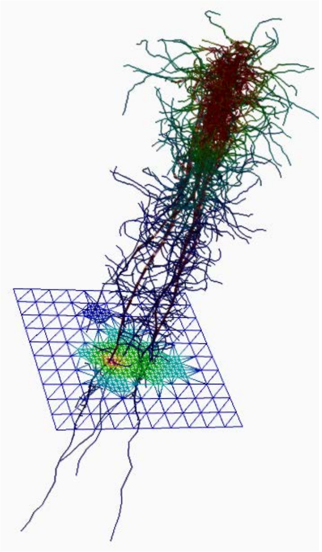
R-SWMS

Modeling Root-Soil Water Movement and Solute transport

A three-dimensional model for root growth, transient soil water flow, and solute transport and uptake



Release 9, September 2018

JÜLICH

FORSCHUNGSZENTRUM

Disclaimer

Contact j.vanderborght@fz-juelich.de for commercial licensing opportunities.

Table of Contents

Table of figures

Table of tables

Variables

Introduction

R-SWMS is a numerical model for simulating solute transport and water flow in and between the soil and the plant systems. The acronym stands for *Modeling "Root-Soil Water Movement and Solute transport"*. Based on the flow and transport equations in the 3D soil matrix and within the 3D root xylem network, it simulates the uptake of solute and water by plant roots for a growing plant. Three-dimensional root growth is function of environmental conditions (soil strength, temperature) and plant parameters (gravitropism, sensitivity to strength, etc.).The code has been used in several projects and labs around the world. An updated list of publications dealing with R-SWMS can be found at https://www.zotero.org/groups/r-swms.

The code programming language is Fortran 90. It has originally been created from the model of Clausnitzer and Hopmans (1994), and thus originally contained similar water flow and solute transport routines as SWMS_3D(Simunek, Huang, and Van Genuchten 1995). The first version was developed between 2004 and 2006 by Mathieu Javaux (see Javaux et al.,2008)by coupling the Clausnitzer and Hopmans (1994) to subroutines solving water flow in roots (Doussan et al., 2006) and to root growth (Pages et al., 2004). Since then, it has been extended by several contributors through PhD and post-doc projects, amongst others:

- Tom Schroeder: PhD student at FZJ (2006-2009)
- Natalie Schroeder: PhD studentat FZJ/UCL (2010-2013)
- Valentin Couvreur : PhD student at UCL (2009-2013)
- KatrinHuber : PhD student and postdoc at FZJ (2011-)
- FélicienMeunier : PhD student at UCL (2013-)
- Asta Kunkel: PhD student at FZJ (2014-)
- Nimrod Schwartz : post-doc at UCL (2014-2016)
- Magdalena Landl: PhD student at FZJ (2014-)
- Helena Jorda Guerra : PhD student at KuLeuven/FZJ (2015-)

Jan Vanderborght, Andrea Schnepf, and Mathieu Javaux were the leading scientists for this manual.

In this manual, we first describe the main processes affecting solute transport and water flow in the soil and in the plant systems. Then, we describe the numerical equations that are actually solved in the code. Input and output files and how to visualize them is described in the last sections of the User guide.

Users' guide

1. Theory

In this chapter, we present the main equations representing processes taking place in soil and roots. The numerical implementation of these functions into the code is presented in Chapter 2.

1.1. Water flow in soil and roots

1.1.1. Soil water flow equations

Soil water flow is described using the 3-D Richards equation:

$$\frac{\partial\theta}{\partial t} = \nabla \cdot [\boldsymbol{K}\nabla(h+z)] - S \qquad (1)$$

where $\boldsymbol{\theta}$ is the volumetric water content, ***K*** the hydraulic conductivity tensor [L T^{-1}], *h* the matric head [L] and *z* [L] the elevation, which is defined positively upward, and *S* [T^{-1}] a sink/source term. The sink term *S* is positive for root water uptake and negative for water release.

The relations between water content (**θ**)and matric head (**h**) and hydraulic conductivity tensor (**K**) are described by the Mualem-van Genuchten functions(Mualem, 1976):

$$\theta(h) = \theta_r + \frac{\theta_s - \theta_r}{(1+(\alpha|h|)^n)^m} \qquad (2)$$

$$K(h) = K_s\{1-(\alpha|h|)^{n-1}[1+(\alpha|h|)^n]^{-m}\}^2[1+(\alpha|h|)^n]^{-lm} \qquad (3)$$

where $\boldsymbol{\theta_r}$[L^{3}L^{-3}] is the residual soil water content, $\boldsymbol{\theta_s}$[L^{3}L^{-3}]the saturated soil water content, $\boldsymbol{\alpha}$ [L^{-1}] and ***n*** are shape parameters, m = 1-1/n, $\boldsymbol{K_s}$ [L T^{-1}] is the saturated water conductivity, and ***l*** is the pore connectivity parameter.

As an alternative, Durner bimodal pore size distribution model (Durner, 1994) can be used to describe the hydraulic properties:

$$\theta(h) = \{\theta_r + (\theta_s - \theta_r)\big(w_1 S_{e1}(h) + w_2 S_{e2}(h)\big) \;\; at \quad h < 0 \qquad (4)$$

$$\theta_s \; at \;\; h \geq 0 \qquad (5)$$

$$S_{e,i}(h) = [1 + (\alpha_i|h|)^{n_i}]^{-m_i} \qquad (6)$$

$$K(\theta) = K_s \frac{(w_1 S_{e_1} + w_2 S_{e_2})^l \left(w_1\alpha_1\left[1-(1-S_{e_1}^{1/m_1})^{m_1}\right] + w_2\alpha_2\left[1-(1-S_{e_2}^{1/m_2})^{m_2}\right]\right)^2}{(w_1\alpha_1 + w_2\alpha_2)^2} \qquad (7)$$

where $S_{e,i}$ is the saturation degree of the ith pore size distribution,α_i (cm^{-1}) and n_i are shape parameters, $m_i = 1\text{-}1/n_i$, and w_i is the volume fraction of the ith pore size distribution with $w_1 + w_2 = 1$.To include spatial variations in soil hydraulic properties, scaling factors can be defined.

Commenté [MJ1]: To be checked whetehr it is still working

At the soil surface, a time variable matric head or time variable water fluxes can be defined. At the bottom of the soil column, a matric head, a zero matric head (**h**) gradient, a time variable water flux or a no flow boundary condition can be defined. At the lateral sides, a no-flow boundary condition or a periodic flux boundary condition can be defined:

$$\boldsymbol{K}\frac{\partial h}{\partial x}\bigg|_{(0,y,z)} = \boldsymbol{K}\frac{\partial h}{\partial x}\bigg|_{(L_x,y,z)} \text{ and } \boldsymbol{K}\frac{\partial h}{\partial x}\bigg|_{(x,0,z)} = \boldsymbol{K}\frac{\partial h}{\partial x}\bigg|_{(x,L_y,z)} \quad (8)$$

Where L_x and L_y are the widths of the domain in the x-direction and in the y-direction respectively.

1.1.2. Water flow in the root system

When osmotic potential is not considered, the radial flux density entering a root segment, $\boldsymbol{q_r}$ [$L^3L^{-2}\,T^{-1}$] is given by:

$$q_r = L_r\,[h_s - h_x] \quad (9)$$

where $\boldsymbol{L_r}$ [T^{-1}] is the radial conductivity of the root and $\boldsymbol{h_s}$ [L] and $\boldsymbol{h_x}$[L] are the pressure heads at the soil-root interface and in the xylem respectively.The axial volumetric flow along a root segment, $\boldsymbol{J_x}$[$L^3\,T^{-1}$] is described by:

$$J_x = -K_x\left[\frac{dh_x(z)}{dl_{seg}} + \frac{dz}{dl_{seg}}\right] \quad (10)$$

where $\boldsymbol{dl_{seg}}$ is the arc length [L] of a root segment and positive in the direction from the root base to the root tip, i.e. in the direction of root growth, and $\boldsymbol{K_x}$ is the xylem axial conductance [$L^3\,T^{-1}$]. A negative $\boldsymbol{J_x}$ implies then a flux towards the root collar.

The volume balance in the root system leads to the following equation, valid for any root segment ***dl*** with a diameter ***r*** [L]:

$$2\pi r q_r = \frac{dJ_x}{dl} \quad (11)$$

In the following, we define the radial volumetric flow of a root segment$\mathbf{J_r}$ [L^3T^{-1}] as:

$$J_r = 2\pi r q_r dl \quad (12)$$

Equation (12) can be written for each segment of a complex root system and solved provided that the root system architecture is known, that the soil water potential is given, and that boundary conditions are given at the root tips and the root collar.

The boundary condition at the root tips is no flux. At the root collar, the following boundary conditions can be defined as follows:

<u>Imposed flux</u>: a potential transpiration is imposed at the collar: $J_x = -T_{pot}$ [$L^3\,T^{-1}$] where $\mathbf{T_{pot}}$is the prescribed transpiration rate estimated based on climatic conditions. When the prescribe transpiration cannot be met due to low soil water potential (**ψ**), stress appears, which decreases the transpiration. We describe it as a isohydric stress process, in which plant tries to keep the water potential at the collar (or, by extension in the leaf) above a critical value $h_{x,lim}$. It writes:

if : $h_{leaf} \approx h_{x,0} > h_{x,lim} : J_{x,0} = -T_{pot}$ (13)

else: $h_{leaf} \approx h_{x,0} = h_{x,lim}$

Where $\boldsymbol{h_{x,0}}$ is the root water potential at the root collar. Note that the leaf water head $\boldsymbol{h_{leaf}}$ is not explicitly modeled and is assumed to be equal to the xylem water potential at the root collar.

Imposedpressurehead: The pressure head at the collar $\boldsymbol{h_{x,0}}$ can be defined as a constant or function of time.

Stomatalfunction: Actual transpiration ($\boldsymbol{T_{act}}$ [L^3 T^{-1}]) can be described as a function of the collar water pressure head and the hormone concentration (Tardieu and Davies, 1993):

$$T_{act} = T_{pot}\{\alpha_r + (1 - \alpha_r)\, exp[-s_c c_{leaf}\, exp(s_p|\psi_{leaf}|)]\} \tag{14}$$

where, $\boldsymbol{T_{pot}}$ [L^3 T^{-1}] is the potential transpiration, i.e. the transpiration when stomata are fully open, $\boldsymbol{\alpha_r}$ (-) is the residual stomatal aperture, $\boldsymbol{c_{leaf}}$ [N L^{-3}] is the chemical concentration in the leaves, $|\boldsymbol{\psi_{leaf}}|$ [P] is the absolute value of the leaf water potential, and $\boldsymbol{s_c}$ [L^3 N^{-1}] and $\boldsymbol{s_p}$ [P^{-1}] are fitting parameters.

1.1.3. Dealing with non-resolved processes/properties in the rhizosphere

Macroscopic solution of water flow in a hydraulic architecture(Couvreur et al., 2012).

In case the complete root architecture is not known, an alternative, simpler, representation of the root hydraulic architecture exists based on so-called emergent parameters.

The solution of water flow in root segments connected as a hydraulic architecture has a structure that can be described with plant-scale parameters and variables (Couvreur, et al., 2012).A first emergent relation connects the flux- and pressure-type collar boundary conditions as follows:

$$H_{collar} = H_{s,eq} - \frac{T_{act}}{K_{rs}} \tag{15}$$

where $\boldsymbol{H_{collar}}$ [L] is the plant collar xylem total water head, $\boldsymbol{T_{act}}$ is the actual transpiration stream [L^{3}T^{-1}],$\boldsymbol{K_{rs}}$ [L^3 L^{-1} T^{-1}] is the root system hydraulic conductance, and $\boldsymbol{H_{s,eq}}$[L] is the plant-sensed (or “equivalent”) total soil water potential calculated as follows:

$$H_{s,eq} = \sum_i H_{sr,i}.SUF_i \tag{16}$$

where $\boldsymbol{H_{sr,i}}$ [L] is the total water head at the *i*th soil-root interface, and SUF_i is the standard uptake fraction at the *i*th soil-root interface, which corresponds to the normalized partitioning of water uptake when the soil water pressure head is uniform. When used at the soil element scale instead of the root segment scale, they are called standard sink fractions (SSF_j) where “j” is the soil element number. SUF values are included in the [0; 1] interval and their sum for one root system is 1 by definition (same properties for SSF). Hence $H_{s,eq}$ is a plant-sensitive weighted-average soil water matric head.

A second emergent relation describes the water uptake distribution as the summation of two processes, i.e. the standard water uptake and the compensatory water uptake (i.e., passive adjustment of the uptake distribution due to the uneven distribution of soil water pressure heads):

$$J_{r,i} = T_{act}.SUF_i + K_{comp}.\left(H_{sr,i} - H_{s,eq}\right).SUF_i \qquad (17)$$

whereK_{comp} (L^3 L^{-1} T^{-1}) is the compensatory root water uptake conductance. It is worth noting that (i) the spatial integration of compensatory root water uptake terms is zero, (ii) hydraulic redistribution is simulated when T_{act} is smaller than the range of negative $K_{comp}.\left(\psi_{sr,i} - \psi_{s,eq}\right)$ values, and (iii) compensatory root water uptake occurs whenever the soil water pressure head is non-uniform.

For the compensatory root water uptake term (second term on the right-hand side) to be accurate, root radial conductances need to be much lower than root axial conductances, which is commonly observed in plants(French et al. 2009).

Accounting for the effects of gaps or change in root properties on rhizosphere hydraulic properties.

The impact of gaps between soil and root interface or specific processes occurring in the rhizosphere at a scale lower than the spatial resolution of the soil, can be accounted for by defining apparent root radial conductance parameters. In that case equation 8 becomes:

$$q_r = \widehat{L_r}\,(h_s)[h_s - h_x] \qquad (18)$$

where$\widehat{L_r}$ represents not only the radial root conductivity but also the rhizosphere properties and h_s is the bulk soil pressure head in the rhizosphere. The $\widehat{L_r}$ becomes then a function of h_s. Examples of such a parameterization is given in Couvreur et al. (2014).

1.1.4. Coupling root-soil water flow equations

The Sink term for water is defined byLeitner et al. (2014) and Couvreur et al. (2013)as:

$$S = \frac{1}{|V|}\int_V \delta_2\left(\chi(\boldsymbol{x})\right) q_r \, d\boldsymbol{x} \qquad (19)$$

whereχ(**x**) is a function that is 0 when **x** is on the centerline of a root in V and δ_2 is a 2-dimensional Dirac delta function [L^2], and q_r is given by Eq(9).

1.1.5. Analytical solutions for rhizosphere water flow

To avoid to fine spatial resolution, which could generate heavy computational load, analytical solution of the axisymmetric Richards equation can be used. Tom Schroeder developed a series of analytical solutions accounting for the presence of multiple roots within a given voxel (see Schroeder et al., 2008;2009).

Commenté [MJ2]: Add tom PhD

Axisymmetric Richard's equation writes:

$$C(h)\frac{\partial h}{\partial t}=\frac{\partial\theta(h)}{\partial t}=\frac{-1}{r}\frac{\partial}{\partial r}\left(rk(h)\frac{\partial h}{\partial r}\right)=-\frac{k(h)}{r}\frac{\partial h}{\partial r}-\frac{\partial}{\partial r}\left(k(h)\frac{\partial h}{\partial r}\right) \tag{20}$$

One of the first solution for Eq. (20) has been proposed by Gardner, under steady-state conditions, assuming a constant k(h).Given the pressure head boundary conditions h_{bulk} and h_{int} [L] at the bulk (r_{bulk}) and soil root interface (r_{int}) respectively, it yields:

$$j_r=\frac{k(h)2\pi}{\ln\rho}\left(h_{bulk}-h_{\text{int}}\right) \tag{21}$$

Where, $\rho=r_{bulk}/r_{\text{int}}$ is a dimensional geometrical factor. The total flow J_r [L^3 T^{-1}] at the soil root interface is equal to:

$$J_r=\frac{k(h)2\pi l}{\ln\rho}\left(h_{bulk}-h_{\text{int}}\right) \tag{22}$$

where l is the root segment length [L]. The flux at the soil root interface q_r [LT^{-1}]is

$$q_r=\frac{k(h)}{r_{\text{int}}\ln\rho}\left(h_{bulk}-h_{\text{int}}\right) \tag{23}$$

Under flux and water potential type boundary conditions at the inner (r_{root}) and outer edges (r_{out}) representing the bulk soil, and under steady-rate behavior, i.e., (∂θ/∂t) constant, the general solution for the radial flux [LT^{-1}] was proposed by Schroeder et al. (2009) without the assumption of uniform k(h):

$$q_r=\frac{\bar{k}B}{r_{\text{int}}}\left(h_{bulk}-h_{\text{int}}\right)+B\chi_1+\chi_2 \tag{24}$$

where

$$\bar{k}=\frac{\int_{h_{\text{int}erface}}^{h_{bulk}}K(h)dh}{h_{bulk}-h_{\text{int}}}=\frac{\varphi_{bulk}-\varphi_{\text{int}}}{h_{bulk}-h_{\text{int}}} \tag{25}$$

$$B=\frac{2\left(1-\rho^2\right)}{-2\rho^2\left(\ln\rho-{}^1\!/_2\right)-1}$$

$$\chi_1=q_{bulk}\rho\ln\frac{1}{\rho}$$

$$\chi_2=q_{bulk}\rho$$

where φ is the matric flux potential [LT^{-1}]. Note that if If $h_{bulk}=h_{\text{int}}=h$ then $\bar{k}=K(h)$, like in the Gardner equation.

Note that other assumptions exist and affect B, and solutions under other conditions have been propsoed in the litterature. Gardner steady-state or steady-state give:

$$B_{steady-state} = \frac{1}{\ln(\rho)}$$

$$B^{*}_{steady-rate} = \frac{1}{\left[1-1/\rho^{2}\right]^{-1}\ln(\rho)-1/2}$$

Peter de Willigen gives, for steady-rate (reference):

$$B^{*}_{deW} = 1\frac{\left(\rho^{2}-1\right)}{G_{0}(\rho)}$$

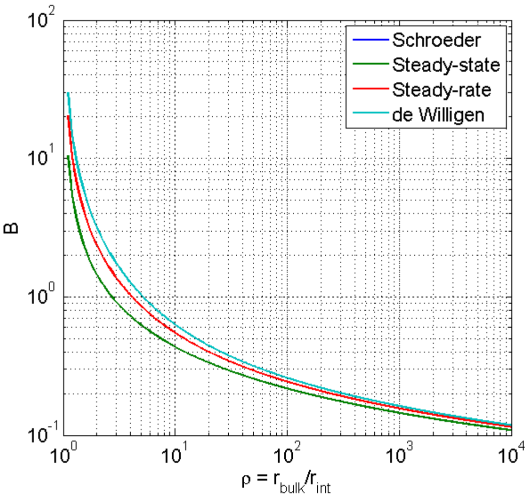


Figure 1.B function by several authors. Schroeder and Gardner steady-rate solutions overlap each other.

1.2. Solute transport

1.2.1. Solute transport in soil

The solute transport model solves the three-dimensional form of the advection-dispersion equation for solute concentration in the soil$c$ [ML^{-3}]:

$$\frac{\partial \theta c}{\partial t}+\frac{\partial \rho s}{\partial t}=\frac{\partial}{\partial x_i}\left(\theta \boldsymbol{D}_{ij}\frac{\partial c}{\partial x_j}\right)-\frac{\partial q_i c}{\partial x_i}+\mu_w \theta c+\mu_s \rho s+\gamma_w \theta+\gamma_s \rho-S_{root} \quad (26)$$

where θ is the water content [L^3 L^{-3}], ρ the soil bulk density [M L^{-3}], s the adsorbed concentration [M M^{-1}] with s=K$_d$ c with K$_d$ [L^3 M^{-1}], t the time [T], $q_i$is the i-th component of the volumetric water flux tensor [L^3 T^{-1}], μ$_w$ andμ$_s$ are the first-order rate constant for liquid and solid phases [T^{-1}], γ$_w$ and γ$_s$ are the zero-order rate constants for liquid [M L^{-3} T^{-1}] and solid [T^{-1}] phases respectively, and S$_{root'}$[M L^{-3} T^{-1}] sink term for root solute uptake. The variable $\boldsymbol{D}_{ij}$[L^{2}T^{-1}] is the ij-th component of the dispersion tensor, given for partially saturated soil bySimunek et al. (1995):

$$\theta \boldsymbol{D}_{ij}=a_T|q|\delta_{ij}+(a_L-a_T)\frac{q_i q_j}{|q|}+\theta D_m \tau \delta_{ij} \quad (27)$$

whereα_L and α_T are the longitudinal and transversal (perpendicular to a streamline) dispersivities [L], respectivel, δ_{ij} is the Kronecker delta function (i.e., δ$_{ij}$ =1 when i=j and δ$_{ij}$ =0 when i≠). D_m [L^2 T^{-1}] is the molecular diffusion coefficient of solute in free solution and τ is the tortuosity factor which accounts for the increased path length in soil. The relationship of Millington and Quirk (1961) is used to describe the tortuosity factor as a function of the volumetric water content:

$$\tau=\frac{\theta^{7/3}}{\theta_s^2} \quad (28)$$

1.2.2. Root solute uptake

Different options exist for defining the solute uptake by root at or below soil element scale.

Root solute uptake – Option 1: The purely macroscopic model

Root solute uptake can be accounted for in the soil solute transport equation (Eq. (26)) through a sink term S$_{root}$ defined at the soil element scale. Two different mechanisms of solute uptake are defined in R-SWMS: passive and active uptake. The uptake mechanism depends on the solute type (Gregory, 2006). The definition ofSomma, et al. (1998) and Hopmans and Bristow (2001)is adopted and the root solute uptake term S_{root}writes:

$$S_{root}=\delta Sc+(1-\delta)A \quad (29)$$

where δ∈*[0,1]* [-] is a partition coefficient, regulating both uptake mechanisms and determines how much solute is taken up by which uptake mechanisms, S [T^{-1}] is the sink term for root water uptake defined in Eq. (19), *c* is the solute concentrationin the soil and *A* [M L^{-3} T^{-1}] is the active solute uptake.

The first term of the right-hand side refers to passive uptake. This term contains the water sink term *S* from the Richards equation (Eq.(1)) and is independent of concentration gradients around a root. In the second term, the active solute uptake *A* is driven by ion uptake through electro-chemical gradients, ion pumping or ion channels.

Experimental results(Kochian and Lucas 1982; Siddiqi et al. 1990) have shown that the kinetics of active uptake is best described by the sum of a Michaelis-Menten component(Barber 1995)and a linear component (Kochian and Lucas 1982):

$$A = (\frac{V_{max}}{K_m + c} + f)cR_d \qquad (30)$$

whereV_{max} [M L^{-2} T^{-1}] is the maximum uptake rate per area of soil-root interface, K_m [M L^{-3}] Michaelis-Menten constant denoting the concentration where the concentration uptake is half of the maximal possible uptake, *f* first-order rate coefficient [L T^{-1}], and *Rd* [L^2 L^{-3}] is the root surface area density. An example is given at Figure 2. The soil concentration c can be obtained through the solution of the CDE in the soil (Eq.(26)).

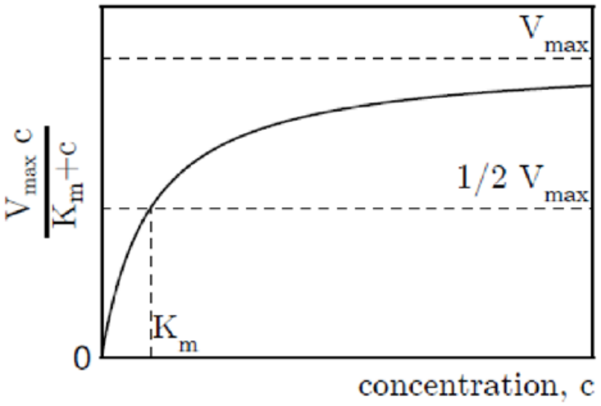


Figure 2 Michaelis-Menten description for active solute uptake by roots and a linear component f = 0, Eq. (30), adapted from Hopmans and Bristow (2001), (Schröder, 2013)

Root solute uptake – Option 2: The hybrid model including concentration gradients in the rhizosphere of each individual root segment

To avoid heavy computational requirements when solving soil-root interaction problems with high spatial resolution on the whole domain, we can solve the flow and transport equations with higher resolution only around the roots, i.e. in the rhizosphere. We can then consider mass and flow transport to be axi-symmetric and perpendicular to the root axis. This assumption is assumed to be valid for distances not too far from the root surface.

In that case, the flow and transport to a root segment can be described by radially one-dimensional equations.

The CDE in cylindrical coordinate can be written as:

$$\frac{\partial(b+\theta)c}{\partial t}=\frac{1\partial}{r\partial r}\left(\theta D_m\tau\frac{r\partial c}{\partial r}\right)-\frac{\partial(qc)}{\partial r}-\frac{qc}{r} \tag{31}$$

where$r[L]$is the distance from the soil domain geometry center coaxial with the root axis, q $[L^3L^{-2}T^{-1}]$ is radial flux of water at r which is positive in the direction away from the root-soil surface.

