permanent wilting point (m)

#initializing the arrays
set c30      # potential transpiration values (mm day-1)
0.5 1 2.5 3.7 5 5.5
end
let k10=count(c30)

set c1      # soil matric potential (m)
-200:-0.1/0.1
end
let k11=count(c1)

let k1=-0.1   # anaerobiosis point
let k2=-0.25
let k4=-150    # permanent wilting point
let k5=2     # start recording alpha from c2 (through c7)

do k12=1:k10
  if c30(k12)<1
    let k3=-11
  elseif c30(k12)>5
    let k3=-5
  else
    let k3=1.5*c30(k12)-12.5
  endif

  do k13=1:k11
    if c1(k13)< k4 or c1(k13)>= k1
      let c33(k13)=0
    elseif c1(k13)< k3 and c1(k13)>= k4
      let c33(k13)=(c1(k13)-k4)/(k3-k4)
    elseif c1(k13)< k2 and c1(k13)>= k3
      let c33(k13)=1
    else
      let c33(k13)=(c1(k13)-k1)/(k2-k1)
    endif
  enddo
  copy c33 ck5
  erase c33
  let k5=k5+1
enddo
```

```
name c1 'h' c2 'alpha-0.5pt' c3 'alpha-1pt' c4 'alpha-2.5pt'
name c5 'alpha-3.7pt' c6 'alpha-5pt' c7 'alpha-5.5pt'

Plot c2*c1 c3*c1 c4*c1 c5*c1 c6*c1 c7*c1;
  Connect;
  Connect;
  Connect;
  Connect;
  Connect;
  Connect;
  Overlay;
  Minimum 1 -165;
  Maximum 1 0;
  Minimum 2 -0.1;
  Maximum 2 1.1;
  ScFrame;
  ScAnnotation;
  Axis 1;
  Axis 2;
  Tick 1;
    NMajor 6;
  Tick 2;
    NMajor 5.

erase c1-c50 k1-k24
endmacro
```

1.2 MINITAB macro for generating Figure 1

```
gmacro
or

let k1=2    # z_{rj}, root depth in m
let k2= 3       # T_j, potential transpiration of day j (mm/day)

let k2=k2/1000   # convet unit to m/day

set c1          # z, depth
0:2/0.01
end

let k10=count(c1)

let k4=2
do k5=0:4/0.5   # beta
do k3= 1:k10
  let ck4(k3)=k2/k1*(k5+1)*(1-c1(k3)/k1)**k5
```

```
enddo
let k4=k4+1
enddo

Plot C2*C1 C3*C1 C4*C1 C5*C1 C6*C1 C7*C1 C8*C1 C9*C1 C10*C1;
  Connect;
  Connect;
  Connect;
  Connect;
  Connect;
  Connect;
  Connect;
  Connect;
  Connect;
    color 1 2 3 4 5 6 7 8 9;
  Overlay;
  ScFrame;
  ScAnnotation;
  Axis 1;
    label "Depth (m)";
  Axis 2;
    label "Water uptake (day-1)";
  Tick 1;
    NMajor 6;
  Tick 2;
    NMajor 4.

erase k1-k10 c1-c10
endmacro
```

1.3 MINITAB macro for generating Figure 2

```
gmacro
ORtest2

# The default for this macro is for the second method.

# This macro is to test the use of the model of root water
# uptake by Ojha:J1996, assuming typical field
# situations that can be seen for a vegetated agricultural
# area. It uses three different methods to compute the
# maximum water extraction from different soil layers

# First method for S_max: To compute the amount of water
# extracted by root from each of the equally divided soil
# profile. It is eq. 8 of Ojha:J1996 times dz. So the unit
# is in mm day-1 and the integration over the whole profile
```

```
# is supposed to be equal to the prescribed potential
# transpiration of the day. But we will see there is some
# error (1%) in the summed potential transpiration, because
# it is assumed that water is extracted uniformly within the
# each dz.

# Second method of S_max: This is exactly eq. 8 of Ojha:J1996,
# in which, the unit is day-1: the volume of water
# absorbed by roots AT depth z from unit soil volume per day.
# However, in any numerical method for soil water flow,
# where the vertical axis is divided into numerous small
# segments, the amount of potential transpiration apportioned
# to one of the segments should be the amount of water
# obtained by integrating the S_max over dz.

# Third method for S_max: This will do a similar job as
# the first method, but here because an analytical solution
# of the integral (of S_max over dz) is used and the summed
# transpiration matches the prescribed one perfectly. So,
# this simple equation should be preferred over the first
# method in computing the amount of  potential transpiration
# to be apportioned to each of the z segments as a sink term.