Equation (31) is subject to a boundary condition that describes the solute uptake at the root surface r =r$_0$, and is given by Michaelis–Menten solute uptake law (Barber, 1995), i.e.

$$\begin{cases} c=c_i(r_0\le r\le r_n,\ \ t=0) \\ -\theta D_m\tau\frac{\partial c}{\partial r}+qc=-\frac{V_{max}(c-c_{min})}{K_m+(c-c_{min})}(r=r_0,\ \ t>0) \\ -\theta D_m\tau\frac{\partial c}{\partial r}+qc=J_{out} \qquad (r=r_n,\ \ t>0) \end{cases} \tag{32}$$

where J$_{out}$[M L^{-2} T^{-1}]is the solute/nutrient mass flux at the outer boundary (r=r$_n$), c_i is the initial concentration, c$_{min}$[M L^{-3}] is the minimal concentration at which uptake takes place.

Root solute uptake – Option 3: Pesticide uptake

Pesticide uptake and translocation in plants can be modeled using R-SWMS. The simple approach developed by Trapp [2000] is used, consisting in an equilibrium model for pesticide uptake, containing three model compartments: soil, root and xylem. In the following, we will refer to these compartments with the subscripts S, R and X respectively. The root compartment describes the root cortex; the xylem compartment describes the root stele, with its inner vascular system. Exchange between the compartments is only diffusive.

Trapp (2000)differentiates between neutral and dissociating organic compounds. The incorporation of dissociated compounds is an enhancement, as many pesticides and other xenobiotic chemicals are dissociating under relevant environmental conditions. Dissociated compounds undergo specific processes when taken up by the plant, which are irrelevant for neutral compounds.

Uptake

Solute uptake for pesticide can be described using micro- or macroscopic approaches.

In the microscopic approach, the radial soil transport equation defined in Eq.(31)is used withthe boundary condition at the soil-root interface defined as:

$$q_{R,s}=P(C_{S,f}-C_{R,f})+\varepsilon q_R C_{S,f} \tag{33}$$

whereP [L/T] denotes the root membrane permeability, $q_{R,s}$ is the solute flux [M L^2 T^{-1}], ε is the portion of the transpiration stream fed via the apoplastic pathway [-], q_R is the water flux [L^3 L^{-2} T^{-1}] and $C_{S,f}$and $C_{R,f}$ the solved solute concentration in the soil and in the cortex, respectively [ML^{-3}].

Commenté [MJ3]: Is it q in equation 26?

If the macroscopic approach is used, then a solute sink term is defined at the soil element scale, with the following definition:

$$S_{root} = \frac{1}{|V|}\int_V \delta_3\big(\chi(\boldsymbol{x})\big)J_{R,s}\,d\boldsymbol{x} \qquad (34)$$

Where $\chi(\mathbf{x})$ is a function that is 0 when **x** is on the center of a root segment located in V and else 1; δ_3 is a 3-dimensional Dirac delta function (L^{-3}), and $J_{R,s}$is the radial solute flow rate through a root segmentdefined as:

$$J_{R,s} = q_{R,s}A_s \qquad (35)$$

where A_s, is the area of the outer surface of the root segment and $q_{R,s}$ is defined by Eq. (33).

Root compartments

For root and xylem compartment, the fractions of neutral, dissociated and sorbed compound are calculated, depending on compartment pH, lipid and water fraction; and compound pKa, activity coefficient and Kow. The fraction of neutral free solutes therefore is

$$f_{nf} = \frac{1}{\left({}^{W}/_{\gamma_n} + {}^{K_n L}/_{\gamma_n} + {}^{K_D W}/_{\gamma_d}\right)} \qquad (36)$$

where W and L describe the water and lipid weight fraction [M/M] of the plant compartment, respectively; $\gamma_n$and $\gamma_d$the activity coefficient of neutral and dissociated compounds [-], K_n is the root-water partitioning coefficient, K_D [-]denotes the partitioning between ions and neutral phase according to the Henderson-Hasselbach equation. In soil solution, the fraction of neutral and ionic compounds can be determined on a similar basis, depending on organic matter content, pore volume and water content of the soil (Trapp 2007).

Exchange rates are calculated, incorporating neutral and dissociated fractions, cell permeabilities, Nernst effects for dissociated compounds and exchange area. Using a mass balance approach for each compartment, change in compound concentration can be expressed as follows:

$$\frac{dC_R}{dt} = \frac{\{E_{SR}C_S - E_{RS}C_R + E_{XR}C_X - E_{RX}C_R\}}{V_R} - kC_R \qquad (37)$$

$$\frac{dC_X}{dt} = \frac{\{E_{RX}C_R - E_{XR}J_X - Q_X J_X\}}{V_X} - kC_X \qquad (38)$$

where $\mathbf{E_{ij}}$ [L^3/T] denotes the exchange rates, $\mathbf{C_i}$ [M/L^3] the concentration, $\mathbf{V_i}$ [L^3] the compartment volume, **k** [T^{-1}] the first order sink term to account for growth and metabolism and $\mathbf{J_x}$ [L^3/T] the xylem flow rate defined by Eq. (10), the indices S, R and X denote the soil, root and xylemcompartments, respectively. Hereby, C_S regards the concentration in soil solution.

The exchange rates are composed of an exchange term for neutral compounds and an exchange term for dissociated compounds. For example, the exchange rate between soil and root is described by

$$E_{RX} = A_X\{f_{nfR}P_{nC} + f_{dfR}P_{dC}exp^N N/(e^N - 1)\} \tag{39}$$

$$E_{XR} = A_X\{f_{nfX}P_{nC} + f_{dfX}P_{dC}N/(e^N - 1)\} \tag{40}$$

where $\boldsymbol{A_X}$ denotes the outer area of the stele [L^2], $\mathbf{f_{nfR}}$ the fraction of neutral free compounds in the root compartment and $\mathbf{P_{nC}}$ [L/T] the cell permeability for neutral compounds. $\mathbf{f_{dfs}}$ denotes the fraction of dissociated free compounds in the soil solution, $\mathbf{P_{dC}}$ [L/T] the cell permeability for dissociated compounds and N is the Nernst factor [-]. The permeability is related to the diffusion coefficient, as it obtained by dividing diffusion coefficient by thickness of the membrane. When calculating plant uptake of a neutral compound, the second term in the exchange rate equation drops out.

Diffusion rates are limited by the membrane permeability. The cell wall has a diffusion coefficient of circa 10^{-10} m/s, and a thickness of 0.4 µm, resulting in a permeability of circa P_W=2.5^{-4} m/s and is constant for water and other solutes. The plasmalemma permeability however, is a function of the chemicals logK_{ow} and differs for neutral and dissociated compounds. Membrane permeability for neutral compounds is

$$P_M = 10^{1.2logK_{ow}-7.50} \tag{41}$$

For dissociated compounds logK_{ow} is 3.5 units lower. The total cell permeability [L/T] results from cell wall permeability and membrane permeability connected in series

$$P_C = \frac{1}{{}^{1}/_{P_W} + {}^{1}/_{P_M}} \tag{42}$$

Currently, we consider the root system as a one compartment, porous media and treat it similar as the soil compartment. This means we do no differentiate between root cortex and root stele. Rewriting the approach by Trapp to the chosen porous media notation we gain

$$\frac{dC_{R,T}}{dt} = \frac{A_R}{V_R}\left[P\left(C_{S,f} - C_{R,f}\right) + \varepsilon J_w C_{S,f}\right] - \frac{Q_X}{V_R}C_{R,f} - k_R\theta_R C_{R,f}. \tag{43}$$

Where $\mathbf{C_{R,T}}$ is the total root concentration [M/L^3], $\mathbf{C_{S,f}}$ and $\mathbf{C_{R,f}}$ are the solved solute concentration in soil and root compartment, respectively. $\mathbf{A_R}$ and $\mathbf{V_R}$ describe the root circumferential area [L^2] and root volume [L^3], **P** the root membrane permeability [L/T], **ε** the advective solute uptake fraction [-], $\mathbf{J_w}$ the root water uptake flux [L/T] and $\mathbf{Q_x}$ the transpiration [L^3/T]. $\mathbf{k_R}$ is the solute decomposition rate [T^{-1}] and $\boldsymbol{\theta_R}$ the root water content [-]. The total solute concentration within the root is composed of solute solved in the root water phase, and solute immobilized due to sorption to root lipid phase. This is described by

$$C_{R,T} = \theta_R C_{R,f} + \rho_R C_{R,s}\ . \tag{46}$$

For linear sorption the total solute concentration within the root can be described by

$$C_{R,T} = (\theta_R + \rho_R K_{D,R})C_{R,f}, \qquad (44)$$

where **K_{D,R}** is the root sorption parameter [L³/M] and for freundlich sorption the total root concentration can be formulated as

$$C_{R,T} = \theta_R C_{R,f} + \rho_R K_{D,R} {C_{R,f}}^n, \qquad (45)$$

Where **n** is the freundlich sorption parameter.

1.2.3. Transport in the root system

For certain processes like chemical signaling or pesticide uptake, the transport of solute within the plant needs to be modeled explicitly. In such case, the solute transport in the xylem vessel network is modeled via a convection equation:

$$\frac{\partial C_x}{\partial t} = \nabla . (J_x C_x) + S_{c,x} \qquad (47)$$

where **C** is the xylem concentration [M L³], **Jx** is the xylem water flux [L³ T⁻¹] and ***Sc,x*** is a source/sink term of solute within the xylem [M L³T⁻¹]. A particle tracking algorithm can be used to simulate transport by convection within the root. If the flow field in the root vascular system is known from the Doussan equation (Eq. (10)), the averaged velocity within a segment is calculated as

$$\tilde{v} = \frac{J_x}{A_x} \qquad (48)$$

where $\boldsymbol{A_x}$ is the cross section area of the xylem vessel (see Eq. (39)).

1.3. Two-way coupled flow and transport in the soil-root system: effect of salinity on root water uptake

Salinity effects on root water uptake can be accounted for in R-SWMS. The transpiration flux drives water towards the root and, as a consequence, thesolute that is dissolved in the water. This leads to salt accumulation around roots, which can lead to a reduction in transpiration as very low osmotic potentials are added to the soil total water potential around the root. The change in transpiration has an effect back to the water flow in the soil and, hence, to the solute transport.

The root surface is a complex of membranes, and that is why a gradient in osmotic potential between the soil and the root can lead to water flux. The degree of permeability of the root to the transport of solute particles across its surface will affect the magnitude of the water flux across the root.

No toxicity effects or plant osmotic adjustments are modeled in R-SWMS. Therefore, osmotic stress accounts only for the reduction of transpiration due to the presence of salt at the soil-root interface which lowers the total water potential.

1.4. Root growth

1.4.1. Root architecture information

In the R-SWMS model, a root system consists of logically connected nodes containinginformation about the coordinates, root diameter and length of the previous segment (see Figure 3). Two options exist to model root architecture:

- **Option 1: Static root system architecture is used as input file for the R-SWMS simulation.**This data can come from experimental measurements or from a root system generator(e.g. Root Typ, Pagès et al. 2004; or RootBox, Leitner et al. 2010).

- **Option 2: Root growth is modelled and coupled with a soil model.** This option is either based on the root growth model of Somma, Clausnitzer, and Hopmans (1997) or on the model RootBox by Leitner et al. (2010) with the root growth direction approach by Somma, Clausnitzer, and Hopmans (1997). At each time step, root growth and soil water and solute transport are simulated alternately.

1.1.1. Growth rules for one time step and one root of a given order

The model by Somma et al. (1997) calculates the root growth direction as the vector sum of various direction affecting components. This approach works well for the simulation fo root growth in bulk soil; it fails, however, to reproduce experimentally observed root growth patterns in soil with macropores. We thus developed a new, more mechanistic modeling approach for the simulation of root growth in structured soil. Both modeling approaches are implemented in R-SWMS and described below.

1.1.1.1. Computation of root growth direction, approach by Somma et al. (1997):

In the model by Somma et al. (1997) the root growth direction $\vec{d}$is computedfor each time step using three direction affecting components: (1) a vector a, which comprises the old root growth direction (defined by the angles α and β) and a random deflection (defined by the angles γ (rad) and δ(rad); the computation of these random deflection angles is explained in section 1.1.1.2.1), (2) a vector g pointing towards the preferred growth direction (e.g. geotropism) and (3) a vector ΔR pointing towards the gradient of least soil resistance.

The direction of root growth is calculated as the vector sum of the three direction affecting components as

$$\vec{d} = \left[\mathrm{a}_{(\alpha,\beta\gamma,\delta)} + s_g * g + s_r * \Delta R\right], \tag{49}$$

Where $\vec{\boldsymbol{d}}$ is the new growth direction, $\mathbf{s_g}$ is is the sensitivity to the preferred growth direction and $\mathbf{s_r}$ is the sensitivity to soil penetration resistance.

1.1.1.2. Computation of root growth direction, alternative approach for root growth in soil with macropores

In our alternative modelling approach, we distinguish between, firstly, the driving force for root growth, which is determined by the orientation of the previous root segment and the influence of gravitropism and, secondly, soil mechanical resistance to root growth. The latter is expressed by its inverse, soil mechanical conductance, and treated similarly to hydraulic conductivity in Darcy's law. At the presence of macropores, soil mechanical conductance is anisotropic, which leads to a difference between the direction of the driving force and the direction of the root tip movement.

In analogy with soil water flow, the direction of the root tip movement corresponds to the water flux vector, while the driving force represents the gradient of the water potential. Soil mechanical forces can be seen as the viscous friction forces that counteract water flow and thus the driving force. In the Darcy equation, the effect of these viscous forces on energy dissipation is represented by the hydraulic conductance tensor, which may show anisotropy. Analogous to the Darcy flow equation, the movement of the root tip is expressed by the following equation:

$$\vec{d} = k \cdot F \tag{50}$$

where $\vec{\boldsymbol{d}}$ is the new root growth direction, **k** is the soil mechanical conductance tensor that represents the ease with which the root can penetrate the soil and **F** is the driving force that influences the root growth direction.The soil mechanical conductance tensor **k** is a symmetric, second rank tensor with nine entry values defining conductances in the three principal directions:

$$\mathbf{k} = \begin{pmatrix} \mathrm{k_{xx}} & \mathrm{k_{xy}} & \mathrm{k_{xz}} \\ \mathrm{k_{yx}} & \mathrm{k_{yy}} & \mathrm{k_{yz}} \\ \mathrm{k_{zx}} & \mathrm{k_{zy}} & \mathrm{k_{zz}} \end{pmatrix}. \tag{51}$$

The driving force **F** could comprise several factors. We chose here the direction vector of the previous root segment and gravitropism. The direction of the previous root segment is expressed by the azimuth angle **α** and the polar angle **β**. To account for small scale variations in the soil matrix and to represent a random behavior of the root tip, random deflection angles **γ** and **δ** are added to α and β. The computation of those random deflection angles is explained in section 1.1.1.2.1.

$$\mathrm{F} = \begin{pmatrix} \mathrm{dx}_{(\alpha,\beta,\gamma,\delta)} \\ \mathrm{dy}_{(\alpha,\beta\gamma,\delta)} \\ \mathrm{dz}_{(\alpha,\beta\gamma,\delta)} \end{pmatrix} + \mathrm{sg} \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \tag{52}$$

While the first term on the right hand side of Eq. (52) represents the previous growth direction vector of F, the second term expresses the gravitropism component with **sg** as gravitropism sensitivity factor.

For the simulation of root growth in soil with macropores, it is necessary to set a value for the conductance in the macropore. The conductance in the macropore can be regarded as a sensitivity factor, which influences the probability of a root to continue growing within the macropore or to re-enter the bulk soil. The higher the conductance in the macropore, the stronger the root will be forced to stay within the macropore. Analyses showed that macropore conductance values of 1e4 ≤ **kmacro** ≤ 2e5 kPa-1 are most plausible.

If no soil anisotropy is present, the approach by Somma et al. (1997) and the alternative approach for root growth in soil with macropores lead to the same result.

1.1.1.2.1. Determination of the random deflection angles

The random deflection angles **γ** and **δ** take into account microscale differences in soil structure. While the deflection of the azimuth angle **γ** is a uniformly distributed random angle between [0 2π], the deflection of the polar **δ** is a normally distributed random angle with mean zero and standard deviation **σdl**, which is calculated following the approach by Leitner et al. (2010). The standard deviation **σdl** is derived from the user defined unit standard deviation **σ** of a root segment of 1 cm length and the maximum root segment length **dl**, which is reached when soil penetration resistance equals zero and temperature and soil solute concentration are at their optimum. In probability theory, standard deviation decreases by the square root of the number of trials. If 1/dl (segments per cm) is regarded as the number of trials, the standard deviation σdl (° cm1/2) can be given as

$$\sigma_{\mathrm{dl}} = \sqrt{\mathrm{dl}} * \sigma \qquad (53)$$

In this way, the deflection from the original root tip location does not depend on the spatial resolution of the root growth model. By using the maximum root segment length as normalization factor for the standard deviation of the random deflection angle, we create a dependency between σdl and soil penetration resistance. In this way, experimental observations of higher root tortuosity in more compact soil (Tracy 2013) are taken into account.

1.1.1.3. Root elongation rate

The length of a newly developed root segment is calculated as the product of the root elongation rate and a user defined root growth time step. The root elongation rate **Er** (cm d-1) is assumed to be a function of soil strength, and – if defined - temperature and solute concentration.

Impact of soil strength

Bengough et al. (2011) observed that the soil penetration resistance sufficient to stop root elongation completely (**Rmax, kPa**) is a function of matric potential (**ψ, kPa**) and can be calculated as

$$R_{max} = 4000 + 2.33 * |\psi|, \tag{54}$$

The root elongation rate is assumed to decrease linearly between zero and maximum soil penetration resistance. The actual root elongation rate can thus be calculated based on the fraction of the maximum root elongation rate by

$$E_r = E_{max} * \left(1 - \frac{R_p}{R_{max}}\right), \tag{55}$$

where Er (cm d-1) is the actual root elongation rate, Emax (cm d-1) is the maximum root elongation rate and Rp is the penetration resistance experienced by the root tip.

In the approach by Somma et al. (1997), Rp is calculated as the average of the penetration resistances of the eight corner nodes of the grid cell in which the root tip is located. In the alternative approach for root growth in soil with macropores, Rp is defined as the inverse of an effective conductance in the direction of root growth, keff:

$$R_p = \frac{1}{k_{eff}}; \tag{56}$$

keff is a function of both the average conductance of the grid cell (see Eq. (51)) and the root inherent driving force and is calculated as:

$$k_{eff} = \left|\mathbf{k} \cdot \hat{\boldsymbol{F}}\right|, \tag{57}$$

where $\hat{\mathrm{F}}$ is the unit length vector of the driving force. In this way, Er is reduced stronger if the root grows perpendicular to a hard soil layer than if it grows along a hard soil layer. This approach corresponds to observations by Kolb et al. (2012) who found that radial constrictions applied to roots did not significantly reduce root elongation rates, while axial constrictions did have a significant impact.

Impact of soiltemperature

The impact of soil temperature on the root growth rate is calculated using the temperature impedance factor imp_t as

$$\mathrm{E_r} = E_{max} * imp_t, \tag{58}$$

where E_r is the actual root elongation rate, E_{max} the maximum root elongation rate and imp_t the temperature impedance factor. If the temperature is at its optimum tem_{opt}, imp_t equals one and the root elongation rate is not impeded. If the soil temperature is below the minimum temperature limit tem_{min}or above the maximum temperature limit tem_{max}, imp_t equals zero and root elongation is stopped completely. If the soil temperature is in between tem_{min} and tem_{max}, imp_t is calculated as

$$imp_t = sin\pi \left(\frac{tem - tem_{min}}{tem_{max} - tem_{min}}\right)^{ln 0.5 / ln \frac{tem_{opt} - tem_{min}}{tem_{max} - tem_{min}}}, \text{ if } tem_{opt} < \frac{tem_{min} + tem_{max}}{2}, \tag{59}$$

and as

$$imp_t = sin\pi \left(\frac{tem - tem_{max}}{tem_{min} - tem_{max}}\right)^{ln0.5/ln\frac{tem_{opt} - tem_{max}}{tem_{min} - tem_{max}}}, \text{ if } tem_{opt} > \frac{tem_{min} + tem_{max}}{2}. \quad (60)$$

Impact of soil solute concentrations

The impact of soil solute concentration on the root growth rate is calculated with the concentration impedance factor imp_c as

$$\mathrm{E_r} = E_{max} * imp_c, \quad (61)$$

where E_r is the actual root elongation rate, E_{max} the maximum root elongation rate and imp_c the concentration impedance factor. If the soil solute concentration is at its optimum c_{opt}, imp_c equals one and the root elongation rate is not impeded. If the soil solute concentration is below the minimum solute concentration c_{min} or above the maximum solute concentration c_{max}, imp_c equals zero and root elongation is stopped completely. If the soil solute concentration is in between c_{min} and c_{max}, imp_c is calculated as

$$imp_c = \frac{c - c_{min}}{c_{opt} - c_{min}}, \text{ if} c_{min} < c < c_{opt}, \quad (62)$$

and as

$$imp_c = \frac{c_{max} - c}{c_{opt} - c_{min}}, \text{ if} c_{opt} < c < c_{max}. \quad (63)$$

1.1.1.4. Branching

The distance between two branches is a function of root segment age. If the elongation rate is impeded by soil strength, temperature or soil solute concentration, the distance between two branches is shorter. The initial growth angle of each branch is a user defined input parameter.

1.1.1.5. Root radius and radial root growth

The root radius depends on the soil penetration resistance and is calculated via the mass of a root segment with unit length.

It is also possible to include radial root growth.

2. Numerics

2.1. Water flow

2.1.1. Numerical solution of soil water flow

The numerical solution for the water flow equation in R-SWMS is almost identical to the SWMS_3D code proposed by Simunek et al. (1995). In general, the Richard's equation (Eq. 1) is solved using the Galerkin finite element method. The discretization of the flow domain is done by dividing it to N cubical elements and subdivides each cube to 5 tetrahedral sub-elements. The corners of the sub-elements are the domain nodes. This procedure leads to a system of ordinary differential equation which in matrix form can be expressed as (for full derivations see Simunek et al. 1995):

$$[F_{mul}]\frac{d\{\theta\}}{dt} + [A]\{h\} = \{Q\} - \{B\} - \{D\} \tag{64}$$

whereF_{mul} , $A, Q, B\ and\ D$ are the matrix coefficient for a detailed description of each of the variables, the reader is referred to Simunek et al. (1995). In R-SWMS the first term in the LHS of eq. 28 is given in term of potential $[FC]d\{h\}/dt$ where FC_n is given by

$$FC_n = \sum_{se} \frac{V_e}{20}(4\bar{C} + C_n) \tag{65}$$

the subscript e and n and se refers to element, the node and the subelment, respectively, V_e is the volume of the element, C_n is the water capacity in the nodes of the sub-element and $\bar{C}$ is the average water capacity in the sub-element. In addition, in R-SWMS the sink term (related to the coefficient D in eq. (64)) is calculated on an element basis and not on sub-element basis (as in SWMS_3D) such that

$$D_n = \sum_{se} \frac{V_e}{4} S_e \tag{66}$$

where S_e is the element (volumetric) sink obtained based on Doussan solution (Eq. (19)).The final set of equations is obtained replacing the time derivative with a backward finite difference scheme and integrating in time.

$$[FC]_{j+1}\frac{h_{j+1} - h_j}{\Delta t_j} + [A]_{j+1}\{h\}_{j+1} = \{Q\}_j - \{B\}_{j+1} - \{D\}_{j+1} \tag{67}$$

or in a more convenient form

$$\left[\frac{[FC]_{j+1}}{\Delta t_j} + [A]\right]\{h\}_{j+1} = [FC]_{j+1}\frac{h_j}{\Delta t_j} + \{Q\}_j - \{B\}_{j+1} - \{D\}_{j+1} \tag{68}$$

A Picard iteration scheme is used to update coefficient matrices.

$$\left[\frac{[FC]_{j+1}^{k}}{\Delta t_j} + [A]^k\right]\{h\}_{j+1}^{k+1} = [FC]_{j+1}^{k}\frac{h_j}{\Delta t_j} + \{Q\}_j^k - \{B\}_{j+1}^k - \{D\}_{j+1}^k \tag{69}$$

Where$k$ is the iteration. Instead of the h-form, a mixed formulation of the Richards equation, which insures mass conservation, can be used (Celia et al., 1990):

$$\left[\frac{[FC]_{j+1}^{k}}{\Delta t_j}+[A]^k\right]\{h\}_{j+1}^{k+1}=[FC]_{j+1}^{k}\frac{h_{j+1}^{k}}{\Delta t_j}+[F]\frac{\{\theta\}_{j+1}^{k}-\{\theta\}_j}{\Delta t_j}+\{Q\}_j^k-\{B\}_{j+1}^{k}-\{D\}_j^k \qquad (70)$$

2.1.2. Root water flow equations

2.1.2.1. Root system Setup

In R-SWMS, the root system is made up of nodes and connecting segments (Figure 3.). The numbering of nodes starts from the collar. The segments are numbered according to the number of the more distal node to the collar that bounds the segment. In the root system file, one record contains the number of the node and the associated segment to that node, the number of the parent node (i.e. the node that connects the segment to the collar), the Cartesian coordinates of the node, the length of the segment and the surface area of the segment.

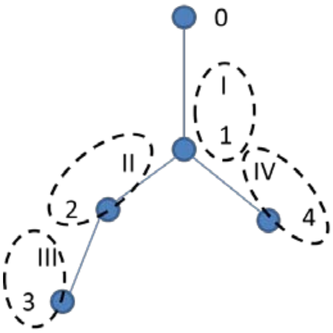


Figure 3.Sketch of the Doussan root system: definition of the nodes (Arabic numbers), segments (Roman numbers), and root nodes and segments that are grouped in records in the root system file (ellipses).

2.1.2.2. Numerical solution

The flux $J_{x,i}$leaving or entering a segment $i$is defined as

$$J_{x,i}=-K_{x,i}\frac{h_{x,i}+z_{x,i}-h_{x,k}-z_{x,k}}{l_i} \qquad (71)$$

where node k is the parent node of segment I and node i

therear end.The axial flow is negative when the flow is from the segment towards the parent node and negative when it is from the parent into the segment.

The water balance is written over the segment i:

$$\frac{\sum_l J_{x_l} - J_{x_i}}{l_i} = \frac{S_{r,i}K_{r,i}\left(h_{s,i} - h_{x,i}\right)}{l_i} \qquad (72)$$

where$\sum_l J_{x_l}$is the sum of the water fluxes from the segments that have node i as their parent node and $S_{r,i}$ is the surface area of segment *i*.

For given $h_{s,i}$, $h_{x,i}$ and $J_{x,i}$ are obtained by solving the Doussan Equations:

$$\boldsymbol{C} \cdot \boldsymbol{H_x} = \boldsymbol{Q} \qquad (73)$$

Here **C** (dimensions n_pxn_p with n_p the number of root nodes) is called the "conectivity matrix".**Q** (dimensions n_p x1) contains the water potentials at soil root interface and $\mathbf{H_x}$(dimensions n_p x1) is the root (xylem) total water potential ($H_{x,l}$ = $h_{x,l}$ + z_i. To have a complete description of these matrices, see Doussan et al. (1998).

The boundary conditions are defined at the root collar and enter the **C** and **Q** matrices. Following the type of boundary condition (pressure head or flux BC), the final form of the matrix is a bit different. For the flux at the collar defined, $J_{w,collar}$, the matrix equations are:

$$\begin{bmatrix} \frac{K_{x,1}}{l_1} & -\frac{K_{x,1}}{l_1} & & & & \\ & \cdots & & & & \\ & & \cdots & & & \\ & \cdots & -\frac{K_{x,i}}{l_i} & \left[\frac{K_{x,i}}{l_i} + \sum \frac{K_{x,l}}{l_l} + S_{r,i}K_{r,i}\right] & \cdots & -\frac{K_{x,l}}{l_l} \\ & & & & & \end{bmatrix} \begin{bmatrix} H_{x,0} \\ \cdots \\ H_{x,i-1} \\ H_{x,i} \\ \cdots \\ H_{x,l} \end{bmatrix} \qquad (74)$$

$$= \begin{bmatrix} J_{w,collar} \\ \cdots \\ \cdots \\ S_{r,i}K_{r,i}H_{s,i} \\ \\ \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & & & & \\ & \dots & & & & \\ & & \dots & & & \\ & \dots & -\frac{K_{x,i}}{l_i} & \left[\frac{K_{x,i}}{l_i} + \sum \frac{K_{x,l}}{l_l} + S_{r,i}K_{r,i}\right] & \dots & -\frac{K_{x,l}}{l_l} \\ & & & & & \\ & & & & & \end{bmatrix} \begin{bmatrix} H_{x,0} \\ \dots \\ H_{x,i-1} \\ H_{x,i} \\ \dots \\ H_{x,l} \end{bmatrix} \tag{75}$$

$$= \begin{bmatrix} H_{collar} \\ \dots \\ \dots \\ S_{r,i}K_{r,i}H_{s,i} \\ \\ \\ \end{bmatrix}$$

2.1.3. Coupling the root system equations with the soil water flow model

The water pressure at the soil-root surface of segment i, $h_{s,i}$ is derived from the water potentials at the soil nodes of the soil grid using the inverse distance weighted average of the pressure heads of the surrounding soil grid nodes. If the segment is located in one soil grid voxel, $h_{s,i}$ is calculated as:

$$h_{s,i} = \frac{\left(\sum_{j=1}^{8} \frac{1}{dist_j} h_{s,g,j}\right)}{\sum_{k=j}^{8} \frac{1}{dist_j}} \tag{76}$$

where$h_{s,g,i}$is the pressure head of the jth soil grid node and dist$_j$ is the distance of this node to the center of the root segment. If the segment crosses two voxels, the segment is divided in two sub segments with lengths $l_{i,1}$ and $l_{i,2}$and for each sub segment $h_{s,i}$ is calculated.

$h_{s,i}$of the entire segment is calculated using a length weighted average of the sub-element $h_{s,i}$:

$$h_{s,i} = \frac{l_{i,1}h_{s,i} + l_{i,2}h_{s,2}}{l_{i,1} + l_{i,2}} \tag{77}$$

And the sink term is calculated by:

$$S_j = \frac{\sum_i^{n_j} J_{r,i}}{V_j} \tag{78}$$

Doussan is re-calculated after each update of the soil matric potential.

2.1.4. Periodic root systems and boundary conditions

The periodic simulation domain allows generating scenarios where one root system interwoven with the roots of neighbouring identical plants is modelled as a repeated pattern (typically for field conditions). It also removes edge effects and may include several different root systems inside the repeated pattern.

Periodic root system

During the model setup, root segments are virtually translated of entire domain lengths in X and Y until their location is included inside the simulation domain. The file Transroot.out records the amount of necessary translations (positive, null or negative) for each root segments in directions X and Y. Each translated root segment is then associated to the specific soil cuboid that contains it. This association is key as it determines (i) in which soil elements the periodic roots take up water, and (ii) which soil-root interface water pressures are to be used as root segments boundary condition when solving Doussan equation. Solving water flow inside the root system is identical for periodic and non-periodic root systems, i.e. the original (non-translated) root geometry is used to generate Doussan matrix and the same equation is solved.

Periodic boundary conditions

While regular soil cuboids are constituted of eight soil nodes in the direct vicinity of each other, the periodic domain combines regular soil cuboids to discontinuous cuboids. If we were to look at a periodic soil domain vertically from the top, its Northern and Eastern boundaries would appear not to have soil nodes on their sides. The reason is that each cuboid at the Eastern boundary has one face on the Western boundary where four of its nodes are located. These four nodes also belong to a regular cuboid on the Western side. The same idea applies on the Northern vs Southern boundaries, which allows for a direct continuity of water flow between vertical boundaries of the simulation domain.

2.1.6. Multiple root systems (to be completed)

2.1.7. Implementation of non-resolved rhizosphere processes/properties(to be completed)

2.1.8. Speed up of simulation using SSFs(to be completed)

2.1.9. Direct solving of Doussan(to be completed)

2.2. Solute transport

2.2.1. Numerical solution of the soil transport equation

Option 1: Finite element macroscopic(from Somma, Hopmans, and Clausnitzer 1998)

Commenté [MJ4]: To check

The convection disperison equation can be written as follows:

$$-\theta R\frac{\partial c}{\partial t}-q_i\frac{\partial c}{\partial x_i}+\frac{\partial}{\partial x_i}(\theta \mathrm{D}_{ij}\frac{\partial c}{\partial x_j})+Fc+G=0\,; \qquad (79)$$

where:

$$F=\mu_w\theta+\mu_s\rho k+S-S'; \qquad (80)$$
$$G=\gamma_w\theta+\gamma_s\rho;$$

and:

$$R=1+\frac{\rho k}{\theta}. \qquad (81)$$

Following the same reasoning presented above we obtain:

$$\int_W[-\theta R\frac{\partial c}{\partial t}-q_i\frac{\partial c}{\partial x_i}+\frac{\partial}{\partial x_i}(\theta \mathrm{D}_{ij}\frac{\partial c}{\partial x_j})+Fc+G]\varphi_n\mathrm{d}W=0\,. \qquad (82)$$

Substituting *c(x,y,z,t)* with *c'*:

$$c'(x,y,z,t)=\sum_1^N\varphi_n(x,y,z)c_n(t)\,; \qquad (83)$$

where, again, φ_n are the linear basis functions and c_n are the weighting coefficient representing the solution of (A.12), and applying Green's theorem to the resulting expression we obtain:

$$\sum_e\int_{W_e}[(-\theta R\frac{\partial c'}{\partial t}-q_i\frac{\partial c'}{\partial x_i}+Fc'+G)\varphi_n-\theta \mathrm{D}_{ij}\frac{\partial c'}{\partial x_j}\frac{\partial \varphi_n}{\partial x_i}]\mathrm{d}W+ \qquad (84)$$
$$+\sum_e\int_{\Omega_N^e}\theta \mathrm{D}_{ij}\frac{\partial c'}{\partial x_j}n_i\varphi_n\mathrm{d}\Omega=0;$$

Where *W*

Where W_e is the volume of element *e* and Ω_e the boundary area of the *e*.The system of equations is then rearranged in the following matrix form:

$$[Q]\frac{\mathrm{d}\{c\}}{\mathrm{d}t}+[S]\{c\}+\{f\}=-\{Q^D\}\,; \qquad (85)$$

where:

$$\begin{aligned} S_{nm} &= \sum_e [(-q_i)_l \int_{W_e} \varphi_l \varphi_n \frac{\partial \varphi_m}{\partial x_i} \mathrm{d}W - (\theta \mathrm{D}_{ij})_l \int_{W_e} \varphi_l \frac{\partial \varphi_n}{\partial x_i} \frac{\partial \varphi_m}{\partial x_j} \mathrm{d}W + \\ &\quad + F_l \int_{W_e} \varphi_l \varphi_n \varphi_m \mathrm{d}W\,] = \\ &= \sum_e \{ -\frac{b_m}{120}(4\bar{q}_x + q_{xn}) - \frac{c_m}{120}(4\bar{q}_y + q_{yn}) - \frac{d_m}{120}(4\bar{q}_z + q_{zn}) + \\ &\quad + \frac{V_e}{120}(4\overline{F} + F_n + F_m)(1 + \delta_{nm}) + \\ &\quad - \frac{1}{36V_e}[b_n b_m \theta \mathrm{D}_{xx} + c_n c_m \theta \mathrm{D}_{yy} + d_n d_m \theta \mathrm{D}_{zz} + (b_n c_m + c_n b_m)\theta \mathrm{D}_{xy} + \\ &\quad + (b_n d_m + d_n b_m)\theta \mathrm{D}_{xz} + (c_n d_m + d_n c_m)\theta \mathrm{D}_{yz}]\}; \end{aligned} \tag{86}$$

and:

$$Q_{nm} = \sum_e (-\theta R)_l \int_{W_e} \varphi_l \varphi_n \varphi_m \mathrm{d}W = -\sum_e \frac{V_e}{20}(4\overline{\theta R} + \theta R_n)\delta_{nm}\,; \tag{87}$$

$$f_n = \sum_e G_l \int_{W_e} \varphi_l \varphi_n \mathrm{d}W = \sum_e \frac{V_e}{20}(4\overline{G} + G_n)\,; \tag{88}$$

Equation $[Q]\frac{\mathrm{d}\{c\}}{\mathrm{d}t} + [S]\{c\} + \{f\} = -\{Q^D\}$; (85) is integrated in time using a finite difference scheme, resulting in the following system of equations:

$$\begin{aligned} &[Q]_{j+\varepsilon} \frac{\{c\}_{j+1} - \{c\}_j}{\Delta t_j} + \varepsilon [S]_{j+1}\{c\}_{j+1} + (1-\varepsilon)[S]_j \{c\}_j + \\ &\quad + \varepsilon \{f\}_{j+1} + (1-\varepsilon)\{f\}_j = 0; \end{aligned} \tag{89}$$

with*j+1* and *j* being the current and previous time levels, respectively, and ε the time weighing coefficient.

Option 2: Finite element hybrid

The diffusive and advective mass/solute flux between two adjacent soil nodes of the microscopic CDE (Eq. **Erreur ! Source du renvoi introuvable.**), involves radial derivatives and a time derivative that is solved using a Crank–Nicolson time stepping approach. We discretized Equation **Fehler! Verweisquelle konnte nicht gefunden werden.** Using a three-point finite difference scheme applied to the grid, as follows:

$$\frac{(b+\theta_i^{j+1})c_i^{j+1}-(b+\theta_i^j)c_i^j}{\Delta t^j}$$

$$= \alpha\left(\theta_i^{j+1} {D_m}_i^{j+1}\tau(\theta)\left[\frac{(c_{i+1}^{j+1}-2c_i^{j+1}+c_{i-1}^{j+1})}{(\Delta r)^2}+\frac{1}{r_i}\frac{(c_{i+1}^{j+1}-c_{i-1}^{j+1})}{2\Delta r}\right]\right.$$

$$\left.-\frac{(q_{i+1}^{j+1}c_{i+1}^{j+1}-q_{i-1}^{j+1}c_{i-1}^{j+1})}{2\Delta r}-\frac{q_i^{j+1}c_i^{j+1}}{r_i}\right)$$

$$+\ (1-\alpha)\left(\theta_i^j {D_m}_i^j\tau(\theta)\left[\frac{(c_{i+1}^j-2c_i^j+c_{i-1}^j)}{(\Delta r)^2}+\frac{1}{r_i}\frac{(c_{i+1}^j-c_{i-1}^j)}{2\Delta r}\right]\right.$$

$$\left.-\frac{(q_{i+1}^j c_{i+1}^j-q_{i-1}^j c_{i-1}^j)}{2\Delta r}-\frac{q_i^j c_i^j}{r_i}\right)$$

$$(i \geq 1, t > 0)\ \ (90)$$

where the index i is the soil segment number (i=1,2,...,n), identifies the mesh point in r-direction. i= 1 refers to the first segment of the soil and n is the outer segment of the soil domain, j refers to the time step. Δr is the distance between consecutive nodes. α is a weighting constant between 0 and 1. The scheme is said to be fully explicit for α=0 and fully implicit for α=1 while a Crank–Nicolson central in time scheme is derived if α =1⁄2.

At the outer boundary of the 1-D axisymmetric model, nutrient and water fluxes need to be defined whereas the sink terms for water and nutrient uptake need to be defined in the soil voxels of the 3-D grid.

Option 3: Particle tracking macroscopic (from Schroeder, 2013)

The solute modules are connected to the water modules either by the velocity field alone, given by the Richards equation, or in addition by a feedback coupling from solute to water (XX). Soil solute transport can be solved by the model PARTRACE. This model uses a random walk particle tracking (RWPT) algorithm(Bechtold et al. 2011), where the CDE is defined as an equivalent stochastic differential equation (Tompson and Gelhar 1990):

$$X(t+\Delta t) = X(t) + \left[\mathbf{u}\big(X(t)\big) + \nabla\cdot\mathbf{D}\big(X(t)\big) + \frac{\mathbf{D}\big(X(t)\big)}{\theta\big(X(t)\big)}\nabla\theta\big(X(t)\big)\right]\Delta t + \mathbf{B}\big(X(t)\big)\xi\sqrt{\Delta t}$$

Here, X is the coordinate vector [L] and ξ is a vector of three random numbers, normally distributed with zero mean and unit variance. **D** is the local-scale dispersion tensor [L^2 T^{-1}] with equation (Bear, J. 1972):

$$\boldsymbol{D} = (\alpha_T \|u\| + D_m)\boldsymbol{I} + (\alpha_L - \alpha_T)\frac{uu^T}{\|u\|'}$$

whereα_T and α_D are the transversal and longitudinal dispersivities [L] respectively, D_m is the effective molecular diffusion coefficient [L^2 T^{-1}] and I is the identity matrix.

The dispersion displacement matrix **B** is related to the dispersion tensor by $2\mathbf{D} = \mathbf{B}^T\boldsymbol{B}$(Salamon, Fernàndez-Garcia, and Gómez-Hernández 2006) and defined for an isotropic porous media (Lichtner, Kelkar, and Robinson 2002):

$$\mathbf{B} = \begin{pmatrix} \frac{u_x}{\|u\|}L & \frac{u_x u_y}{\|u\|\sqrt{u_x^2 + u_y^2}}T & \frac{u_y}{\|u\|\sqrt{u_x^2 + u_y^2}}T \\ \frac{u_y}{\|u\|}L & \frac{u_y u_z}{\|u\|\sqrt{u_x^2 + u_y^2}}T & \frac{u_x}{\|u\|\sqrt{u_x^2 + u_y^2}}T \\ \frac{u_z}{\|u\|}L & \frac{\sqrt{u_x^2 + u_y^2}}{\|u\|}T & 0 \end{pmatrix}$$

where*u*$_x$, *u*$_y$, and *u*$_z$ are the velocities in x-, y-, and z-direction, ǁuǁ is the Euclidean norm of the velocity $L = \sqrt{2(\alpha_L \|u\| D_m)}$and $= \sqrt{2(\alpha_T \|u\| D_m)}$.

To represent the movement of solute mass through the soil, a large number of solute particles is defined. Each solute particle has a predefined mass and is tracked through the soil by Eq. **Fehler! Verweisquelle konnte nicht gefunden werden.**. The advective (deterministic) movement of one particle is caused by the term $\left[u\big(X(t)\big) + \nabla \cdot D\big(X(t)\big) + \frac{D(X(t))}{\theta(X(t))}\nabla\theta\big(X(t)\big)\right]$, which is the sum of the fluid velocity and a velocity correction that originates from the spatial variability of D andθ(Bechtold et al. 2011; LaBolle et al. 2000). The last term$B\big(X(t)\big)\xi\sqrt{\Delta t}$represents the dispersive (stochastic) movement of a solute particle.

The RWPT is a Lagrangian method which solves the transport equation directly and is virtually free of numerical dispersion and artificial oscillations (Salamon, Fernàndez-Garcia, and Gómez-Hernández 2006). Another advantage of the method is the significantly smaller computational times for very large amount of cells with strong heterogeneities, compared to the Eulerian approach. Like every particle tracking method, PARTRACE conserves mass by definition and is supposed to handle high concentration gradients better than a Galerkin finite element method. These gradients occur, for example, at evaporation surface (Bechtold et al. 2011) or at root surfaces where solutes are excluded by the plant or actively taken up. The disadvantages of the RWPT algorithm are the limitations of simulation of non-equilibrium processes and multi-species kinetic reactions(Salamon, Fernàndez-Garcia, and Gómez-Hernández 2006). Boundary conditions can be applied either as concentration inflow or outflow, as reflection, where no solute is entering or leaving the system and solute particles are reflected at the domain border, or as periodic, where solute particles leave the soil domain at one side, and enter it at the opposite side again.

2.2.2. Numerical solution of transport/production within the root system (adapted from Huber et al. 2014)

Solute transport in the root system is simulated using a particle tracking algorithm. At each time step, **Δt**, in any particular segment i, particles with a user defined mass can be created (e.g., depending on the pressure head inside the root segment i).Each particleistransported with the root water flow towards the root collar. By knowing the water flow through the root system, the xylem flowvelocity v_i[m^3 d^{-1}] within a segment *i* is calculated as

$$v_i = \frac{J_{x,i}}{A_i} \tag{94}$$

whereA_i[m^2] is the cross section area of the xylem vessel *i*. At each time step Δt, each particle is moved by a displacement vector estimated for each segment *i*:

$$X(t + \Delta t) = X(t) + \Delta t \cdot v_i \tag{95}$$

where X(t) and X(t+Δt) are the current and the new position of the particle inside the root, respectively (see also Section 1.2.3) until the collar node (considered as connected to a big leaf) is reached. The particle movement is advective and has no diffusive or dispersive component.

The mass of solute is calculated for each segment as:

$$M_{x,i} = mass\ arriving + source \tag{96}$$

In case chemical signaling is modeled, the source term for signaling is defined as follow. At eachtime step, Δt, in any particular segment i, when|hRoot,i| ≥ |h0|, one particle with the massMsignal,i*Δt is generated. The solute concentration in a given root segment is estimated by

$$C_{x,i} = \frac{M_{x,i}}{V_{x,i}} \tag{97}$$

withV_{xi}, the volume of the xylem vessel*i*.

2.2.3. Pesticide uptake

2.2.3.1. Soil transport and uptake (to be completed)

Current implementation of solute uptake takes a lumped root model into account and uses a coupled version of R-SWMS and ParTrace, where the latter is a particle tracking algorithm accounting for the solutes within the soil. R-SWMS delivers the velocity field and the current water contents on a voxel or a nodal basis. This velocity field is used by ParTrace to move the solute particles within the soil. The resulting concentrations in the soil voxels are handed back to R-SWMS, where root solute uptake is calculated as a mass flux based on the gradient between root and soil solute concentration. The uptake is handed back to ParTrace, where the particles within the affected voxels are reduced in their mass.

$$m_u = \left(c_{S,f,i} - c_{R,f,j}\right) A_{R,j} P_j dt + \varepsilon_j J_{w,i} c_{S,i} dt, \qquad (98)$$

where m_u [M] is the root solute uptake per time step dt, $c_{S,f,i}$ [M L^{-3}] the solved concentration in the surrounding soil voxel i, $c_{R,f,j}$ [M L^{-3}] the mobile concentration within the root segment j, $A_{R,j}$ [L^2] the circumferential area of the root segment j, P_j [L T^{-1}] the age dependent permeability of the root membrane, ε_j [-] the fraction of apoplastic uptake, dependent on the root age, $J_{w,i}$ [L^3 T^{-1}] the radial root water uptake from a single root segment, and dt [T] the current time step size.

The root membrane permeability P [L T^{-1}] is related to the diffusion by the diffusion length (membrane thickness) Δx [L] and the relative solubility of the solute in the membrane K [-]. K can be approximated by K_{ow}(Trapp 2004),

$$P = \frac{DK}{\Delta x} \qquad (99)$$

The solute particles are transported via advective water flow within the roots. As solutes can also sorb to the lipid phase within the roots, a retardation factor, depending either on a linear or a Freundlich isotherm is introduced to slow down the velocity of the solute particles.

The main advantage with respect to the particle tracking method is that for this method of describing the sorption, no particles are lost and immobilized but all particles are slower.

$$v_{r,i} = \frac{J_{x,i}}{RA_{x,i}} \qquad (100)$$

where $v_{r,i}$ [L T^{-1}] is the transport velocity of the particle, $J_{x,i,}$ [L^3 T^{-1}] the axial water flow through the root segment, $A_{x,i}$ [L^2] the cross section of the segment, and R [-] the retardation factor which can take any positive number.

Sorption can be described as solutes that take one out of two different states: sorbed and dissolved. Only the dissolved solutes can move freely within the medium. For a particle tracking algorithm there are two ways to describe sorption. Either a single particle can be in its sorbed or dissolved state, which would, in case of high sorption, lead to a substantial reduction of the amount of particles moving in the system and thus to numerical dispersion effects. The second option is that all particles move at a slower velocity, which will conserve the number of particles. Due to this reason, we chose to implement the latter option.

The retardation is equal to the fraction of solutes in the dissolved state. Assuming that the sorption follows the Freundlich isotherm, the total concentration retardation needs to be calculated iteratively by

$$C_{R,f,i+1} = C_{R,f,i} - \frac{C_{R,f,i} + K_{D,R} {C_{R,f,i}}^{n} - C_{R,T,i}}{1 + nK_{D,R} C_{R,f,i}^{(n-1)}}, \qquad (101)$$

where i is the iteration number, $C_{R,T}$ [M L^{-3}] is the total and $C_{R,f}$ dissolved concentrations, respectively. $K_{D,R}$ [L^3 M^{-1}] and n [-] are Freundlich coefficient and exponent, respectively. The iteration is stopped when the second term on the right hand side reaches a sufficiently small value. In R-SWMS this is solved using the Newton method with the convergence criterion $|f/C_{R,f}| \leq 1e\text{-}5$,

$$f = C_{R,f} + C_{R,s} - C_{R,T} = C_{R,f} + K_{D,R} C_{R,f}^{n} - C_{R,T} \qquad (102)$$

where$C_{R,T}$ [M L^{-3}] is the total and $C_{R,f}$and$C_{R,s}$ the dissolved and the sorbed concentrations, respectively. $K_{D,R}$ [L^3 M^{-1}] and n [-] are the parameters to describe the Freundlich isotherm. To do so all concentrations are given as mass per volume of water, and $K_{D,R}$ = $K_{D,R}$/rho_R.