############ Advantages of the model   ###################
#
# 1. It only has one parameter, beta, determining the shape of
# vertical distribution of root water uptake
# (due to variations in root density).
# 2. At root bottom (z=z_{rj}), the optimal uptake rate
# (without soil water limitation) takes zero value. However,
# this is not always ensured in some other models, such as
# the one in WuJQ:Opt1999.
# 3. The general behavior of the OR model is quite similar
# to some other common ones, such as the one in WuJQ:Opt1999.
############################################################

############ Definition of input parameters  ################
# There are eight parameters that can be changed to represent
# different situations (k1-k2,k4,k80-k90)
# k1:  anaerobiosis point of soil pressure head (m)
# k2:  starting point of optimal head for water uptake (m)
# k4:  permanent wilting point (m)
# k80: beta value
# k81: z_{rj}, rooting depth of the jth day (m)
# k82: T_j, potential transpiration of the jth day (mm d-1)
# k83: h, hypothetical pressure head for the soil profile (m)
```

```
# k87: the number of segments soil profile is divided
################ end of input parameters definition ##########

######################## How to use this model###############
# 1. To update the following eight parameters.
# 2. Read through the program and comment out the unwanted lines
# for S_max computation, but keep the lines for the interested
# method. Make sure for each macro run only one method is used.
############################################################

###### default values for the eight input parameters ########
let k1=-0.1
let k2=-0.25
let k4=-150
let k80=2
let k81=1
let k82=4
let k83=-30
let k87=100
##### end of default input parameter values  #################
#
#
let k82=k82/1000    # convert unit of t_p to m/day

############### generate c1 for first or second method ####
let k88=k81/k87         # dz (each of k87 z segments)
let k90=k81-k88
set c1          # soil depth profile
0:k90/k88
end
name c1 'Depth (m)'
# count(c1)=k87!
###### end of c1 generation for first/second method ######

################# generate c1 for the third method #######
#let k88=k81/k87         # dz (each of k87 z segments)
#let k90=k87+1           # length of c1
#set c1         # soil depth profile
#0:k81/k88
#end
#name c1 'Depth (m)'
# count(c1)=k90!
################# end of c1 for third method #############

set c2          # soil pressure head profile
k87(k83)
```

```
end
name c2 'Head (m)'
# count(c2)=k87!

#############  First method of S_max   ####################
# for use in soil water modeling where the S_{max} function
# of OR1996 must be integrated within the depth range of dz,
# or multiplied by the dz (with an error of 1% for 100
# divisions of 1 m rooting depth). However, the time (day-1)
# in p_t data can only be converted into the appropriate dt
# duration, assuming to spread evenly the whole day's
# potential transpiration in time. This will underesimate
# diurnal transpiration and over estiamte nocturnal
# transpiration, but may be able to keep an averagely
# accurate account of transpiration flux duirng a day.
#
# count(c1)=k87!
#do k71=1:k87            # S_{max}
#  let c3(k71)=k88*k82/k81*(k80+1)*(1-c1(k71)/k81)**k80
#enddo
#name c3 'Smax_d-1'
########## end of first method for S_max ##################

#############  Second method for S_max ####################
###### for use to plot OR's S_max (m day-1) function (with z)
# count(c1)=k87!
do k71=1:k87            # S_{max}
  let c3(k71)=k82/k81*(k80+1)*(1-c1(k71)/k81)**k80
enddo
name c3 'Smax_d-1'
######other wise comment out this part######################
########### end of second method ###########################

## This new program 'ORtst1' use the following third method##
## eq. 10 of Ojha:J1996
#let k101=k80+1     # beta+1
#let k102=k82/k81*(k80+1)   # alpha of OR:J1996
#do k71=1:k87
#  let k72=k71+1
#  let c3(k71)=k102*k81/k101*((1-c1(k71)/k81)**k101-(1-c1(k72)/k81)**k101)
#enddo
################ end of third method ####################

if k82<1
  let k3=-11        # set k3 (ending of optimal head)
elseif k82>5
```

```
  let k3=-5
else
  let k3=1.5*k82-12.5
endif

# now compute the reduction values for each of the k87 layers

do k13=1:k87
  if c2(k13)< k4 or c2(k13)>= k1
    let c33(k13)=0
  elseif c2(k13)< k3 and c2(k13)>= k4
    let c33(k13)=(c2(k13)-k4)/(k3-k4)
  elseif c2(k13)< k2 and c2(k13)>= k3
    let c33(k13)=1
  else
    let c33(k13)=(c2(k13)-k1)/(k2-k1)
  endif
enddo

# now compute the actual water extraction
# over one day from each of the k87 layers (with dz=k88)
# on jth day
let c4=c3*c33
name c4 'S_day-1'

# now plot the results

############### only used in third method ###########
# delete last element of c1
# so that is has the same length with c3 or c4
#delete k90 c1
####### otherwise, comment the above line out. ######

let k81=k81+c1(k87)*0.1     # just for plotting

Plot c3*c1 c4*c1;
  Connect;
  Connect;
  Overlay;
#  Minimum 2 0;
#  Maximum 2 0.00012;
  Minimum 1 -0.01;
  Maximum 1 k81;
  ScFrame;
  ScAnnotation;
  Axis 1;
```

```
    label "Depth (m)";
  Axis 2;
#    label "Water uptake (mm/day)";
    label "Water uptake (day-1)";
  Tick 1;
    NMajor 6;
  Tick 2;
    NMajor 4.

# Print the sums of c3 (S_max) and c4 (S_h).
# This is only useful if one uses the first or third method
# for computing S_max and intend to verify that
# the sum of transpiration from all layers equals the
# prescribed potential transpiration of the day.

let k95=sum(c3)*1000
name k95 'T_pot'
let k96=sum(c4)*1000
name k96 'T_act'

note Maximum & actual water uptate for day j (mm d-1) are:
print k95-k96

erase c1-c107 k1-k107
endmacro
```