In R-SWMS this equation is solved for $C_{R,f}$, using the Newton method with the convergence criterium $|f/c_{R,f}| \leq 1e^{-5}$. More details of the implementation can be found in Neuendorf (1997).

2.2.3.2. Pesticide transport in roots

The root system used in pesticide uptake and hormonal signaling is assembled according to the Doussan root system. However, to model solute transport in the root the first root segment needs to be defined as a buffer segment, which stores the particles exiting the root system (into the above ground plant). Therefore, the first segment is defined in the input file by a very small extend and volume, so the influence of the segment to the root water uptake and solute uptake can be neglected. Its only connection is towards a second root segment, which is located underneath. There cannot be any branching at the first root segment, as it will create numerical errors. The buffer volume is defined separately in the source code. For pesticide uptake the buffer volume is defined as the total root volume. For hormonal signaling the buffer size can be defined in multiple ways. In the input file CondRoot.in a multiplication factor for the root volume can be defined, which determines the buffer size. If the factor is not greater than zero, the buffer volume is defined as the volume of the first root segment.

The buffer volume stores the solute particles until they exit the system with the transpiration flux. The mass transport into the buffer is described for each time step by

$$m_{col,j} = m_{col,j-1} + m_{in,j} + m_{in,col,j} \qquad (103)$$

$$c_{col,j} = m_{col,j}/V_b \qquad (104)$$

For hormonal signaling, the hormones can leave the root buffer via the transpirations stream. This is not implemented for pesticide uptake.

$$m_{col,j} = max(m_{col,j} - c_{col,j} * T_{act,j} * dt, 0) \qquad (105)$$

Where V_b [L^3] = V_r [L^3]are the volume of buffer and root respectively, m_{col} is the solute mass in the buffer in the current time step j and the previous time step j-i, m_{in} and $m_{in,col}$ the mass added from the root system and the root buffer (the first root segment).

Particle uptake:

In each time step the amount of mass uptake is calculated and a constant amount of particles is created and inserted into each root segment. The particles are associated with a mass, which is defined by

$$m_{ci,j} = m_{adv,ci,j} + m_{diff,ci,j} \tag{106}$$

where mass$_{adv}$ and mass$_{diff}$ are defined for each element c and time step j by

$$m_{adv,ci,j} = \frac{\varepsilon J_{r,i,j} C_{c,j} w_{sub,i,j}}{SegPerCube} * dt \tag{107}$$

and

$$m_{diff,ci,j} = \frac{(C_{c,j} - (C_{R,T,i,j} - S_R))A_i w_{sub,i,j}}{SegPerCube} P_i * dt \tag{108}$$

where $\boldsymbol{\varepsilon}$ is the fraction of convective solute transport (PassRoot in CondRoot.in), $\boldsymbol{J_{r,i,j}}$ the radial root flux, $\boldsymbol{C_{c,j}}$ soil concentration per element, $\boldsymbol{w_{sub,i,j}}$ the weighted segment length when segment in multiple soil voxels. SegPerCube is one (is not needed). $\boldsymbol{C_{R,f,i,j}}$ [M/L^3] is the concentration inside the root segment, S_R the sorbed root segment concentration (linear or freundlich) , $\boldsymbol{w_{sub,i,j}}$the weighted segment length in the respective soil voxel and P_i the root segment permeability [L/T].

Particles are inserted into all segments greater than one, mass which would enter the first segment is directly transferred to m$_{col}$, the mass in the root buffer.

Particle transport:

The particles attributes in the root system are referred to by a pointer. Pointers declare the particle number, the mass associated to a particle and the position in the root system.

Particles are moved within the root system according to the axial flow velocity. Particle movement can be slowed down by application of two different sorption terms; the retardation is calculated in the subroutine 'CalculateRootRetard'.

For linear sorption

$$R = \frac{1}{(\theta_R + \rho_R K_{D,R})}$$

where R is a constant retardation factor, θ_R, ρ_R, $K_{D,R}$are root water content, wet root density, and root sorption parameter, respectively. They can be specified in CondRoot.in.

Using the Freundlich sorption isotherm retardation needs to be obtained for each root segment separately, due to its non-linear relation. Freundlich sorption needs to be solved iteratively and was implemented in R-SWMS according to {Neuendorf, 1997 #64}:

$$c_{f,i,j+1} = c_{f,i,j} - \frac{c_{f,i,j}+kc_{f,i,j}{}^{n}-c_j}{1+nkc_{f,i,j}^{(n-1)}}\ , \tag{109}$$

$$R_{i,j+1} = \frac{c_{f,i,j+1}}{c_{i,j+1}}, \tag{110}$$

where $c_{f,i,j+1}$ is the solved solute fraction for the segment i and the time step +1j, k and n are sorption parameters and $c_{f,i,j}$ is the root segment concentration of the previous time step. $R_{i,j}$ is the retardation factor [-] for the segment i and the time step j.

Particles are moved along a root segment by a relative segment distance by

$$d_{p,i,j} = v_{r,p,i} R_{p,i} dt \tag{111}$$

Where d [L] is the particle displacement in the root segment in the time step j, v_r the velocity [L/T] in the root segment i in which particle p is located, R the retardation factor [-] (for root segment i and particle p) and dt the time step size.

The new relative particle position is obtained by

$$Pos_{p,j} = Pos_{p,j-1} + \frac{d_{p,i,j}}{1-k} fork \leq 1$$

where

$$k_{j,p} = \frac{l_{i,p} - Pos_{p,j-1}}{d_{p,i,j}}(1 - k_{j-1,p}) \tag{112}$$

if$d_{p,i,j} \leq 0$ solute movement is downwards and the position new relative particle position is obtained by

$$k_{j,p} = -\frac{Pos_{p,j-1}}{d_{p,i,j}}(1 - k_{j-1,p})$$

When the relative position in a root segment is greater one, the particle is transferred to the next segment and removed from the root when the first segment is reached; there it enters the buffer volume.

2.2.4. Two-way coupled flow and transportin the soil-root system: effect of salinity on root water uptake (adapted from Schröder et al., 2014)

Since the transport influences the distribution of osmotic potentials and therefore root water uptake, the transport process has an impact on the flow. This back-coupling betweenwater flowand solute transport within one time step was implemented in an explicitway, using the solute concentrations (transformed to osmotic potentials) at the beginning of one time-step to calculate the radial flows at the soil-root interface. Thus, the radial flow from soil into the root, and thusalso water sink termin the soil at time t+Δt, is based on an updated matric potential at time t+Δt with an osmotic component derived from the concentration at time t.

The solute concentration c [N L-3] in the model is transferred to the osmotic potential ho [L] according to:

$$h_o = \beta \cdot c$$

(108)

In Schröder (2013), the parameter β was calculatedfrom data of Shani and Ben-Gal (2005) and Hamza and Aylmore (1992). However, a different value can be set manually in the code. The parameter β is called Fac in the R-SWMS code and can be found in the subroutine Salinity in SolutePartrace.f90.

Once solute concentrations are transformed into osmotic potentials, the osmotic potentials of element/s surrounding a specific element are added to the water potential gradient. Therefore, the radial flux density entering a root segment described previously is transformed into:

$$q_r = L_r\left((h_s - h_x) + \sigma(h_{o,s} - h_{o,x})\right)$$

(109)

where $h_{o,s}$ is the osmotic potential at the soil-root interface and $h_{o,x}$ is the osmotic potential in the xylem. The parameter σ is the reflection coefficient and can vary between zero and one. It represents the effectiveness of the membrane complex to selectively allow water flow but not salt transport across the membrane. When σ is 1, we have a case of total exclusion and no solute is allowed to cross the membrane. On the contrary, when σ is 0, all solute particles can enter the root with the water radial flux and therefore osmotic potentials do not drive water flow across the membrane. This parameter can be manually set in the code: it corresponds with variable sigma in the Modules.f90 file.

If simulations with salinity effect want to be performed, the l_salinity flag must be set to true in the input file control.in.

3. Input files

Four types of input files exist, for fourdifferent programs: input files for the grid generation, R-SWMS, RootTyp and ParTrace.

All input files are explained in this chapter of the users' guide and are gathered in different sections according to the process or processes in which they are needed. At the beginning of each section, the input file or files that are going to be described is/are highlighted. An overview of all the input files can be found in Table 1 -Table 4, which also explains their basic functions and points out when an input file is or not dispensable.

Table 1 Overview of input file used to run the grid generator

Input file	Mainfunction	Necessaryfor	Accesses
Mesh.in	File to run the grid generator program	When new soil grid needs to be generated	control.in soil.in

Table 2 Overview of the input files used to run R-SWMS.

Input file type		Input file	Mainfunction	Necessaryfor
General inputfile		Control.in	Contains general parameters and processes	Always
Grid files (can be generated with the grid generator)		Nodes.in (nodes2.in)	Geometry and initial boundary conditions of the nodes	Always
		Grid.in (grid2.in)	Definition of voxels by its nodes	
Root system	Root description	RootSys	Description of initial root structure	Root growth and root water uptake
	Root growth	Root.in	Root growthparamenters	Root growth with the Somma model
	Root growth	RootBox.in	Root growthparamenters	Root growth with the RootBox model
	Root wateruptake	BCroot.in	Root collarboundaryconditions	Root wateruptake
		CondRoot.in	Hydraulicrootproperties	
Water flow		Soil.in	Parameterizedsoilhydr	Always

		aulicproperties	
	Soiltab.in	Tabulatedsoilhydraulic properties	
	Bc.in	Soilboundaryconditions	Water and solute transport
Solutetransport	Chem.in	Solute transportparameters	Solute transport
Plant characteristics	Plant.in	Shoot plant parameters	Functionalshootgrowth
Temperature	Temp.in	Defines parameters to account for temperature effects	Temperatureeffects

Table 3 Overview of input files used to run RootTyp

Input files	**Mainfunction**	**Necessaryfor**
Limites.txt, noeuds.txt, param.txt, sol.txt	Files torunRootTyp	Root growth and root water uptake (if no RootSys file exists)

Table 4 Overview of input files used to run Partrace

Input files	**Main function**	**Necessary for**
Input_PARTRACE.pTraceInpV12	Contains general parameters to run Partrace	Solute transport (if CDE in R-SWMS is not used)
decay.in	Defines the type of particle decay	
Geometry_X.in	Nodal information on the soil grid	
out.veloci.00001 out.water_content.00001	Nodal information on water content and velocity field	
partrace.index	Indexing to define soil layers or blocks with different properties	
partrace.soilprop	Define soil properties and diffusion parameters	

In the Somma model different procedures may be performed following the type of simulations which is needed and the types of input files which are used. In contrast, most of R-SWMS' options are defined in the input file control.in and the concept of level was replaced by logical flags which represent the chosen options, this gives R-SWMS more flexibility and clarity.

3.1. Building the soil grid

nodes.in, grid.in, mesh.in

Two files are needed for describing the soil domain: **nodes.in** and **grid.in**. Both can be built with the fortran file Grid_SWMS.f90 and the input file **mesh.in**. First, Grid_SWMS.f90 has to be compiled in R-SWMS6/src_R-SWMS/ via the command

```
gfortran -O0 -o grid Grid_SWMS.f90
```

to generate the program grid. The program not only accesses **mesh.in** with the basic grid information, but also **control.in** and **soil.in**. These two latter files provide the information if the grid should be periodic or not (control.in, continous domain), if solute transport is considered (control.in) and how many soil materials are present in the domain (soil.in). So far the automatic grid generator can only handle two soil materials and generate simple configurations (see below). For more sophisticated shapes, nodes.in can be defined with e.g. matlab. Descriptions of control.in and soil.in are found below in more detail.

In **mesh.in** (Fig. XX), the user must specify the grid resolution, the number of “blocks” (elements) in each direction, the coordinate of the first node, the type and value of initial water potential conditions (1=constant, 2=hydrostatic equilibrium, 3=from file), the split type if 2 materials are defined in soil.in and the initial solute concentration if he wants to simulate solute transport. **The z-coordinate of the soil surface should always be set to 0!**

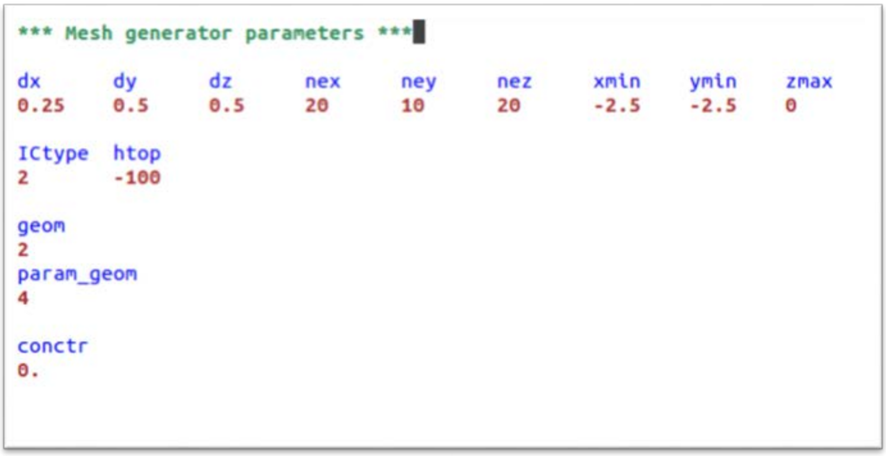
The output file **grid.in** gives the element information. An element consists of 8 nodes with the first node being the top, left, front node. One line in grid.in defines the number of the element and the node numbers belonging to that element. The output **nodes.in** stores the information about soil nodes. The second column is the material number. If macropores are included, material number 2 **always** refers to the macropores.Further to the right the nodal coordinates are given, followed by the pressure head and the solute concentration.

To perform simulations of root growth and root water uptake in soil with biopores (or with similar small-scale soil structures) a fine grid resolution must be used. The calculation of soil water flow in high resolution soil grids, however, is computationally extremely expensive. For this reason, it is possible to use two soil grids (flag ‘ltwo_grids’ in control.in): The coarse soil grid, which is defined by nodes.in and grid.in, is used to compute soil water flow. The second soil grid, which is defined by nodes2.in and grid2.in, is used to compute soil penetration resistance. The coarse and the fine grid must have the same total size. If two grids are used, material numbers are taken from the second grid, thus the file nodes2.in.

Soil penetration resistance is a function of both bulk density and soil water potential (Whalley et al. 2007). Soil water potential, however, is calculated on the coarse grid. For the calculation of soil penetration resistance, every fine grid node is therefore given the soil water potential of the closest coarse grid node.

Please, payattentionto:

- The upper left front node is node number 1. The numbering first goes to x, then to y, and finally to z direction.
- The central node at the top plane should have the coordinates (0,0,0).
- The spatial resolution of the soil grid should carefully be chosen, to avoid large error at the soil root interface.
- By default the z axis is positive upward/negative downward.
- mesh.in allows to build (horizontal or vertical) split root setups as well as cylindrical domains. For this, more than one material has to be defined in soil.in. Material number 2 has to be the non-conductive or filling material: either for the split layer or for the ‘cladding’ of the cylinder.



```
*** Mesh generator parameters ***

dx      dy      dz      nex     ney     nez     xmin    ymin    zmax
0.25    0.5     0.5     20      10      20      -2.5    -2.5    0

ICtype  htop
2       -100

geom
2
param_geom
4

conctr
0.
```

Figure 4: Example of mesh.in input file.

Line	Record	Type	Variable	Unit	Description
2	1	Real	dx	Cm	size of the element in the x-direction
2	2	Real	dy	Cm	size of the element in the y-direction
2	3	Real	dz	Cm	size of the element in the z-direction
2	4	Int	nex	-	Number of elements in the x-direction
2	5	Int	ney	-	Number of elements in the y-direction
2	6	Int	nez	-	Number of elements in the z-direction
2	7	Real	xmin	cm	X-position of the first node
2	8	Real	ymin	cm	Y-position of the first node
2	9	Real	zmax	cm	Z-position of the first node
5	1	Int	Ictype	-	Type of the initial pressure head condition : constant head (1), equilibrium (2), from old outfem file (3)
5	2	Real	htop	cm	Value of the top soil pressure head (Ictype =1 or 2)
8	1	Int	geom	-	Geometry of the domain (if nMat > 1) : horizontal split (1), vertical split (2), cylinder (3)
10	1	Int	param_geom : zlim/width/radius	cm	Geometrical information for geom
13	1	Real	conctr	µmol/cm³	Initial constant soil solute concentration

3.2. R-SWMS Input

3.2.1. What are the initial and final simulation times in R-SWMS?

In R-SWMS the time 0 is always the time at which the seed has been planted. When simulating a root which has already a certain age, the initial time for the simulation can be larger than 0. The initial time that is taken by the code is defined by the root system (e.g, **RootSys** file):

- when **Roottyp** is used for generating the root structure, initial time is the time defined in param.txt (first argument: number of days of growth simulation)
- when an input file for root architecture (**RootSys**) is given, the initial time is written as first argument (*initial time*).
- when **RootSys** or **RootTyp** architectures are not used (lno_archi=true), the initial time is given bythe first time of the boundary conditions defined in BCroot.in (tBCr(1)).
- In addition, if lno_archi=true and lretry=true (to start over an old simulation), the initial time is defined through the variable last_Out in control.in. Last_out is the number of the last outfem file, which was created from the previous simulation (before its end). The code will read outfem.last_out file and start over from that time ontout(last_out).

Take care when defining output times in **control.in** that the initial time may differ from 0. Also when defining the boundary conditions for root (**BCroot.in**), take care that these comply with your root age and simulation time window.

The final time is the maximum observation time prescribed in control.in for outfem or for outroot.

1.2.2. General input

Control.in gives general information about units, error levels, and outputs. The units should be kept in cm and days. In order to simulate root growth with Somma, dtRoot (time step for root growth, line 18) should be adjusted to a lower level (e.g. 2.0 days). Several changes are made as compared to the Somma model control.in input files:

- Relative (=1) or absolute (=2) convergence criteria can be chosen. In the former case, the second figure is the relative criterion.
- Absolute errors are defined for the soil PH (errH), the soil water content (errWC), root water potential (errRootPH) and sink term (errSink)
- Z-profiles: when depth averaged profiles for WC, or PH or Sink term are needed, put 1. The second value sets the time discretization in the output. If dtprof=999, z profiles are given at each time step.
- Choice of the root water uptake model.
- Choice of the source of the root architecture.
- Choice of the root growth model.
- Use of assimilate allocation (for leaf growth and evolution of potential transpiration).
- Useofsolutetransport.
- Use of domain periodicity. In this case, you can perform “field-like” simulations with interaction between several identical root systems in a periodic simulation domain (periodicity for soil water flow at vertical boundaries, for root architecture(s) and root water flow).
- Useofvtkoutputs.
- Use of partrace outputs (for partracevisualisation).
- Use of direct Doussan if Doussan model is used.
- Averaging method: original R-SWMS where the water potential at the soil root interface is based on the weighted average between soil voxel nodes.
- Use memory reduction if averaging method is used.

An example of a control.in file is given in Figure x.

control.in

Table 5

line	record	type	variable	Unit	description
1-8	-	Char	hed	-	comment lines / heading
9	1	int	itMax	-	maximum number of iterations for the soil calculation
9	2	int	itMaxRoot	-	maximum number of iterations for the root calculation
10-11	-	Char	hed	-	comment lines / heading
12	1	logical	RelEps	-	do you want to use relative convergence criterium? (t or f)
12	2	int	factorRelEpsT	-	factor of convergence criteria, may not be zero
13-14	-	Char	hed	-	comment lines / heading
15	1	real	errH	cm	absolute error of the soil water potential
15	2	real	errwc	cm^3/cm^3	absolute error of the soil water content
15	3	real	errRootPH	cm	absolute error of the root water potential
15	4	real	errsink	cm	absolute error of the sink term
16-17	-	Char	hed	-	comment lines / heading
18	1	real	dt	d	time step , soil

18	2	real	dtMin	d	minimum time step , soil
18	3	real	dtMax	d	maximum time step , soil
18	4	real	FacInc	-	?
18	5	real	FacDec	-	?
18	6	real	dtRoot	d	time step, root
19-20	-	Char	hed	-	comment lines / heading
21	1	logical	Lretry	-	do you want to continue an old simulation?
21	2	real	lastOut	d	time point, at which new simulation should start (old finished)
22-24	-	Char	hed	-	comment lines / heading
25	1	int	nOut	-	Number of FEM- and Root- Outputs
26-27	-	Char	hed	-	comment lines / heading
28	1	real	tOut	d	time points of FEM- and Root- Outputs
29-30	-	Char	hed	-	comment lines / heading
31	1	logical	lvtk		do you want to use vtk files as a output?
31	2	logical	lOutpartrace		do you want to use partrace files as a output?
31	3	logical	profOK	-	do you want time discretization for the z-profiles?
32-33	-	Char	hed	-	comment lines / heading

34	1	real	dtprof	d	time discretization for the z-profiles (if profOK=1), if dtprof=999 --> all time steps
35-38	-	Char	hed	-	comment lines / heading
39	1	logical	lno_RWU	-	do you want to include Root water uptake?
39	2	logical	lFeddes	-	do you want to include Feddes function?
39	3	logical	lDoussan	-	do you want to include Doussan model?
39	4	logical	lCouvreur	-	do you want to include Couvreur function?
39	5	logical	lsinkCube	-	do you want to calculate the sink based on soil cubes (voxels)?
40-42	-	Char	hed	-	comment lines / heading
43	1	logical	lno_Archi	-	do you want not to use any root architecture?
43	2	logical	lrootsys	-	do you want to use an initial root system defiend in a rootSys file?
43	3	logical	lroottyp	-	do you want to use rootTyp?
44-46	-	Char	hed	-	comment lines / heading
47	1	logical	lno_root_growth	-	do you want not to use a root growth model?
47	2	logical	lRootTyp_growth	-	do you want to use roottyp for root growth?
47	3	logical	lSomma_growth	-	do you want to use Somma root growth?
47	3	logical	lRootBox_growth	-	do you want to use root growth with RootBox?
47	4	logical	lUpdate_growth	-	If there is no root growth but that you still want to update root parameters (Kr,Kx) with time

48-50	-	Char	hed	-	comment lines / heading
51	1	logical	lcalloc	-	do you want to include carbon allocation?
51	2	logical	lchem	-	do you want to include solute transport (chem.in file)?
51	3	logical	ltemp	-	do you want to include the temperature (temp.in file)?
51	4	logical	lcontinu	-	do you want to have a periodic boundary condition?
51	5	logical	lsalinity	-	do you want to include salinity (if compiled with ParTrace)?
51	6	logical	lRhizo	-	do you want to include rhizosphere dynamics?
51	7	logical	lClimate	-	do you want to use climatic input data (with in/Climate.in)?
51	8	logical	ltwo_grids	-	do you want to include two soil grids (for root growth in macropores)?
52-54	-	Char	hed	-	comment lines / heading
55	1	logical	ldirect	-	do you want to use the direct solver method?
55	2	logical	laveraging	-	do you want to use averaging solver method?
55	3	logical	ana_aan	-	do you want to use memory reduction solver method?
55	4	logical	ltab	-	doesn't work

1.2.3. Root architecture

The root architecture for R-SWMS is described in a file called RootSys, which contains the information about root node locations and their logical connections. There are three ways to get the root architecture for R-SWMS: Importing and existing root architecture via a **RootSys**file, generating a RootSys file with the root growth model RootTyp before running the R-SWMS simulation, or using dynamic root growth with the Somma root growth model. The information, which option is used, has to be defined in **control.in**:

- Lno_archi= true: there is neither Rootsys, nor growth model which is used. In that case, no root water uptake modeling can be performed and only soil water and solute transport without plant is simulated.
- Lrrt=true: rootTyp will be used. rootTyp input files should be given (param.txt, etc.) and no initial rootsys file will be read.
- Lrrs=true: an initial RootSys file will be read.

3.2.3.1 Importing and existing root architecture

RootSys

The root architectural file contains a common block with information about root system age (line 2), number of seeds (=1), and location of the seed (usually 0,0). In line 11 the root and shoot dry mass as well as the leaf area (cm2) are written. These parameters are only important if carbon allocation in control.in is set to true. The average root sensed soil strength and solute concentration in line 13 are only important if (Somma) root growth is activated. In general a root system can be defined as axes (order = 1), originating at the seed, and laterals (higher orders). The second order originates from a first order root and so forth. The following information gives the number of axes, branches and segment records.

The next block contains the segment information. A segment is defined between a root node and its previous root node (Figure 3). A segment is defined with an ID, nodal coordinates, previous segment node (node #0 is the collar or seed), the branch order and number, length of the segment, its lateral surface and mass, and in the subsequent line the origination time.

The third block contains the tip information, where a tip is the growing end of a branch. A tip is defined similar to a segment with its own ID, the coordinates, the segment number behind the tip, the root order and branch number, the total branch length, and the axis number. In case of (Somma) root growth the next lines define locations along the root branch that are ‘ripe’ for branching. This information is given in time units. Thus, if the current runtime is larger than the establishing time AND the branching distance is larger than the distance defined in root.in

, a branch will be initiated.

Table 6

Line	Record	Type	Variable	Unit	Description
1		Char			Comment lines/Headings
2	1	Real(sp)	t	[d]	age of the root system; also starting time of simulation
3-4		Char			Comment lines/Headings
5	1	Integer	nplant	[-]	number of plants; can only be set to 1 at the moment
6-7		Char			Comment lines/Headings
8	1	Integer	iplant	[-]	plant ID
8	2,3	Real(sp)	xplant, yplant		with these two the complete root architecture can be shifted within the domain [x_new = x(RootSys)+xplant]
9-10		Char			Comment lines/Headings
11	1-3	Real(sp)	mroot, mshoot, LA	[g], [g], [cm^2]	root and shoot mass, leaf area
12-13		Char			Comment lines/Headings
14	1-2	Real(dp)	sAvg, cAvg	[MPa], [µmol cm^3]	Average soil strength and solute concentration sensed by the root system – all the voxels surrounding the roots are taken into account for averaging
15-16		Char			Comment lines/Headings
17	1	Integer	naxes	[-]	Number of axes of the root system
18-19		Char			Comment lines/Headings
20	1	Integer	nbr	[-]	Number of branches of the root system
21-22		Char			Comment lines/Headings
23	1	Integer	nrec	[-]	Number of root segments
24-26		Char			Comment lines/Headings
27	1	Integer	irec	[-]	Segment ID
	2-4	Real(sp)	xs, ys, zs	[cm]	x, y, z coordinates of root node
	5	Integer	irecpr	[-]	ID of previous segment
	6	Integer	ordseg	[-]	segment order (1-3)
	7	Integer	ibrseg	[-]	branch number, which the segment belongs to
	8	Real(dp)	seglen	[cm]	segment length; calculated between root node and previous segment

	9	Real(dp)	segsur	[cm^2]	lateral surface of the segment
	10	Real(dp)	segmas	[g]	segment mass
	11	Integer	segmat	[-]	Segment material
28	1	Real(sp)	timorg	[d]	origination time of segment
29-30		Char			Comment lines/Headings
31	1	Integer	ngrow	[-]	number of growing tips; has to be equal to nbr
32-35		Char			Comment lines/Headings
36	1	Integer	igrow	[-]	ID of root tip
	2-4	Real(sp)	xg, yg, zg	[cm]	x, y, z coordinates of root tip
	5	Integer	irecsg	[-]	previous root segment
	6	Integer	ordgrw	[-]	growth order (1-3)
	7	Integer	ibrgrw	[-]	branch number that this root tip belongs to
	8	Real(dp)	brlgth	[cm]	total branch length
	9	Integer	iaxis	[-]	axis number to which this tip belongs
37	1	Real(sp)	ovrtime	[d]	time the tip has been growing without branching; a negative number defines that there are no established branching points
	2	Integer	nestbl	[-]	number of established branching points
38	1-nestbl	Real(sp)	timest	[d]	root age at which branches will develop from that branch

```
Time:
   0.0000000

Number of seeds
      1

ID, X and Y coordinates of the seeds (one per line)
      1 0.00E+00 0.00E+00

Root DM          , shoot DM      , leafarea:
8.18457641E-03   3.15000005E-02   1.5000000

Average soil strength and solute concentration experienced by root system:
   3.8331202268600464       0.0000000000000000

Total # of axes:
      1

Total # of branches, including axis(es):
      1

Total # of segment records:
      1

segID#      x    y    z     prev or  br#  length   surface  mass  segmat
origination time
1  0.000E+00   0.000E+00  0.000E+00  0  1     1 8.00E-01 2.68E-01 0.00E+00 1
0.0000E+00

Total # of growing branch tips:
      1

tipID#     xg    yg    zg    sg.bhd.tp. ordbr#  tot.br.lgth. axs#
overlength  # of estblished points
time of establishing (-->)
      1  0.000E+00  0.000E+00 -8.000E-011     1     1  8.0000E-01    1
-1.0  0
```

Figure 5 Example of a RootSys file.

3.2.3.2 Generating a root system with RootTyp

limites.txt, noeuds.txt, param.txt, sol.txt

If RootTyp is used for generating the root system, the user must define specifically that RootTyp will be used (lrrs=true) in control.in to generate a new architecture. When RootTyp is used, specific input files must be given:

sol.txt: The soil strength is defined in z direction as a factorial value. High values mean high soil strength. Root growth can be influenced by the soil strength.

param.txt: Root growth is parameterized for several species (e.g. zea maize, loliumperenne ...). Thereareseveral param.txt available.

noeud.txt

limites.txt: this imposes soil domain limitations for the root growth. Take care to adequately impose the limiting values a bit smaller than the soil domain. Of course, the root age should be in agreement with the box size (too old root could not grow inside of a too small container).

The age of the generated root is given in param.txt. This time will be taken to be the initial time of the simulation for R-SWMS as well.

Take care that R-SWMS does not support root mortality in RootTyp: then the transfer from RootTyp to R-SWMS format is not good and the model does not converge. To avoid that, impose in param.txt that the time before necrosis is larger than the simulation duration.

3.2.3.3 Dynamic root growth

Dynamic root growth needs to be defined in **root.in**, with an adequate root growth time step in **control.in**, and a **RootSys** file, which contains only a single initial segment.

root.in

Line 5 indicates thenumber of times when axis emergence occurs. Starting with line 7, a list of axis emergence times, associated numbers of emerging axes, and initial growth angles of emerging axes is given. For each time, number and angle triplet a separate line is used. The axis emergence times must be consistent with the root growth times that are specified in the control.in file.

In line 11, the weighting factor for the preferential growth angle with the horizontal plane (geotropism) is given for each axis group.

The next block specifies the preferential growth angle with the horizontal plane for each axis group as a function of temperature. If geotropism is included, the preferential growth angle points vertically downwards and needsthus to be set to -90°. In line 13 the number of temperature/angle – points is given. A list with the temperature and angle values follows starting in line 15. If there is more than one pair of temperature/angle – points for one single root order, the list is expanded in the same line (temperature1 angle1 temperature2 angle2). Usually, the temperature is not taken into account and is thus always set to the optimum value of 20° C. An angle of -90° means vertically downwards.

The next block defines the same parameters, however, not for the axis groups, but for the main laterals, which are all roots with orders larger than 1.

In line 28, the maximum branching order is defined. If Doussan root water uptake is modelled, the branching order should not exceed 3 because the root hydraulic conductivities in BCroot.in are defined for three root orders.

In the following, the maximum elongation rates for each root order as a function of age is defined. The first input in line 31 is the number of age/elongation rate – pairs for each root order. The second input, starting with line number 33,is then a list of age points and associated elongation rates where a separate line is used for each root order. If one root order has different elongation rates at different times, the list is expanded in the same line (age1 elongation rate 1 age 2 elongation rate 2 …).

The value in line 37 is the specific weight of root dry mass (default value 0.0679 g cm^{-3}).

The next block describes the root radius as a function of penetrometer resistance (around the growing root tip) and has the same structure as the block for the root elongation rate. In general, the root radius should become larger when the soil becomes harder. The root radius is a function of the mass per unit length (MPL) and the specific weight of root dry mass (spwgt). In the root.in file the root radius is specified implicitly via the mass per unit length which is calculated as follows:

$$MPL = r_i^2 * spwgt * \pi$$

MPL = mass per length [M L^{-1}]

r_i= root radius

spwgt = specific weight (M L^{-3}) → 0.0679 g cm^{-3}

In line 41 the number of penetrometer resistance/mass per unit length – pairs for each root order is given. The list of penetrometer resistance and associated mass per unit length is then written in a separate line for each root order starting in line number 43.

In line 49, it can be chosen if the root growth model by Somma et al. (1997) or the more mechanistic approach for root growth in structured soil shall be used (false initiates Somma root growth). Adifference between the two methods is only noticeable when soil strength is highly variable e.g. presence of biopores. If l_conduc is set to true, a value for the conductance in the macropore needs to be defined in line 51. A value between 1e4 and 2e5 kPa-1should be chosen.

Line 53 defines the sensitivity of penetrometer resistance on root growth for each root order. Line 56 gives the maximum random heading angle deviation per 1 cmroot growth for each root order (default value = 30°).The next three input blocks describe the influence of temperature and solutes on root growth. If these factors shall be considered, further input files (temp.in, chem.in) are needed and the consideration of the factors also needs to be affirmed in control.in. Line number 59 defines the maximum, optimum and minimum temperatures; line 62 determines whether solute deficiency or toxicity effects are included or not (t for true and f for false). In line 64, the minimum, maximum and the optimum range of the solute concentration values are given.

Line 68 defines the maximum branch length for each root order. It should be set to a value that is longer than the length that is reached within one simulation period because the routine that stops a branch from growing is currently not working. In line 70, the distance or between two branches is given in centimeters. Line 72 specifies the branching angle which is the angle between a 2nd and 3rd order lateral and its parent. In line 74 the branching delay time at the growing tip is given. The branching delay time is the age of the 2nd and 3rd order laterals when they first start branching.

In the last block secondary radial growth can be defined, however, it should be noted that the root radius must not exceed a single voxel size! In line 80, a factor for radial root growth is defined for each root order. If the factor is equal to 1, no radial root growth occurs.

For the plants wheat, maize and barley root.in files are provided on svn in the path rswms/root_soil_inputfiles.

line	Record size	type	variable	Unit	description
1-4	-	Char	hed	[-]	comment lines / heading
5	1	int	naxemg	[-]	Number of axis emergence times
6	-	Char	hed	[-]	comment lines / heading
7-8	2	Real, int	tnewax, nnewax, inaxs	[d],[-],[↓°]	Time of axis emergence, Number of newly emerging axes, minimum Initial angle from the horizontal of newly emerging axes
9-10	-	Char	hed	[-]	comment lines / heading
11	naxemg	int	geoaxs	[-]	Weighting factor for preferential growth angle
12	-	Char	hed	[-]	comment lines / heading
13	naxemg	int	nangax	[-]	Number of temp/angle point pairs
14	-	Char	hed	[-]	comment lines / heading
15-16	Naxemg, nangax	real	Tempax, angax	[°C], [°]	Temperature and associated pref growth angle
17-19	-	Char	hed	[-]	comment lines / heading
20	1		geolat	[-]	Weighting factor for preferential growth angle for main laterals
21	-	Char	hed	[-]	comment lines / heading
22	1	int	nanglt	[-]	Number of temp/angle point pairs for main laterals
23	-	Char	hed	[-]	comment lines / heading
24	nanglt	real	Templt, anglat	[°C], [°]	Temp and associated pref growth angle for main laterals
25-27	-	Char	hed	[-]	comment lines / heading
28	1	int	norder	[-]	Maximum branching order

29-30	-	Char	hed	[-]	comment lines / heading
31	norder	int	nVch	[-]	Number of elongation rate/age – pairs
32	-	Char	hed	[-]	comment lines / heading
33-35	Norder, nVch	real	ageVch, Vch	[d], [cm/d]	Age – elongation rate pairs
36-37	-	Char	hed	[-]	comment lines / heading
38	1	real	SpWgt	[g/cm^3]	Specific weight of root dry mass
39-41	-	Char	hed	[-]	comment lines / heading
42	norder	int	nMPLch	[-]	Number of Penetration resistance and mass per unit length pairs for each root order
43	-	Char	hed	[-]	comment lines / heading
44-46	norder, nMPLch	real	sMPLch, MPLch	[Pa], [g/cm]	Penetration resistance and mass per unit length
47-48	-	Char	hed	[-]	comment lines / heading
49	-	Char	l_conduc	[-]	Do you want to use the conductivity based soil strength approach (ssgcom2)?
50	-	Char	hed	[-]	comment lines / heading
51	-	real	condMP	[kPa-1]	Conductance in the macropore
52	-	Char	hed	[-]	comment lines / heading
53	norder	real	strsen	[-]	Sensitivity to penetration resistance (for each root order)
54-55	-	Char	hed	[-]	comment lines / heading

56	norder	real	rdmang	[°]	Random angle deviation per 1 cmroot growth (for each root order)
57-58	-	Char	hed	[-]	comment lines / heading
59	3	real	Tempermin, top, tempermax	[°C]	Min, opt and max temperature values
60-61	-	Char	hed	[-]	comment lines / heading
62	1	Char	ltoxi	[-]	Do you want to include solute deficiency/toxicity effects?
63	-	Char	hed	[-]	comment lines / heading
64	4	real	Cmin, coptmi, coptma, cmax	[micromole/cm³]	Minimum, optimum range and maximum solute concentration
65-67	-	Char	hed	[-]	comment lines / heading
68	norder	real	brlmax	[cm]	Maximum branching length (per order)
69	-	Char	hed	[-]	comment lines / heading
70	norder-1	real	brspac	[cm]	Inter-branch distance (per root order larger than 1)
71	-	Char	hed	[-]	comment lines / heading
72	norder-1	real	brnang	[°]	Branching angle (per root order larger than 1)
73	-	Char	hed	[-]	comment lines / heading
74	norder-1	real	dtbrch	[d]	Branching delay time at growing tip (per root order larger than 1)
75-77	-	Char	hed	[-]	comment lines / heading
78	1	Char	l_secrad	[-]	Do you want to include secondary radial growth?
79	-	Char	hed	[-]	comment lines / heading
80	norder	real	f_rad	[-]	Factor for radial growth (per root order)

```
***** ROOT PARAMETERS *****

* Axis emergence
          Number of axes emergence times:
          2
          List of (time; number of emerging axes)-points [T];[--]:
          20 4  24 4

* Weighting factor for pref. growth angle with horiz. plane for each axis group:
         5.  5.

* Range of randomness in pref. gr. angle with horiz. pl. for each ax. gr. [DEG]:
         20. 20.

* Pref. growth angle with horiz. plane for each axis group as f(temperature):
          Number of (temp.;angle)-points:
          1 1
          Lists of (temp.;angle)-points [DEG];[DEG], increasing temp.:
         20. -90
         20. -60.

* Weighting factor for pref. growth angle with horiz. plane for main laterals:
          10

* Pref. growth angle with horiz. plane for main laterals as f(temperature):
          Number of (temp.;angle)-points:
          1
          List of (temp.;angle)-points [DEG];[DEG], increasing temp.:
          20.  -35.

* Maximum branching order:
  3

* Unimpeded elongation rate, v = f(branch age) for each branching order
          Number of (age;v)-points:
          1 1 1
          Lists of (age;v)-points [T];[L/T]:
0  0.7      10      1.5
0  0.35      10     0.37
0  0.2      2.5     0.05

* Mass per unit length, MPL = f(soil strength) for each branching order
          Number of (soil strength;MPL)-points:
          2 2 2
          Lists of (soil strength;MPL)-points [P];[M/L]:
0  .00029     1.5 .0012
0  .00008     1.5 .00029
0  .00002     1.5 .00008

* Minimum, optimum, maximum temperature values [DEG]:
  0        20       35

* Include solute deficiency/toxicity effects (yes=t; no=f):
f

* Minimum, optimum range (minimum and maximum), maximum conc. [microMole/cm3]:
0.000  0.1  1.5  2.0

* Somma growth or conductance approach? (l_conduc), false=Somma, true=cond approach
t

* conductance in the macropore (kPa-1):
200

* Heading angle sensitivity to soil strength at reference strength gradient:
  1.  1.5  1.3

* Maximum random heading angle deviation per root growth time step [DEG]:
  30.  30.  30.

* Maximum branch length [L]:
  40. 30. 15.

* Branch spacing [T]!!!   [L]:
  .6  .4

* Branching angle [DEG]:
90  90

* Branching delay time at growing tip [T]:
  1.  .5

* Secondary radial growth (l_secrad): yes=1, no=2
  2
  Factor for radial growth for each order (f_rad, factor=1 means no secondary
growth)
1.6  1  1
```

Figure 6 ***Example of a root.in file.***

rootBox.in

There's the possibility to simulate root growth according to the growing rules developed by Leitner et al. 2010. Some parameters (e.g. gravitropism), however, were still implemented in the same way as in root.in and we also developed some new rules. Differences as compared to Somma root growth are shown in the following table:

line	Record size	type	variable	Unit	description
	1	int	norder	[-]	Maximum branching order
			Lmax (norder)	[cm]	Maximum root length
			diffnum	[-]	How many groups of 1st order roots shall have different lengths to lmax?
		Real	Maxlast (diffnum)	[-]	Maximum length of these axial roots
		Int	Numlast (diffnum)	[-]	Number of last axial roots emerging that have a length, which is smaller than the maximum length specified above
		Real	r	[cm d-1]	Initial elongation rate (decreases with time following a negative exponential function, the rate of decrease is computed with the maximum root length)
		Int	sg	[-]	Sensitivity to gravitropism
		Real	sigma	[°]	Standard deviation of the random angle
		Real	radius	[cm]	Root radius
		Real	lb	[cm]	Basal root length
		Real	ln	[cm]	Inter-branch distance
		real	theta	[°]	Branching angle

Rootbox to RootSys

Matlab subroutines (RB_2_RSWMS.m) exists that can convert the root system architecture from RootBox format to RSWMS readable RootSys file.

3.2.4. Boundary conditions

3.2.4.1. Soil boundary conditions

Bc.in

Bc.in defines the soil upper and bottom boundary conditions.

The **top boundary conditions** can be of two types: water flux or pressure head. The different options are described in the following table:

Line	**Record**	**Type**	**Variable**	**Unit**	**Description**
1-4		Char			Comment lines/Headings
5	1	Integer	qfun	[-]	Qfun describes a water flux as the top boundary condition: Qfun=0: no water flux qfun = 1: irrigation on total top-surface in an homogenous way qfun = 2: irrigation on a partial area(xqmin and xqmax have to be defined) qfun = 3: alternating irrigation on two partial areas (xqmin1, xqmin2, xqmax1, and xqmax2 have to be defined) qfun = 4: radial irrigation (only on material #1, cylindrical domain must be defined in mesh in)
5	1	Boolean	ltop	[-]	Flag argument that indicates R-SWMS whether the water flux is applied at the top or at the bottom
6		Char			Comment lines/Headings
7	2	Integer	Xqmin / Xqmin1	[-]	If qfun=2 (xqmin): node that limits the left boundary for irrigated area If qfun=3 (xqmin1): node that limits left boundary for irrigated area
7	3	Integer	Xqmax / Xqmax1	[-]	If qfun=2 (xqmax): node that limits the right boundary for irrigated area If qfun=3 (xqmax1): node that limits

Commenté [H5]: This has been incorporated to simulate rhizoslides experiments in which irrigation occurs at the bottom of the samples

					the right boundary for irrigated area
7	4	Integer	Xqmin2	[-]	If qfun=3: node that limits the left boundary for irrigated area
7	5	Integer	Xqmax2	[-]	If qfun=3: node that limits the right boundary for irrigated area
8	1	Char			Comment lines/Headings
9	1	Integer	nQbcCh	[-]	Number of 'time ; water flux' points
10	1	Char			Comment lines/Headings
11	1	Real	tQbcCh	[T]	Time values for which water flux is specified
11	2	Real	Qbcrec	$[L T^{-1}]$	Water flux at the corresponding time
12-14		Char			Comment lines/Headings
15	1	Integer	nIrrig	[-]	Number of irrigators
16	1	Char			Comment lines/Headings
17	1	Integer	xIrrig	[-]	Node in x direction where irrigator is located
17	2	Integer	yIrrig	[-]	Node in y direction where irrigator is located
17	3	Integer	zIrrig	[-]	Node in z direction where irrigator is located
18	1	Char			Comment lines/Headings
19	1	Integer	nIBcCh	[-]	Number of 'time ; water flux' points for each irrigator
20	1	Char			Comment lines/Headings
21	1	Real	tIbcCh	[T]	Time values for which water flux is given
21	2	Real	Irbc	$[L^3 T^{-1}]$	Water flow for each irrigator
22-24	1	Char			Comment lines/Headings
25	1	Integer	hfun	[-]	Hfun describes a pressure head as boundary condition: Hfun=0: no pressure head Hfun=1: pressure head for the total surface Hfun=2: pressure head for a partial surface
25	1	Boolean	ltop	[-]	Flag argument that indicates R-SWMS whether the pressure head is applied at the top or at the bottom

26	1	Char			Comment lines/Headings
27	2	Integer	xhmin	[-]	If hfun>1: node that limits the left boundary for the area to which a pressure head is applied
27	3	Integer	xhmax	[-]	If hfun>1: node that limits the right boundary for the area to which a pressure head is applied
28	1	Char			Comment lines/Headings
29	1	Integer	nhBcCh	[-]	Number of 'time ; pressure head' points
30	1	Char			Comment lines/Headings
31	1	Real	thbcCh	[T]	Time values for which pressure head is given
31	2	Real	hbc	[L]	Pressure head at the corresponding time
32-34	1	Char			Comment lines/Headings
35	1	Boolean	lFrdr	[-]	Flag argument that indicates R-SWMS whether a free drainage condition is applied at the bottom of the soil profile or not
36-38	1	Char			Comment lines/Headings
39					Boundary condition for solute transport
40-41	1	Char			Comment lines/Headings
42	1	Integer	nCBnd1	[-]	Number of 'time ; concentration' points for 1st time dependent solute transport BC
	2	Integer	nCBnd2	[-]	Number of 'time ; concentration' points for 2nd time dependent solute transport BC
43	1	Char			Comment lines/Headings
44	1	Real	tCBnd1	[T]	Time values for which concentration is given
	2	Real	CBnd1	[M L^{-3}]	Concentration values
45	1	Char			Comment lines/Headings
46	1	Real	tCBnd2	[T]	Time values for which concentration is given
	2	Real	CBnd2	[M L^{-3}]	Concentration values
47-48	1	Char			Comment lines/Headings
49	1	Real	tPulse	[T]	Pulse times

- Note that the flux boundary time series is given in **L/T**. In the code this value is multiplied by each element surface so that the total volume of a precipitation is Dt*soil boundary surface*Q in cm³.
- Note also that the first BC should correspond to the initial time.
- Note that the values of the boundary fluxes are interpolated between time. If the last input time of the BC conditions has been passed, the last value of the BC is considered.
- The **bottom boundary condition** can be defined as free drainage. To do so, the lFrdr flag must be set to 1. R-SWMS will apply a free drainage to all nodes corresponding to the last soil layer. If lFrdr is set to 0, the bottom boundary acts like a sealed layer.

```
***** BC INFORMATION *****
* Water flux boundary conditions *********************
  *UNIFORM*
  qfun (0-4)  ltop
    0       t
  if qfun=0,1: ---; if qfun=2: xqmin,xqmax ; if qfun=4:
xqmin1,xqmax1,xqmin2,xqmax2
    ----
  nQbcCh
    0
  tQbcCh  Qbcrec
  ---

  *IRRIGATORS*
   nIrrig(nI)
    2
   nIrrig>0: xIrrig     yIrrig      zIrrig
     0           0            0
     0           .5           0
   nIBcCh(nI)
    4 4
   nIBcCh>0: tIbcCh     Irbc
   0       0            0.0139      8.800 0.0556      8.800 0.0557      0
   0       0            0.0139      8.800 0.0556      8.800 0.0557      0

* Head boundary conditions *****************************
  hfun (0-2)      ltop
    0       t
  if hfun>1: xhmin, xhmax
   ----
  nhBcCh
    0
  nhBcCh>0: thbcCh      hbc
    ----

* Free drainage****************************************
  lFrdr
    t

* Solute Transport - FEM Solution *********************
  assCode
    -1

  nCBnd1    nCBnd2
    0       0
  nCBnd1>0: tCBnd1      CBnd1
    ----
  nCBnd2>0: tCBnd2      CBnd2
    ----

* tPulse
    .0
```

Figure 7 Example of bc.in

Climate.in

If the flag lClimate is set to TRUE in control.in, the file Climate.in provides the top soil boundary conditions. If additionally stresfun=6 in CondRoot.in, the file changes. The bottom boundary conditions remain as set in bc.in.

Line	Record	Type	Variable	Unit	Description
1		Comment			
2		Integer	N	-	nclimaticdata
3		Comment			
4		Header			

5	1	Real	dummy		line number
5	2	Real	time_climate	d	time for climate input
	3		Precip	cm	precipitation
5	4	Real	E_pot	cm^3 d^{-1}	Potential evaporation
5	5	Real	T_pot	cm^3 d^{-1}	Potential transpiration
5	6	Real	PPFD	?	if stresfun = 6 (CondRoot.in): photosynthetic photon flux density
5	7	Real	Temperature	°C	if stresfun = 6: air temperature
5	8	Real	VPD	cm	if stresfun = 6: vapour pressure deficit

```
N
5

#           Days        T_act[cm3/day]    P_act [cm]
1.000000    124.000000  0.000000    -0.300000
2.000000    124.990000  0.000000    -0.300000
3.000000    125.000000  0.000000    -0.200000
4.000000    125.990000  0.000000    -0.200000
5.000000    126.000000  22.500000   -0.200000
```

Figure 8 Climate.in for stresfun = 1-5

```
N
5

#     Days        T_act[cm3/day]    P_act [cm]  PPFD  Temperature VPD
1     124.0000    0.000       -0.300      0.0000      15.400      0.574
2     124.9900    0.000 -     0.300       11.000      17.800      0.470
3     125.0000    0.000 -     0.200       233.000     20.000      0.328
4     125.9900    0.000 -     0.200       578.000     18.000      0.233
5     126.0000    22.500      -0.200      1275.000    23.000      1.612
```

Figure 9 Climate.in for stresfun = 6

3.2.4.2. Root boundary conditions

BCroot.in

If the root water uptake process is simulated but without assimilate allocation, the BCroot.in file is read so as to know what the potential transpiration time series is.

In BCroot.in, the user can choose between different type of input boundary function at the root collar:

- BC = 1: free format. The user gives ‘measured’ values at certain times. The values between are then linearly interpolated.
- BC = 2: the collar BC is kept constant (only one time is then needed).
- BC =3: a sinusoidal function is generated which is 0 at t = 1, 2, 3 d and maximum at t =0.5, 1.5, …
- BC = 4: a ‘jump’ function is used, which is maximum at t=0.5, 1.5 … (corresponding to 12h00) and zero between t=0.25d and 0.75 d.

Then, the user must give the number of input nBC (if BC=1 or 2) or the number of time which should be generated (if BC=3 or 4). If BC =1, the user must then provide a list of nBC triplets including the time, the type of boundary condition and the value of the boundary condition. In all other cases, the user must only give one triplet. The triplet is composed of:

The time at which a value is given or the initial time for that BC (if the last user defined BC time is smaller than the final time, this last BC is kept constant to the end).

The type of boundary condition. BCtype= 1 for water potential and BCtype=2 for water flux

The value for the BC if BC =1. If BC =2, the constant value must be given. In case BC =3 and BC =4, the daily transpiration must be imposed (note that BCtype should be equal to 2, flux type BC).

Figure 10 Example of bcroot.in

Commenté [MJ6]: Add the file

The grey line in Figure 1 shows the simulated collar flux time series resulting from the example BCroot.in here before, where BC =3 (sinusoid) and BCvalue= 5 cm³/d (daily transpiration). If BC would be BC =4 and BCvalue= 5 cm³/d, we would get the blue line. Note that for both cases the integration of functions over time for one day are 5 cm³/d (see subplot below), but note that the maximum is much higher for the blue line (BC =4). Units for transpiration are L³/T.

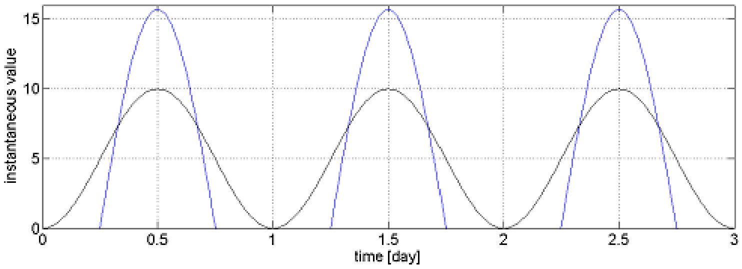


Figure 11 Sinusoidal (green) and 'jump' (blue) function for collar BC.

Line	Record	Type	Variable	Unit	Description
1		Comment			
2		Comment			
3		Header			
4	1	Integer	funBC		Function for the root collar BC: 1- Free format 2- Constant value 3- Sine function 4- Jump function
5		Header			
6	1	Integer	nBCr		Number of different of BC
7		Header			
8	1	Float	tBCr	[T]	Time for boundary condition
8	2	Integer	typeBCr		Type of boundary condition 1 – head 2 – flow
8	3	Float	BCroot	(typeBCr 1) [L] (typeBCr 2) [L3/T]	Value for the boundary condition For BC 1 – constant value For BC 2 - constant value For BC 3 & 4 - daily transpiration

CondRoot.in

This file specifies the root hydraulic characteristics for the Doussan root water uptake model, i.e. the radial conductivity and axial conductance of the root. These parameters can be defined in function of the root age. An example is given in Figure x.

The radial root conductivity L_r is given in T^{-1}. First the number of points for each type of root (3 types by default, i.e. primary, secondary and tertiary) must be given. In Figure x, 2 points are given per each type of root and it is written as '2 2 2'. For constant properties (no change with age), you can type '1 1 1'. Then, the user can enter n t-value pairs (age - Lr) for each root type. Each line corresponds to a root type and in each line the user must write as many t-value pairs as specified above. The same can be done for the xylem conductance K_x [L^3T^{-1}].

The block solute conductivities defines root properties which are needed to calculate pesticide uptake. Here we first set root permeability pairs, for root order and age. In nPass the advective solute uptake is defined, for root order and age. nPass corresponds to the fraction ε [-] (defined in theory part).

Pesticides entering the root can be immobilized by either linear sorption or freundlich sorption, which is defined in the next lines, sorp(1) and sorp(2) correspond to the sorption parameters $K_{D,R}$ [L^3/M] and n [-]. Further root water content and bulk density can be set according to root information. θ_R and $\rho_R$influence solute transport and sorption in the root system.

The three next entries define the way how multiple roots located in one soil voxel are averaged (usual settings are 2, 2).

The third block defines stress parameters. Five stressfunctionsexist:

- Stresfun=1: stress occurs when a threshold value for the water potential (hlim) is reached at the root collar, then the flux BC is modified to a PH CBC with PH= hlim. This corresponds to a isohydric behavior.
- Stresfun =2: when the PH at the root collar is higher (in absolute value) than a threshold value hlim1, the flux is multiplied by the reduction function:

  $$reduc = \frac{h - h\lim_2}{h\lim_1 - h\lim_2}$$

- Stresfun =3: the flux at the root collar is multiplied by the following reductionfunction (Tuzet et al., 2003):

  $$reduc = \frac{1 + \exp\left[s_f h_f\right]}{1 + \exp\left[s_f\left(h_f - h\right)\right]}$$

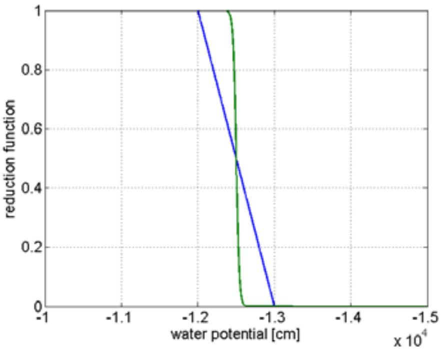
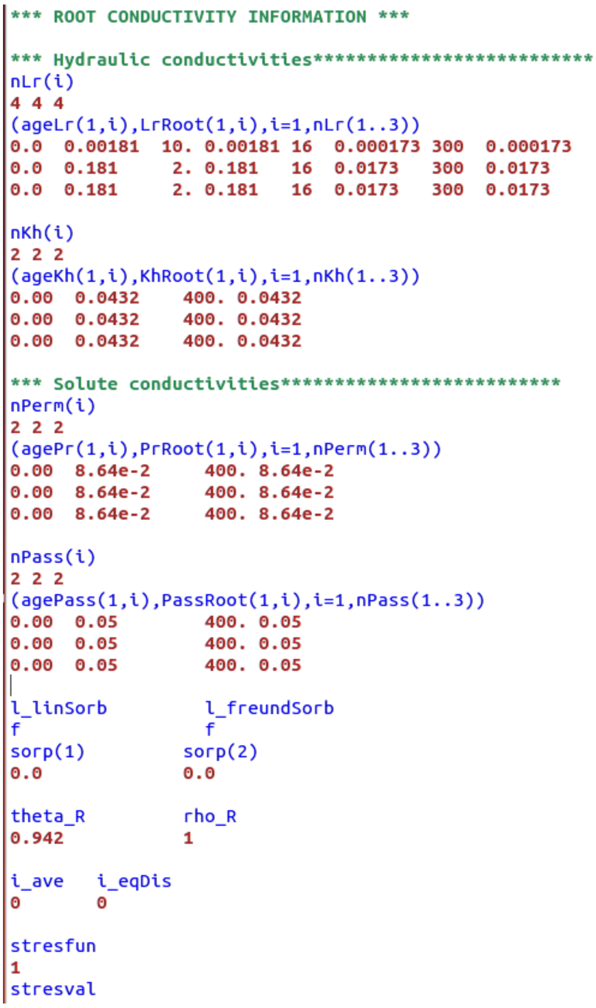


Figure 12 Reduction functions with stresfun=2 (blue) and hlim1=-12000 cm and lim2=-13000cm ; and with stresfun=3 (green) and sf= 0.05 and hf=-12 500 cm.

- Stresfun = 4: Stomatal closure is based on hormone concentration within the leaves (i.e. at the collar). The function is based onTARDIEU, ZHANG, and GOWING (1993). The hormones are produced in the root segments as a function of the local root pressure head and modelled as particles with a certain mass. Afterwards they are transported via advection towards the collar, where they are intermediately stored in a buffer. The hormone concentration of this buffer affects the stomatal aperture. The detailedprocessisdescribed in Huber et al. (2014).
- Stresfun = 5: Same as Stresfun=4, without the transport. The particle mass is immediately effective in stomatal closure.

The last block defines rhizosphere or aquaporin influences on local conductivities. For more details, seeCouvreur et al. (2014).



```
*** ROOT CONDUCTIVITY INFORMATION ***

*** Hydraulic conductivities***************************
nLr(i)
4 4 4
(ageLr(1,i),LrRoot(1,i),i=1,nLr(1..3))
0.0  0.00181  10. 0.00181 16  0.000173 300  0.000173
0.0  0.181     2. 0.181   16  0.0173   300  0.0173
0.0  0.181     2. 0.181   16  0.0173   300  0.0173

nKh(i)
2 2 2
(ageKh(1,i),KhRoot(1,i),i=1,nKh(1..3))
0.00  0.0432    400. 0.0432
0.00  0.0432    400. 0.0432
0.00  0.0432    400. 0.0432

*** Solute conductivities***************************
nPerm(i)
2 2 2
(agePr(1,i),PrRoot(1,i),i=1,nPerm(1..3))
0.00  8.64e-2     400. 8.64e-2
0.00  8.64e-2     400. 8.64e-2
0.00  8.64e-2     400. 8.64e-2

nPass(i)
2 2 2
(agePass(1,i),PassRoot(1,i),i=1,nPass(1..3))
0.00  0.05        400. 0.05
0.00  0.05        400. 0.05
0.00  0.05        400. 0.05

l_linSorb          l_freundSorb
f                  f
sorp(1)          sorp(2)
0.0              0.0

theta_R          rho_R
0.942            1

i_ave    i_eqDis
0        0

stresfun
1
stresval
```

Figure 13 Example of CondRoot.in

Line	Record	Type	Variable	Unit	Description
5	1…3	Int	nLr	-	Number of age/radial conductivity couples for primary, secondary and tertiairy
7…9	1/3/…	Real	ageLr	days	Age of the root

7...9	2/4/…	Real	LrRoot	cm/day/hPa or day^{-1}	Radial conductivity
12		Int	nKh	-	Number of age/axial conductivity couples for primary, secondary and tertiairy
14...16	1/3/…	Real	ageKh	days	Age of the root
14...16	2/4/…	Real	KhRoot	cm^4/hPa/day or cm^3/d	Axial conductance
20	1…3	Int	nPerm	-	Number of age/root radial permeability couples for primary, secondary and tertiairy
22…24	1/3/…	Real	agePr	days	Age of the root
22…24	2/4/…	Real	PrRoot	cm/day	Root permeability
27	1 … 3	Int	nPass	-	Number of age/root advective solute uptake fractions for primary, secondary and tertiairy
29…31	1/3/…	Real	agePass	Days	Age of the root
29…31	2/4/…	Int	PassRoot	-	Root advective uptake fraction
34	1	bool	L_linSorb	-	Chose linear root sorption
34	2	bool	L_freundSorb	-	Chose freundlich root sorption
36	1	Int	Sorp(1)	cm^3/day	Sorption parameter $K_{D,R}$
36	2	Int	Sorp(2)	-	Sorption parameter n
39	1	Int	Theta_R	-	Root water content
39	2	Int	Rho_R	-	Root wet density
43	1	Int	i_ave	-	Use of incorporating one large root
43	2	Int	i_eqDis	-	Incorporate equivalent Distance (eqDis) approach, only if dx=dy (yes=1) or (no=0) -> radius equal (r=1/2d0*sqrt((dx)/noNodes_voxel**(1/3d0))); position changed
46	1	Int	stresfun	-	Stress function type : no stress (0), stress value (1)/ linear function (2) /Tuzet function (3) /Additional signaling (4) /Instantaneous signaling (5)
48	2 …	Real/Int	stresval		values of stress function parameters: stresfun=1: hlim stresfun=2: hlim1 and hlim2 stresfun=3: sf and hf stresfun=4 or 5: a_r=rel. min. stomatal

					aperture, a_1=sensitivity of stomatal conductance to hormone conc., a_2=sens. to PH, PH_crit, a_bool (either 0 or 1), hormone production rate [microMol/T/L],size of buffer (multiple of root volume) --> 7 values in case (4) or (5)
51	1	Int	cavitfun	-	Cavitation consideration : yes (1), no (0)
53	1	Real	cavitb		b parameter of the Weibull function
53	2 ...	Real	cavitc		c parameter of the Weibull function
56	1	Int	i_Gap	-	Gaps considered between soil and roots : yes (1), no (0)
58	1	Real	g1	cm	Gap parameters g1
58	2	Real	g2	cm	Gap parameter g2
61	1	Int	i_AQPc	-	Aquaporins expressions considered during simulation : yes(1) or no (0)
61	2	Int	nAQPc	-	Number of h/value couples for aquaporins expresssion
63	1	Real	AQPh	cm	Pressure head value
63	2	Real	AQPv	-	Ration between new and normal radial conductivity at this pressure head

3.3.4 Soil Input

Soil.in or Soiltab.in

Either**soil.in**or**soiltab.in**canbe used to summarize soil hydraulic properties. By default, when soil.in exists, the code does not search for soiltab.in. Only is soil.in does not exist, soiltab is used. If neither soiltab.in or soil.in exist, the program stops.

When soil.in is used, a variable *nMat* must be given in soil.in, setting the number of different soil types for the simulation, which will increase the number of lines for the retention and conductivity parameters below. For faster program execution, an internal interpolation table for each soil-type's soil hydraulic conductivity and soil water capacity functions is created. The values *hTab1* and *hTab2* denote the two limiting soil water pressure head values for this interpolation table. Independent of how they are specified in '*soil.in*', they are always converted to negative values. During the FEM iterations, only for pressure head values outside the tabulated range will the hydraulic functions be evaluated directly.

The parameters defining van Genuchten (1980)'s soil hydraulic functions θr, θs, a, n, Ks, λ are listed in file 'soil.in'. The Durner (Mod=2) hydraulic model is also implemented in R-SWMS. In this case the nine parameters are θr, θs, a, n, Ks, λ, w2, a2, n2.

```
***** SOIL PARAMETERS *****

nMat   hTab1[L]  hTab2[L]
1      100000.      100000.1  100

NMod (1=VGM, 2=Durn)       thr[L3/L3]  ths[L3/L3]  a[1/L]    n      Ks[L/T]  lambda
             w2     a2[1/L]      n2     ssMAx[P]
1                   .06         .41     0.03           2.5        10.24      0.5
             0      0            0      12
```

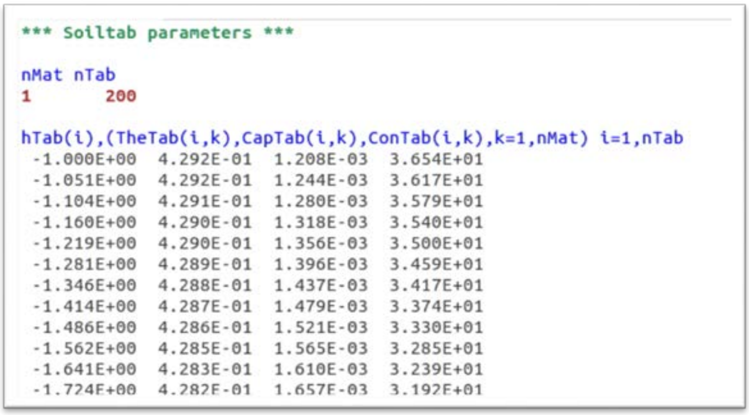
Figure 14 Example of soil.in

Line	Record	Type	Variable	Unit	Description
1		Comment			
2		Comment			
3		Header			
4	1	Integer	nMat		Numberofmaterials
4	2	Float	hTab1	[L]	Minimum abs (pressurehead)
4	3	Float	hTab2	[L]	Maximum abs (pressurehead)
4	4	Float	nTab		Table lengths for h values
5		Comment			
6		Header			
7	1	Integer	NMod		Chose between 1 – van Genuchten 2 - Durn
	2	Float	thr	[L3/L3]	Residual water content
	3	Float	ths	[L3/L3]	Saturated water content
	4	Float	a	[1/L]	Alpha parameter in van Genuchtenequ.
	5	Float	n	[-]	Exponent in van Genuchtenequ.
	6	Float	Ks	[L/T]	Saturated hydraulic conductivity
	7	Float	lambda		Exponent in van Genuchten-Mualem conductivity function
	8	Float	w2		Weight for dual porosity model
	9	Float	a2	[1/L]	Alpha parameter in van Genuchten for the dual porosity
	10	Float	n2	[-]	Exponent in van Genuchten for the dual porosity
	11	Float	rhob	[L]	Bulk density

If nMat is greater than 1, more lines have to be added after line 7.

When a tabulated soil input file is chosen (if no mathematical function exists for your soil properties), which is not based on closed-form functions (Mualem-van Genuchten or Dürner), the file soiltab.in must exist. The first column of the tabulated soil must contain the pressure head values and the other columns the water content, soil capacity and conductivity values. If more than one soil type exists, soil moisture, hydraulic capacity and conductivity must be given for the same pressure head values. By default the first 4 columns represent soil 1, the three next ones soil 2, etc.

Take care to give enough PH tabulations so that good accuracy is achieved. Avoid using 0 as the first value. To build this Table, use log-steps for pressure head.



```
*** Soiltab parameters ***

nMat nTab
1      200

hTab(i),(TheTab(i,k),CapTab(i,k),ConTab(i,k),k=1,nMat) i=1,nTab
 -1.000E+00  4.292E-01  1.208E-03  3.654E+01
 -1.051E+00  4.292E-01  1.244E-03  3.617E+01
 -1.104E+00  4.291E-01  1.280E-03  3.579E+01
 -1.160E+00  4.290E-01  1.318E-03  3.540E+01
 -1.219E+00  4.290E-01  1.356E-03  3.500E+01
 -1.281E+00  4.289E-01  1.396E-03  3.459E+01
 -1.346E+00  4.288E-01  1.437E-03  3.417E+01
 -1.414E+00  4.287E-01  1.479E-03  3.374E+01
 -1.486E+00  4.286E-01  1.521E-03  3.330E+01
 -1.562E+00  4.285E-01  1.565E-03  3.285E+01
 -1.641E+00  4.283E-01  1.610E-03  3.239E+01
 -1.724E+00  4.282E-01  1.657E-03  3.192E+01
```

Figure15: Example of soiltab.in

Line	**Record**	**Type**	**Variable**	**Unit**	**Description**
4	1	Int	nMat	-	Number of considered materials
4	2	Int	nTab	-	Number of interpolation values
7…	1	Real	hTab	cm	Pressure head
7…	2/5/…	Real	ThetaTab	cm³/cm³	Water content
7…	3/6/…	Real	CapTab	1/cm	Capacity term
7…	4/7/…	Real	ConTab	cm/day	Conductivity
EoF	1	Real	ssMaxTab	g/cm³	Soil bulk density

For typical soils such as sandy soil, loam, silty loam and clay there are sample input fiels provided on svn in the path rswms/root_soil_inputfiles.

3.2.5. Solute Input

Chem.in

Chem.in describes the soil solute transport and root solute uptake parameters. These are needed when you want to simulate solute transport and uptake in R-SWMS. Chem.in will be read by R-SWMS when the variable 'lchem' in 'control.in' file is set to 1.Table x contains shows which parameters and where (line and record) in the chem.in file needs to be written. The parameters in line 6 need to be given for all materials defined in 'soil.in' (or soil_tab.in).

Line	Record	Type	Variable[1]	Unit	Description
1,2		Char			Comment line
3	1	Real	epsi	-	Temporal weighing coefficient. =0.0 for an explicit scheme. =0.5 for a Crank-Nicholson implicit scheme. = l.0 for a fully implicit scheme.
3	2	Real	PeCr	-	Stability criterion - Peclet Courant number (for more details, check SWMS manual section 5.3.6)
4,5		Char			Commentline
6	1	Real	Bulk d. [ChPar(1,M)]	[M L^{-3}]	Bulk density of material M
	2	Real	Diff. [ChPar(2,M)]	[L^2T^{-1}]	Ionic or molecular diffusion coefficient in free water
	3	Real	Alpha_L [ChPar(3,M)]	[L]	Longitudinal dispersivity for material type M
	4	Real	Alpha_T [ChPar(4,M)]	[L]	Transverse dispersivity for material type M
	5	Real	Adsorp. [ChPar(5,M)]	[$M^{-1}L^3$]	Freundlich isotherm coefficient for material type M
	6	Real	SinkL1 [ChPar(6,M)]	[T^{-1}]	First-order rate constant for dissolved phase, material type M
	7	Real	SinkS1 [ChPar(7,M)]	[T^{-1}]	First-order rate constant for dissolved phase, material type M
	8	Real	SinkL0 [ChPar(8,M)]	[$ML^{-3}T^{-1}$]	Zero-order rate constant for dissolved phase, material type M

	9	Real	SinkS0 [ChPar(9,M)]	[T-1]	Zero-order rate constant for solid phase, material type M
7-9		Char			Comment line
10	1	Real	CMm	$[M L^{-3}]$	MichaelisMenten constant
	2	Real	VMax	$[M L^{-2} T^{-1}]$	maximum uptake rate per area of soil-root interface
	3	Real	Xin	[-]	partition coefficient between active and passive uptake
	4	Real	fk	[L/T]	parameter defining the linear component of active uptake
11,12		Char			Comment line
13	1	Integer	nURF	[-]	number of (segment age, reduction factor)-points making up the stepwise-linear input function for level '2' and '3' simulations
14		Char			Comment line
15	1	Real	age	[T]	segment-age values of the stepwise-linear input function
	2	Real	urf	[-]	reduction factor values of the stepwise-linear input function

Commenté [H7]: needs to be revised

[1] In brackets [] there is the name of the variable as it is found in the R-SWMS code.

3.2.6. Plant characteristics

Plant.in

If carbon allocation is set to true in control.in, plant.in is read. This file provides the input parameters for the above ground part of the plant, the shoot.

3.2.7. MORE INPUT FILES

Temp.in

Temp.in contains temperature values for the soil profile and the atmosphere. This information may be used to model the influence of temperature on root growth. At the moment, this is not yet implemented in the root growth module. Temp.in will be read by

R-SWMS when the variable 'Itemp' in 'control.in' file is set to 1.Table x contains shows which parameters and where (line and record) in the temp.in file needs to be written.

Line	Record	Type	Variable	Unit	Description
1-4		Char			Comment lines / Headings
5	1	Integer	Nz_tempS	[-]	Number of soil temperature measurement depths
	2	Integer	Nt_tempS	[-]	Number of soil temperature measurement times
6-7		Char			Comment lines / Headings
8	1	Real	depth	[L]	List of depths at which temperature is given. Surface always have to be 0.
9-10		Char			Comment lines / Headings
11	1	Real	Time_S	[T]	Time at which temperature is given
	2	Real	temtim	[DEG]	Temperatures in soil profile at the depths determined in nz_tempS for each given time
13-15		Char			Comment lines / Headings
16	1	Integer	Nt_tempA	[-]	Number of atmospheric temperature measurement times
	2	Integer	Nt_presA		Number of pressure measurements times
	3	Integer	Nt_presD	[-]	Number of pressure deficit measurement times
17-18		Char			Comment lines / Headings
19	1	Real	Time_TA	[T]	Time at which atmospheric temperature is given
	2	Real	T_atm	[DEG]	Atmospheric temperature
21-22		Char			Comment lines / Headings
23	1	Real	Time_PA	[T]	Time at which atmospheric pressure is given
	2	Real	P_atm	[P]	Atmospheric pressure
25-26		Char			Comment lines / Headings
27	1	Real	Time_PD	[T]	Time at which pressure deficit is given
	2	Real	P_diff	[P]	Atmospheric pressure deficit

h_profile.in

In *h_profile.in* a profile of initial soil matric potentials can be defined.

Commenté [H8]: (I am guessing this file was intended to be read by ./grid when initial water potential = 3 (from file) is defined in mesh.in - at the moment in the code only outfem.xxx files can be read by ./grid and the file name and path needs to be written in the code before compiling ./grid - I couldn't find in the code where the h_profile.in file can be read)

Line	Record	Type	Variable	Unit	Description
1-3		Char			Comment lines/Headings
4	1	Real	Depth	[L]	Depth in the soil profile (0= surface, any height below soil surface must be in negative)
4	2	Real	Soil matric potential	[hPa]	Soil matric potential at the corresponding height

Feddes.in

The *Feddes.in* input file is read when lFeddes in control.in is set to 1. This file contains de parameters for 2 types of stress reduction functions, a compensatory RWU function, and an uptake reduction function with segment age.

The first stress reduction function is the Feddes funtion (Feddes et al., 1978):

$$\alpha(h)=\begin{Bmatrix} \frac{h-h3}{h2-h3}, h2>h>h3 \\ 1, h1\gg h\gg h2 \\ \frac{h-h0}{h1-h0}, h0>h>h1 \\ 0, h\ll h3\ or\ h\gg h0 \end{Bmatrix}$$

The second stress reduction function was proposed by van Genuchten (1987). In this case, the function can account for only water stress, or both water and salinity stresses as a multiplicative effect.

$$\alpha(h)=\frac{1}{1+\frac{h}{h_{50}}^{p1}}$$

if only water stress is accounted for, or

$$\alpha(h,p)=\frac{1}{1+\frac{h}{h_{50}}^{p1}}*\frac{1}{1+\frac{p}{p_{50}}^{p2}}$$

if both water and salinity stresses are accounted for. Note that in R-SWMS code 'h' stands for matric potential, whereas 'p' stands for osmotic potential.

The compensatory function that can be used was proposed by Jarvis (1989):

$$S_i = \frac{E_p}{\Delta z_i} R_i \frac{\alpha_i}{\max(\omega, \omega_c)}$$

whereE_p is the potential transpiration rate, Δz_i is the layer thickness, R_i is a root parameter, $\alpha_i$is the stress function, and omega is the ratio between the actual and the potential transpiration.

Line	Record	Type	Variable	Unit	Description
1-3		Char			Comment lines/Headings
4	1	Real	stresfun	[-]	0= no stress function used 1= Feddes function 2= van Genuchten function
5-7		Char			Comment lines/Headings
8	1	Real	h0	[hPa]	Feddes function parameter; pressure close to 0 at which root water uptake stops (alpha = 0) due to oxygen deficiency
	2	Real	h1	[hPa]	Feddes function parameter; pressure above which alpha decreases due to oxygen deficiency
	3	Real	h2	[hPa]	Feddes function parameter; pressure below which alpha decreases due to water stress
	4	Real	h3	[hPa]	Feddes function parameter; pressure at which root water uptake stops (alpha = 0) due to water stress
9-11		Char			Comment lines/Headings
12	1	Real	p50	[hPa]	Van Genuchten function parameter; osmotic pressure at which root water uptake is reduce by half (alpha = 0.5)
	2	Real	h50	[hPa]	Van Genuchten function parameter; matric pressure at which root water uptake is reduce by half (alpha = 0.5)
	3	Real	p1	[-]	Van Genuchten function parameter; shape parameter in water stress reduction function
	4	Real	p2	[-]	Van Genuchten function parameter; shape parameter in salt stress reduction function

13-14		Char			Comment lines/Headings
15	1	Boolean	lJarvis	[-]	Flag to indicate R-SWMS if the Jarvis function for compensatory water uptake is used (1=yes, 2=no)
16-17		Char			Comment lines/Headings
18	1	Real	OmegaC	[-]	Compensation factor (0-1); 0=fully compensated, 1=uncompensated RWU
19-20		Char			Comment lines/Headings
21	1	Boolean	lUrf	[-]	Flag to indicate R-SWMS if an uptake reduction function (URF) is used (1=yes, 2=no) - only if rRLD is not given
22-24		Char			Comment lines/Headings
25	1	Integer	Nex	[-]	Number of elements in x
	2	Integer	Ney	[-]	Number of elements in y
	3	Integer	Nez	[-]	Number of elements in z
	4	Real	dxGrid	[L]	Element length in x
	5	Real	dyGrid	[L]	Element length in y
	6	Real	dzGrid	[L]	Element length in z
26		Char			Comment lines/Headings
27	1	Integer	nPt	[-]	Element number
	2	Real	RLD	[L L^{-1}]	Root length density

couvreur.in

When using the Couvreur et al. (2012) model in R-SWMS, the three macroscopic root water uptake parameters (K_{rs}, K_{comp} and SSF) can either be fed in the input file Couvreur.in, or can be calculated from a hydraulic architecture (RootSys and CondRoot.in files) during the model setup.

dJvL.in

1.3. ParTrace Inputs

If partake shall be used, the partrace flag in Makefile.defs must be set to 1. Partrace input files are found in the folder PartraceInOut. Partrace input information are often node or element based and the size of this information has to be adjusted to the grid size from the R-SWMS inputs.

Note that the **grid definition in ParTrace is different from R-SWMS**!!! It starts at the left, front bottom node, moves then in x, afterwards in y and last in z direction upwards.

Input_PARTRACE.pTraceInpV12

Input_PARTRACE.pTraceInpV12 is the controlling file for ParTrace simulations. There is a GUI to generate this file in the cluster, called partracemenu.

It first states in which folder the additional input files are found. Next the soil grid has to be defined by the number of elements in each direction and the discretization. Further dispersion properties and weights as well as velocities can be defined. The velocity is an input from R-SWMS, thus the mean and variance of the velocity components is set to zero. Next the scaling factor for the time step size (?) and the MaxTime (? should this be the same as in control.in ?) are defined. The diffusion model should be water content based as this is another input that is forwarded from R-SWMS. The porosity and bulk density of the soil should match the soil properties from soil.in. To use the R-SWMS-ParTrace coupling the velocities have to based on the soil nodes! As R-SWMS can only run serial at the moment, the number of processors can only be 1.

Commenté [H9]: I do not know, maybe Natalie can check this

The next three paragraphs should / can remain unchanged… More explanation necessary!

Commenté [H10]: Still some parameters need to be explained

… Next the geometry type and the geometry file needs to be defined. Water content can be defined from file (out.water_content.00001) or not. Then the number of particle types and the particle name need to be given.

The next block contains information related to particle properties. First the sorption type is defined. Next the initial conditions for the particle concentration needs to be defined. The initial conditions can be: Dirac, cube, sphere, ellipse, homogeneous initial concentration or concentration from file. Start position and radius in each direction need to be defined when a non homogeneous starting concentration or injections are modeled. The number of particles to inject (if injection is defined) or initial concentration is specified. Next the particle mass is defined. Then sorption constants need to be given. If initial conditions are set as 'conc from file', the file name needs to be written. This has been implemented to start simulations from previous simulation output, so that the output savcon file can be read. Next the decay order has to fit with the decay given in decay.in file.

Commenté [H11]: types 1 to 9?

Commenté [H12]: kd, n and b?

Commenté [H13]: Needs tobefinished

The last block defines the boundaries of the domain. First the total number of boundaries that will be defined needs to be given to the program. For each boundary, its type (reflection, concentration, injection or periodic), side and element information is necessary. A reflection boundary prevents particles from leaving the soil domain through the boundary. A concentration or injection boundary is defined when solute is applied at that boundary. For boundaries of type concentration, the particle name, the concentration at the flux boundary, the start and end time of the solute inflow need to be added. A periodic boundary is used for continuous domains and enables a particle to leave the soil from one side and enter it through the opposite side. If no boundary is defined for a side, particles are able to leave the soil domain through that side. The starting and last element (node) of the two directions defining the plane of a side need to be given. The number of the last element has to be changed to the actual grid.

Line	Record	Type	Variable	Unit	Description
1		Char			Begin of input file for Partrace
2-3		Char			Folder where input and output files are (PartraceInOut)
4		Char			Restart file
5		Char			Velocity input filename
6		Char			File name for concentrations
7		Char			File name for breakthrough curves
8		Char			File name for moments output
9	1	Real	Time	[T]	Simulation time at start
10	1	Real	Save distance for concentration	[T]	Time interval between concentration outputs (e.g. if this parameter is set to 0.1 and time is measured in days, concentration output will be created every 0.1 days)
11	1	Integer	No elements x	[-]	Number of elements in x direction
12	1	Integer	No elements y	[-]	Number of elements in y direction
13	1	Integer	No elements z	[-]	Number of elements in z direction
14	1	Real	Side length x	[L]	Length of soil domain in x direction
15	1	Real	Side length y	[L]	Length of soil domain y
16	1	Real	Side length z	[L]	Length of soil domain z
17	1	Boolean	Use soil field	[-]	
18	1	Real	Dispersion mean x		

19	1	Real	Dispersion mean y		
20		Char			Filename of partrace input file with soil properties (partrace.soilprop)
21	1	Real	Dispersion weight x		
22	1	Real	Dispersion weight y		
23	1	Real	Dispersion weight z		
24	1	Real	Diffusion coeff		
25	1	Boolean	Use velocity field		Can be set to 1(=no) or 2 (=yes)
26	1	Real	Velocity mean x	[L/T]	
27	1	Real	Velocity mean y	[L/T]	
28	1	Real	Velocity mean z	[L/T]	
29	1	Real	x-velocity variance	[L/T]	
30	1	Real	y-velocity variance	[L/T]	
31	1	Real	z-velocity variance	[L/T]	
32	1	Real	Time step size scaling factor	[-]	1 / (number of time steps a particle should stay within a soil element)
33	1	Real	MaxTime	[T]	
34	1	Integer	Diffusion model	[-]	The diffusion model can be simple (1) or water content based (2)
35	1	Integer	Water content		Read water content from file based on (1) elements, or nodes (2)
36	1	Real	Porosity	[-]	Soil porosity
37	1	Real	Bullk density	[M/L^3]	Bulk soil density
38		Char			Name of the executable (partrace)
39	1	Integer	Velocity based		Velocity can be based on elements (1) or nodes (2)
40	1	Boolean			Distribute data over PEs

					(1=no or 2=yes)
41	1	Integer			Number of processors
42					Blank line
43	1	Integer		[hours]	CPU time
44	1	Integer			Run job in interactive mode (1) or batch (2)
45	1	Char			Job name
46	1	Integer			Auto initialize random generator (1=yes, 2=no)
47	1				Seed for random generator
48					Blank line
49	1	Real			Save distance for moments outputs
50	1	Integer			Save distance for restart file
51	1	Real			First time for restart file
52	1	Real			Save distance for particle positions – these files can be very large!
53					
54					
55	1	Integer			Geometry type (1= regular; 2=xy regular, z not; 3=irregular, 4=rectilinear)
56		Char			Filename of input file for geomtry (Geometry_4.in)
57	1	Boolean			Read water content from file (1=no, 2=yes)
58	1	Integer			Number of particle types
59		Char			Particle name
60		Char			Abbreviation of particle name
61		Char			Sorption file name
62	1	Integer			Sorption type (1=no sorption or conservative, 2=linear sorption, 3=Freundlich isotherm, 4=Langmuir isotherm, 5=linear non-equilibrium, 6=Freundlich non-equilibrium, 7=Langmuir non-equilibrium, 8=Freundlich isotherm with retardation, 9=interaction with Freunlich

					isotherm)
63	1	Boolean			Use sorption field (1=no, 2=yes)
64	1	Real			Mean (weight) of Kd
65	1	Real			Mean (weight) of n
66	1	Integer		[-]	Number of injections or initial concentration
67	1	Integer		[-]	Start model as 1=dirace injection, 2=cube injection, 3=sphere injection, 4=ellipse injection, 5=homogeneous initial concentration, 6=initial concentration from file (savcon file)
68	1			[-]	Start position x (only for injection initial conditions)
69	1			[-]	Start position y (only for injection initial conditions)
70	1			[-]	Start position z (only for injection initial conditions)
71	1			[-]	Start radius x (only for injection initial conditions)
72	1			[-]	Start radius y (only for injection initial conditions)
73	1			[-]	Start radius z (only for injection initial conditions)
74	1			[T]	Time when injection starts
75	1			[T]	Time when injection finishes
76	1	Real / Char		[M/L³] / [-]	If initial condition is initial concentration, here you need to write the initial concentration. If initial condition is restart file, here you write the filename of the concentration restart file
77	1	Real		[-]/[M]	If initial condition is injection, this is the number of particles to be injected / If initial condition is concentration or restart file, this is the mass of one particle – in this case there should be approx. 150-200 particles per voxel; conc. is based on the conc in the

Commenté [AK14]:
The initial concentration is given in M/L³_water
If sorption is activated, the initial concentration is given as if all mass would be solved M_tot/L³_water

					solution, thus m_part_ini = V_soil_tot*theta_ini*conc_ini / n_elm_soil / 175
78	1	Real			Particle c0
79	1	Real	kd		Value of sorption parameter Kd
80	1	Real	n		Value of sorption parameter n
81	1	Real	b		Value of sorption parameter b
82	1	Integer			Decay order (1=no microbial decay, 2=zero order decay, 3=first order decay)
83	1	Boolean			Use decay field (1=no, 2=yes)
84	1	Real			Decay rate
85		Char			Filename of input file for decay (decay.in)
86	1	Integer			Uptake order (1=no uptake, 2=advective or passive uptake, 3=active (Michaelis –Menten) uptake, 4=diffusive uptake coupled to r-swms)
87	1	Boolean			Use uptake field (1=no, 2=yes)
88	1	Real			Uptake rate
89					Blank line
90	1	Integer			Number of breakthrough curves (BTCs)
91	1	Integer			Number of interactions
92	1	Integer			Number of boundaries
93	1	Integer			Boundary type (1=reflection, 2=concentration, 3=injection, 5=periodic)
94	1	Integer			Boundary side (1=bottom, 2=top, 3=front, 4=back, 5=left, 6=right)
95	1	Integer			Starting element for first dimension

96	1	Integer			Last element for first dimension
97	1	Integer			Starting element for second dimension
98	1	Integer			Last element for second dimension
99	1	Integer			Number of element sequences
100	1	Integer			Output type (1=text, 2=HDF5, 3=both)
101	1	Integer			Restart type (1=full, 2=particles only)
102	1	Integer			Convection method
103					Dispersion method
104		Char			End of input file for partrace

decay.in

The coupled R-SWMS-ParTrace version mimics solute uptake as decay! Thus, at the moment there is no 'real' decay rate possible, as the decay is already defined as solute sink.

Element based information! This file has to be adjusted to the current grid size. If decay values are greater than zero, beware of the different grid logics!

Geometry_X.in

Depending on the type of geometry X chosen in Input_PARTRACE.pTraceInpV12, which is currently 4=rectilinear, this file has to be adjusted with **nodal** information. The first line contains the total number of nodes per each direction (x, y, and z). Next, for each direction, the number of all nodes need to be written.

out.veloci.00001, out.water_content.00001

Nodal information about the velocity components and the water content in the soil at the beginning of the simulation.The information need to fit the grid defined in R-SWMS and also the simulation time as defined in RootSys.These files can be generated if only R-SWMS is executed (compilation without ParTrace!) but with the ParTrace output set to true in **control.in**.

partrace.index

This file contains **element-wise** index information. Indexing can be used to define layers or blocks of the soil domain with different soil properties.The amount of elements and entries have to be adjusted to the currently used grid size. If an homogeneous soil profile is modeled, then a value '1' is assigned to all elements.

partrace.soilprop

This file contains the soil properties with respect to the diffusion and dispersion parameterization from the dispersion tensor (Eq. **Fehler! Verweisquelle konnte nicht gefunden werden.**) used in the particle tracking algorithm (α_L, α_T and difussion=D_m)(more information needed!). Furthermore, porosity and bulk density are specified and should match the values already given in **soil.in**.If different soil layers or blocks have been specified in the partrace.index file, here different parameterizations can be listed.

partrace.sorption

This file defines multiple sorption properties of the soil domain. First the number of elements into each direction has to be given, then the sorption model can be chosen.

In the next line the number of parameters per set has to be specified, eg. for linear equilibrium sorption only one parameter is needed (Kd) whereas for freundlich sorption 2 parameters are needed (Kd, n). Next line the number of parameter sets is specified, which is the number of different soil sorption property sets. Parameters are given in the following lines. If a property distribution in the soil domain is given (more than one

```
#partrace adsorption file
# version number
1
# elements x y z
5 5 12
# sorption model number:
#  1 = conservative
#  2 = linear
#  3 = freundlich
#  4 = langmuir
#  5 = linear none equilibrium
#  6 = freundlich none equilibrium
#  7 = langmuir none equilibrium
#  8 = freundlich with retardation
#  9 = freundlich with interaction
3
# number of parameters per set
1
# number of parameter sets ( 0 means no indexing )
2
# kd  n
 0.5  0.8
# distribution
# distribution included (1) or from file (2)
2
# distribution file name
PartraceInOut/partrace.index
```

Take care that when the sorption input file is used, you still need to indicate the sorption type in the InputPartrace-file. If you chose sorption type one (conservative) the input file is not read. If you chose a sorption type which is inconsistent with the sorption you define in the partrace.sorption file .... Check which one will be used.

When sorption field is set to true (2), the file sorption.index must be provided. The structure is usually equal to partrace.index. Thus, partrace.index can simply be copied and renamed.

4. Output files

This chapter explains what information can be obtained from the output files created by R-SWMS. The units of length, time, mass and concentration in the output files are the ones defined in the file **control.in**. The following table gives an overview on the different output files and what information they provide.

Table 7 Overview of output files from R-SWMS

Output file type	**Output file**	**Maininformation**
Simulation information	simul_sum.out	Main inputs and boundary conditions used in the simulation
Root System and root water uptake	outRoo1	Root segmentproperties
	Log1	Transpiration
	snkElm	Water sink elements
Waterandsolutetransport	balance.out	Waterandsolutebalances
	remove.out	Water and solute transport across boundaries
	outfem.x	Soil variables at each node
	veloci	Water velocity at the soil nodes
VTK	outRoo01	Root properties
	veloci	Soilfeatures

Moreover, you will also find information on the outputs created by Partrace. These are resumed in the following table.

Table 8 Overview of output files generated by Partrace.

Output file type	Output file	Maininformation
Solutetransport	savcon.p00000	Solute concentration, water velocity and water content
VTK	velociC	Soilfeatures

4.1. Simulation information

Simul_sum.out

This output file summarizes the main inputs and boundary conditions imposed for a specific simulation. Very helpful to check the numerous input files which have been associated with a scenario.

4.2. Root system and root water uptake

OutRoo1

The output files outRoo1.x with x being an integer between 1 and the number of outputs specified by the user in 'control.in' summarize the properties of all the root segments at prescribed times:

- col.1: segID: segment ID
- col. 2,3,4: x, y, z: coordinate position of the node from which the segment originated
- col. 5: br#: branch number to which the segment belongs
- col. 6: prev: previoussegment
- col. 7: Lr: radial root conductivity of the segment
- col. 8: Khr: axial root conductance of the segment
- col. 9: Phinter: pressure head at the root-soil interface
- col. 10: PHxylem: pressure head inside the segment xylem
- col. 11: radialRootFlow: radial root flow (positive value indicates flow from the soil into the root)
- col. 12: axialRootFlow: axial root flow (positive value indicates flow towards the collar)
- col. 13: Qi: Radial conductance * surface in diagonal matrix
- col. 14: Qd: Pressure head at soil root interface [L]
- col. 15: Q_bc: Boundary condition flow
- col.16: radius: segmentradius
- col. 17: veloRoot: Velocity of root water
- col.18: segconc: solute concentration in the segment
- col. 19: segsorb: sorbed concentration in the segment

Log1

This file summarizes information on transpiration for each time step:

- col.1: Time
- col. 2: Tpot: potential transpiration
- col. 3: Tact: actualtranspiration
- col. 4: sAvg: average soil strength which the root systems senses - only calculated / updated in case of Somma root growth
- col. 5: soil_lim_nodes.: number of soil limiting nodes in the whole soil profile
- col. 6: H_collar: water potential at the collar

snkElm

The output files snkElm.x with x being an integer between 1 and the number of outputs specified by the user in 'control.in' summarize the sink elements at prescribed times:

- col. 1: Element number
- col. 2: SinkElm: water uptake in each element [V (water) / V (soil)]. Positive value indicates water flowing from soil into the root.

4.3. Water and solute transport

balance.out

This file summarizes absolute and relative mass balance error for water and solute transport. The contained information is:

- col.1: Time
- col. 2: WatVol: total water volume in the soil
- col. 3: WatBalT: total water balance in the soil
- col. 4: WatBalR: relative water balance in soil
- col. 5: ww.: maximum between the change in water volume between time 0 and current time step and the fluxes crossing the soil and root boundary
- col. 6: wCumT: sum of all cumulative fluxes across the soil boundaries including those resulting from sources and sinks in the domain
- col. 7: WatVol-wVolT: Volumetric mass balance between time 0 and current time step
- col. 8: wCumA: sum of absolute values of all fluxes across boundaries including those resulting from sources and sinks in the domain
- col. 9: Deltaw: cummulative difference between the volumetric water content at time 0 and current time step
- col. 10: TotalRootFlow: total rootflow
- col. 11: TotRadialFlow: total radial flow

remove.out

This file shows all the water and solute fluxes across boundaries or taken up by the root system at all time steps. Definitions:

- col. 1.: time
- col. 2: CumCh0: cumulative amount of solute removed by zero-order reactions
- col. 3: CumCh1: cumulative amount of solute removed by first-order reactions
- col. 4: CumChR: cumulative amount of solute removed by the roots
- col. 5: ChemS(1): cumulative solute flux across boundary type 1 (flux or PH boundary nodes)
- col. 6: ChemS(2): cumulative solute flux across boundary 2 (free drainage)
- col. 7: CumRt: cumulative root water uptake
- col. 8: CumQ(1): cumulative water flux through boundary 1 (flux or PH boundary nodes)
- col. 9: CumQ(2): cumulative water flux through boundary 2 (free drainage)

outfem

The output files outfem.x with x being an integer between 1 and the number of outputs specified by the user in 'control.in' summarize the soil variables for each soil node. If the first output time for Fem is equal or lower than the initial time, then outfem.1 is not created.

The first two lines of this file give information on the total volume of water and solute at the corresponding time of the file. Afterwards, information for each individual node is provided in columns:

- col.1: Node #: nodenumber
- col. 2: Mater.#: material number
- col. 3,4,5: x, y, z: node position coordinate
- col. 6: h: water potential
- col. 7: conc.: soluteconcentration
- col. 8: theta: watercontent
- col. 9: wsink: water sink term in Richards equation (node based sink)
- col. 10: csink: solute sink term in Richards equation (bode based sink)

veloci

The output files veloci.x with x being an integer between 1 and the number of outputs specified by the user in 'control.in' summarize the water velocity at the soil nodes:

- col.1: Node#: nodenumber
- col. 2,3,4: x, y, z: coordinate position of the node
- col. 5,6,7: Vx, Vy. Vz: water velocity in x,y and z directions respectively

savcon.p00000x

The output files savcon.p00000x with x being an integer between 1 and the number of outputs specified by the user in 'control.in' summarizes the information obtained from Partrace per each soil element (corresponding to each row):

- col.1: soluteconcentration (solved) [M/L^3_{water}]
- col. 2: solute concentration (sorbed) [M/L^3_{water}]
- col. 3: volume of one soil voxel
- col. 4: water content

5. Visualisation and animation

To visualize outputs, several softwares can be used, and routines have been developed by the developers to plot the R-SWMS output.

5.1. Matlab

Matlab programs are available on request, which can read most of the output and input files of R-SWMS (RootSys, outfem, outroot, etc.) and allows to visualizate the data.

The existing matlab scripts are:

- draw_root.m : function to plot the root system in white, once the RootSys file has been read (see the below figure).

- read_balance.m: function to read the output file *balance.out*.

- read_log1.m: function to read the output file *log1*.

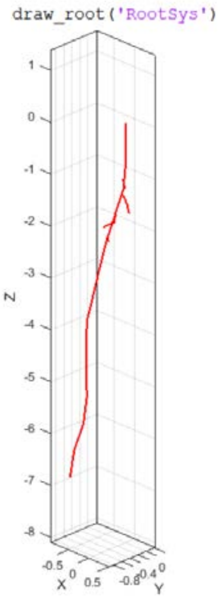
- read_outfem.m: function to read the output file *outfem.xx*.

- read_outroot.m: function to read the output file *outRoo1.xx*.

- read_rootsys.m: function to read the input file *RootSys*.

- read_savcon: function to read the output file *savcon.xxxx*.

- reorganize_savcon: function to reorganise the savcon variable (loaded from savcon file into matlab environment) so that the elements are in the same order as in R-SWMS.

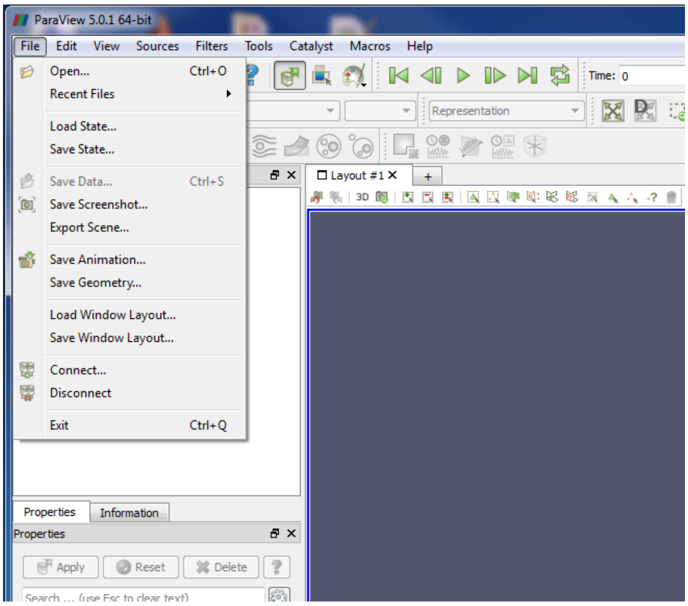


```
draw_root('RootSys')
```

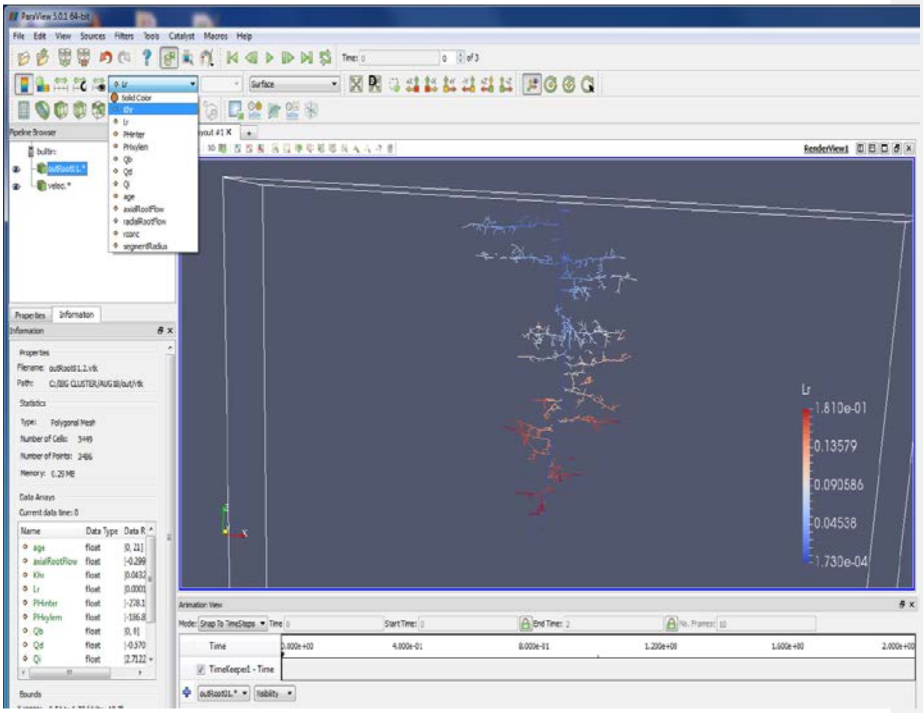
5.2. PARAVIEW

The output files outroot.vtk and veloci.vtk are written in the vtk directory. They can be imported in Paraview for visualization. The outroot.vtk file contains root information such as hydraulic conductivity, age, water potential, radial and axial flow and segment radius. The veloci.vtk contains soil information such as water content, pressure head etc. Following steps describe the visualization of vtk files generated in R-SWMS in Paraview.

Step 1: Click on open in the file menu and select outroot.vtk and veloci.vtk files and click apply



Step 2: The files that are loaded: outRoot01.* and veloci.* can be seen in the left top corner in the pipeline browser of Paraview. The root properties such as age, radial flow etc can be visualized by selecting the appropriate option in the menubar. The information section describes the various variables inside the outRoo01 file.



Properties
Filename: outRoot01.2.vtk
Path: C:/BIG CLUSTER/AUG18/out/vtk

Statistics
Type: Polygonal Mesh
Number of Cells: 3449
Number of Points: 2486
Memory: 0.25 MB

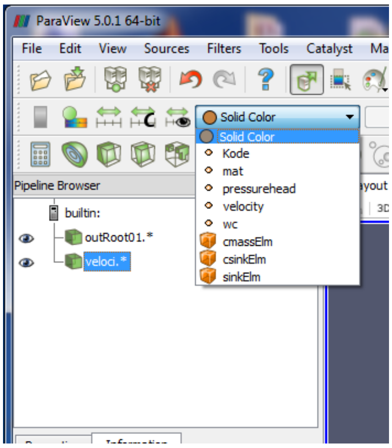
Data Arrays
Current data time: 0

Name	Data Type	Data R
age	float	[0, 21]
axialRootFlow	float	[-0.299
Khr	float	[0.0432
Lr	float	[0.0001
PHinter	float	[-278.1
PHxylem	float	[-186.8
Qb	float	[0, 0]
Qd	float	[-0.570
Qi	float	[2.7122

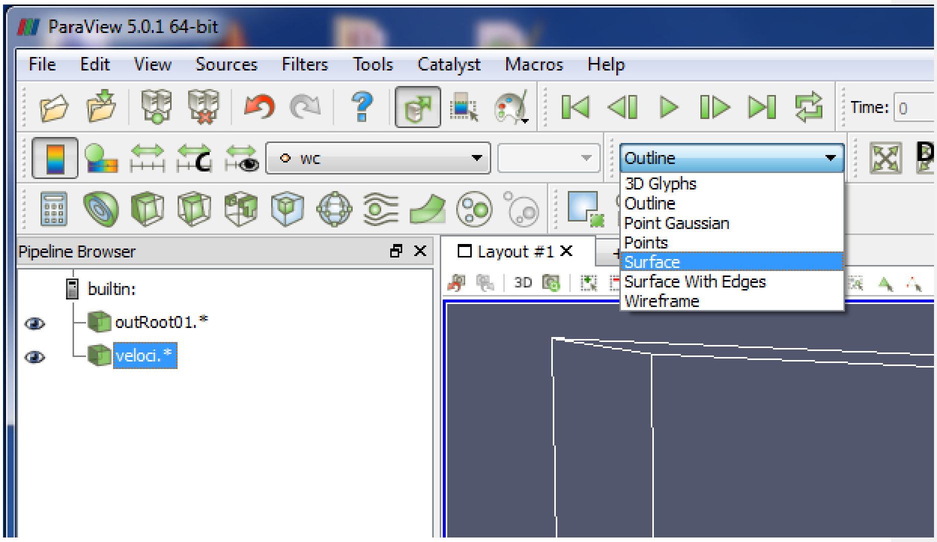
Bounds

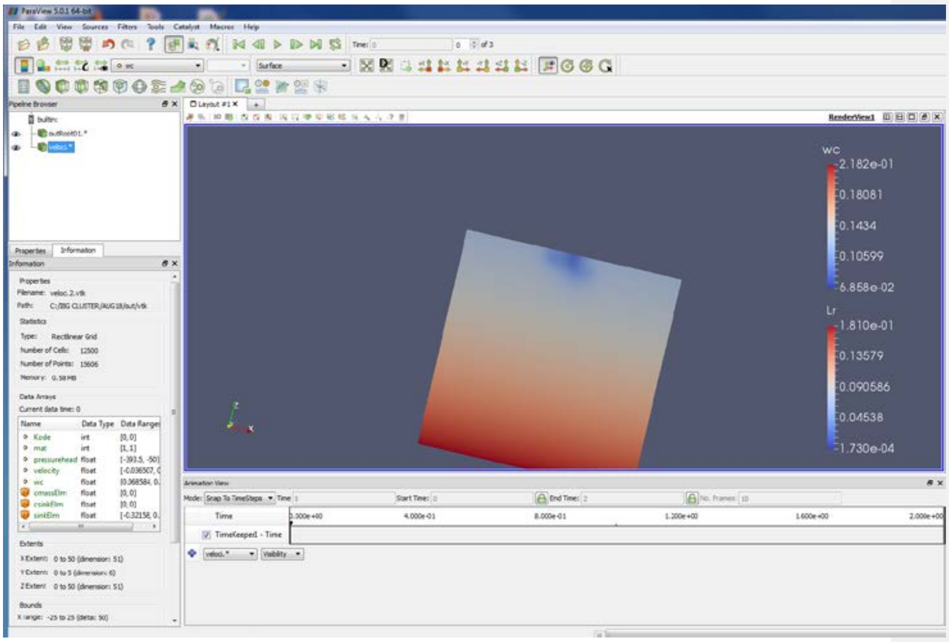
Mode: Snap To TimeSteps Time: 0 Start Time: 0 End Time: 2 No. Frames: 10

Step 3: To view the soil property, click on veloci.* file in the pipeline browser. Now the properties in the menubar changes to soil properties such as pressure head, water content etc.



Step 4: To view a property of soil say water content along with root property, select wc from the menubar and select surface





Information

Properties

Filename: veloc.2.vtk

Path: C:/BIG CLUSTER/AUG18/out/vtk

Statistics

Type: Rectlinear Grid

Number of Cells: 12500

Number of Points: 15606

Memory: 0.58 MB

Data Arrays

Current data time: 0

Name	Data Type	Data Ranges
Kode	int	[0, 0]
mat	int	[1, 1]
pressurehead	float	[-393.5, -50]
velocity	float	[-0.036507, 0
wc	float	[0.368584, 0.
cmassElm	float	[0, 0]
csinkElm	float	[0, 0]
sinkElm	float	[-0.32158, 0.

Extents

X Extent: 0 to 50 (dimension: 51)

Y Extent: 0 to 5 (dimension: 6)

Z Extent: 0 to 50 (dimension: 51)

Bounds

X range: -25 to 25 (delta: 50)

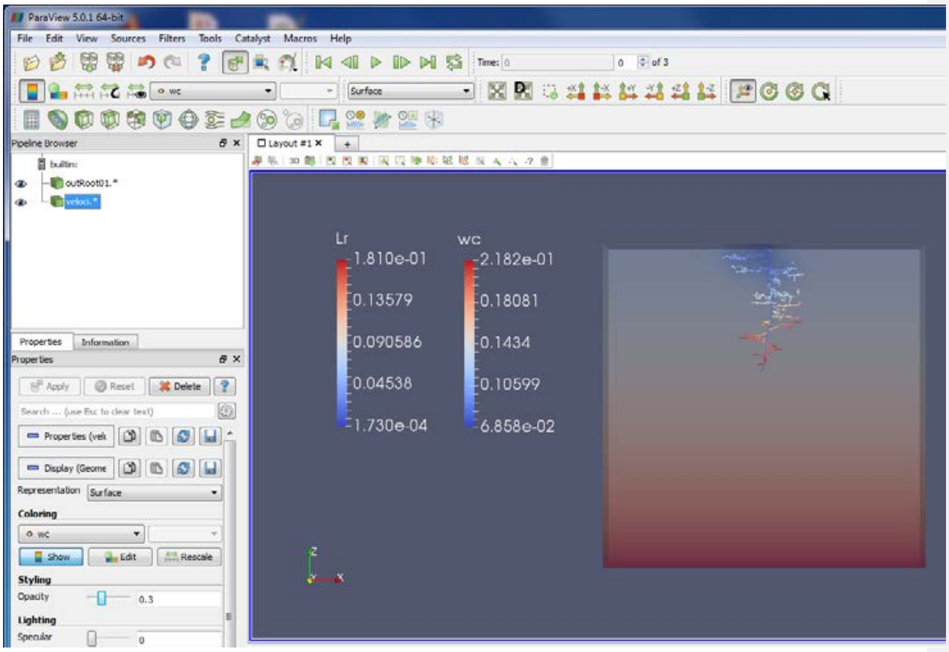
Mode: Snap To TimeSteps Time: 1 Start Time: 0 End Time: 2 No. Frames: 10

Time	0.000e+00	4.000e-01	8.000e-01	1.200e+00	1.600e+00	2.000e+00

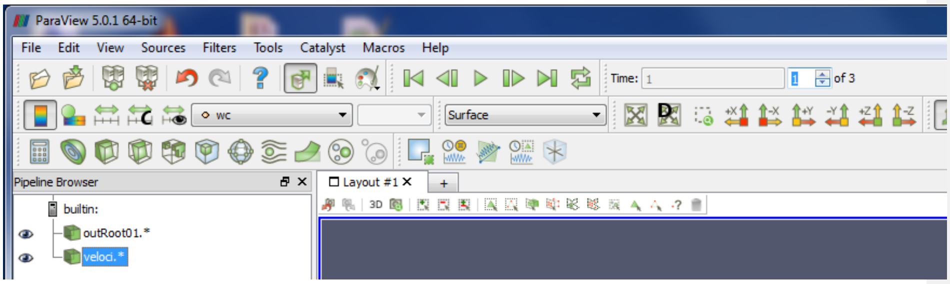
☑ TimeKeeper - Time

veloci.* Visibility

To unmask the hidden root property by the water content, the opacity can be adjusted to suitable level. The opacity adjustment is obtained by clicking properties in the left side of the figure.



Step 5: To view different soil or root property at different times, change the time option from the menu bar in the top panel as shown in the screenshot below.



5.2.1. RSML

Lobet et al. (2014) propose a novel root format for saving root system. Output of R-SWMS can be transformed to RSML thanks to matlab routines.

5.2.2. How to get Vtk files?

If selected in file 'control.in', R-SWMS will create two vtk files which can be read by 3-D visualization programs such as ParaView (later in this manual we will give some examples of relevant information that can be visualized with ParaView).

OutRoo01

As the output file outRoo1, the output files outRoo01.x with x being an integer between 1 and the number of outputs specified by the user in 'control.in' summarize the properties of all the root segments at prescribed times.

veloci

The output files veloci.x with x being an integer between 1 and the number of outputs specified by the user in 'control.in' give information on water velocity, water content, pressure head, solute concentration, water and solute sinks.

velociC

The output files velociC.x with x being an integer between 1 and the number of outputs specified by the user in 'control.in' give information on pressure and osmotic head, water velocity, water content and element solute concentration.

ParRoot.001.0x.vtk

In case particle tracking within the root is chosen (hormornal signaling or pesticide uptake), ParRoot.vtk files are created. They contain information about particle position, particle age, and particle mass.

Installation guide

R-SWMS runs on a Linux environment. It has been tested with Ubuntu 14.04 LTS (available here: http://www.ubuntu-fr.org/telechargement). If you work on Windows, we advise you to create a virtual box: https://www.virtualbox.org/.

1) Install the requested libraries:

   - Open linux terminal
   - Update OS: *sudo apt-get update*
   - Install gcc: *sudo apt-get install gcc*
   - Install gfortran: *sudo apt-get install gfortran*
   - Download/install openmpi: *sudo apt-get install openmpi**
   - Download/installlapack: *sudo apt-get installliblapack-dev*
   - Download/installmumps: *sudo apt-get install *mumps**

2) Download R-SWMS:

   Available on request (see https://sites.uclouvain.be/R-SWMS/index.php/downloads). If you are a developer and have access to the svn repository, export the last version to a local directory using 'export'.

3) Adapt source files for external use:

   In makefile.defs comment with #

   Line 6: # F90C = mpif90

   Line 22: # LDFLAGS = -O0 -fhole-program -fnews-linker-plugin

   Line 28: #LDLIBS = -lstdc++ -L../sparskit/ -lskit -lblas -pthread -lmumps_common -ldmumps -lpord -lmpi –llapack

   Remove comment with #

   Line 8: F90C = gfortran

   Line 31: LDLIBS = -lstdc++ -L../sparskit/ -lskit -lblas -pthread -lmumps_common_seq -ldmumps_seq -lpord -lmpiseq_seq -llapack

   In Main.f90 comment with a !

   Line 23, 36, 37, 144, these lines are already indicated by the comment '!if rswms runs on the cluster'.

4) Compile sparskit:

- *cd ~/sparskit*
- *make clean*
- *make*

5) Compile R-SWMS:

- go to the main directory where the make file is: *cd ..*or*cd ~RSWMS*
- *make clean*
- *make (don't worry about all warning messages appearing)*

6) Run an example by going in the right folder and typing: *./rswms*

Example problems

The following examples are already integrated in the Example folder, which can be found on the svn repository. To run the examples the input folders should be renamed to **in** and copied to the run directory, then R-SWMS can be executed and the output files analyzed.

Example 1 – Infiltration

1. Description

In this example we solve a simple infiltration problem with R-SWMS. The problem is similar to the "INFILTR" example in hydrus 1D, and allow a comparison of the two codes.

The model domain is a 20x20x60 cm box, containing an initially dry sandy soil (h(t=0)=-600 cm). The hydraulic properties of the soil are described by the Van-Genuchten-Mualem model and the hydraulic parameters are given in the input file soil.in. At time t >0, the pressure head at the top of the soil column is set to zero and water infiltrates. The boundary condition at the bottom of the soil column is free drainage. The simulation is carried out for 5 hours and results are printed in one hour time steps. A detailed description of the required input files is given below.

2. Technical description of the relevant input files.

A detailed description of the used input files is given in the following:

control.in

The control.in file is used to define theabsolute error levels, the time intervals in which the simulation is carried out and other general parameters that control the model. In this example we choose 5 output times (every ~ 1hour)

```
***OUTPUT INFORMATION*************************************
nOUT
5

tOut
0.0416  0.0833  0.1249  0.166  0.208
```

In this example we do not consider root water uptake and it is thus necessary to set all parameters that are related to root water uptake to false. The same needs to be done for the parameters which are related to root growth.

```
***ROOT INFORMATION*****************************************

Root Water Uptake model                               | Sink type
lno_RWU          lFeddes     lDoussan   lCouvreur     | lSinkCube
t                   f              f           f           t

Initial architecture type
lno_Archi         lRootSys(lrrs) lRootTyp(lrrt)
f                 t              f

Root growth model
lno_root_growth  lRootTyp_growth lSomma_growthlUpdate_growth
t            f                   f             f
```

soil.in

In this file, all soil hydraulic parameters are defined. In the current example we only have one material (nMat is set to 1); the hydraulic properties are defined by help of the Van Genuchten-Mualem model (NMod set to 1) as shown below.

```
***** SOIL PARAMETERS *****

nMat   hTab1  hTab2 nTab
1      100000.      100000.1  100

NMod   thr  ths  a  n   Ks  lambdaw2a2  n2     rhob
1      .0    .3308   0.0143   2    200.0      0.5         0       0       0      1.2
```

mesh.in

In the mesh.in file the domain geometry, the mesh size and the initial water pressure are defined. In this example, the length of the block sides in x and y direction is 2 cm, in the z direction it is 3 cm.The number of blocks in x - and y - direction is n=10 and in z - direction n=200. The size of the whole soil domain can then be calculated by multiplying the number of blocks with the block side lengths which gets us a domain size of 20x20x600 cm.

It is also necessary to define node 1 which is the node in the top left corner and which in our example has the coordinates -10 -10 0. The initial water potential is set to be constant and equal over depth to -600 cm.

In the following the grid needs to be generated which is done by running ./grid in the src_R-SWMS folder.

```
*** Mesh generator parameters ***

dx     dy    dz     nex    ney   nez    xmin       ymin      zmax
2      2     3      10     10    200    -10        -10         0

ICtype   htop
1        -600

geom
2
param_geom
4

conctr
0.
Initial water potential conditions type (1=cst, 2=eq, 3=from file)
1
ICtype=1 or 2: Initial water pressure head at the top of the soil domain [L]
-600
```

bc.in

In this example, a matric head defines as the top boundary condition. The boundary condition does not change over time and the top matric head is set to be zero. In order to define free drainage as bottom boundary condition, the lFrdr flag needs to be set to true.

```
* Head boundary conditions ******************************
  hfun (0-2)  ltop
    1          t
  if hfun>1: xhmin, xhmax
----------
  nhBcCh
    1
  nhBcCh>0: thbcCh  hbc
0.0

* Free drainage*****************************************
  lFrdr
    t
```

In the below figures, the main outputs are shown. The initial soil water potential was set to -600cm. At time t >0, the pressure head at the top of the soil column is set to zero and water infiltrates. The volumetric water content thus equally increases with time.

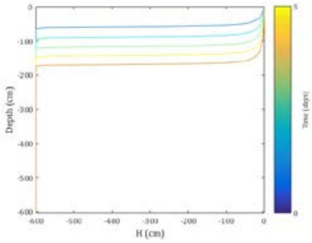


Figure 16: soil water potential over the simulation period

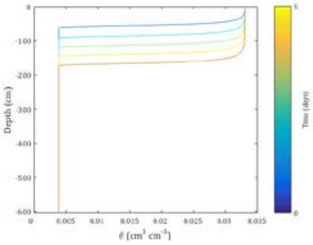


Figure 17: volumetric water content over the simulation period

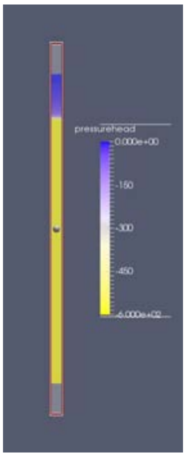


Figure 18: soil water potential, day 01

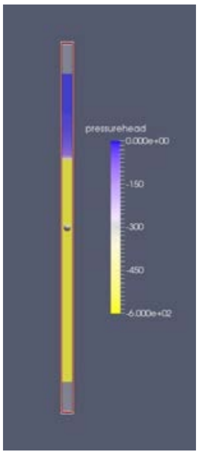


Figure 19: soil water potential, day 05

Example 2: Simple root water uptake with a complex architecture

Description

Here we would like to simulate the root water uptake of a root system whose complete hydraulic architecture is known, i.e. the hydraulic properties of the segments are uniform and completely defined in CondRoot.in while the architecture can be found in RootSys.in file. This root system is submitted to a constant potential transpiration (BCroot.in) and is located in a loamy soil (soil.in).

Input/output files

Input folder: in_RWU1

- Total runtime (control.in): 3 days (discretization: 0.1 days, start at day 5)
- Initial Root architecture type: from Root System (control.in)
- No root growth (control.in)
- Soil domain (mesh.in): 125 cm^3 (20*10*20 blocks of 0.25*0.5*0.5 cm side length)
- Soil boundary conditions (bc.in): no flow at top, bottom and sides
- Soil initial conditions (mesh.in): pressure head = -300 cm at the top; equipotential distribution
- Root boundary conditions (BCroot.in): constant flux
- Uniform radial and axial root conductivities (CondRoot.in)
- Isohydric stomatal behavior (CondRoot.in)

Mainoutputs:

- Root fluxes, pressures, uptakes (OutRoo.XX)
- Soil fluxes and pressures (Outfem.XX → same structure as nodes.in)
- Actual transpiration rate, pressure at root collar (log1)

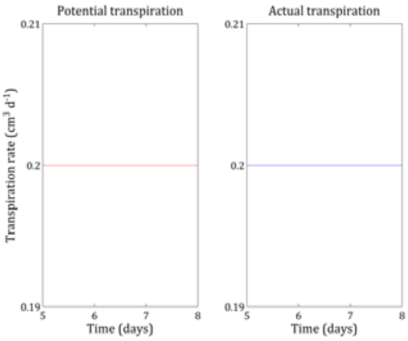


Figure 20: Potential and actual transpiration rates over the simulation period.

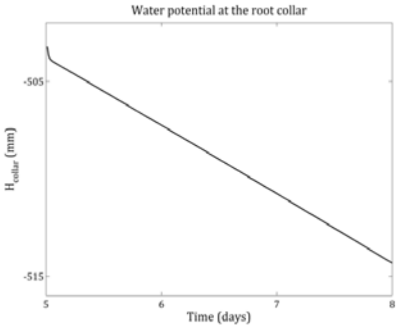


Figure 21: Water potential at the root collar over the simulation period.

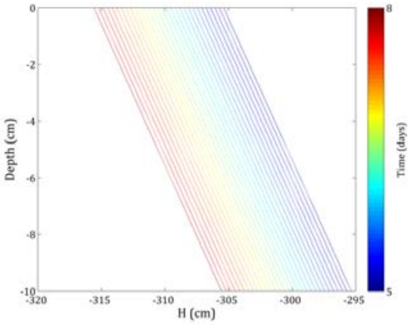


Figure 22: Soil water potential over the simulation period.

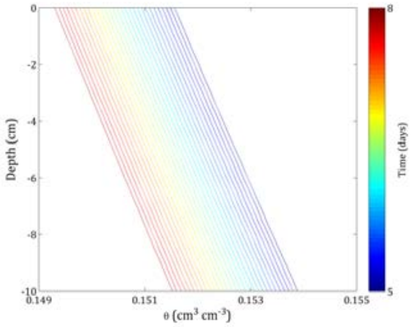


Figure 23: Volumetric soil water content over the simulation period.

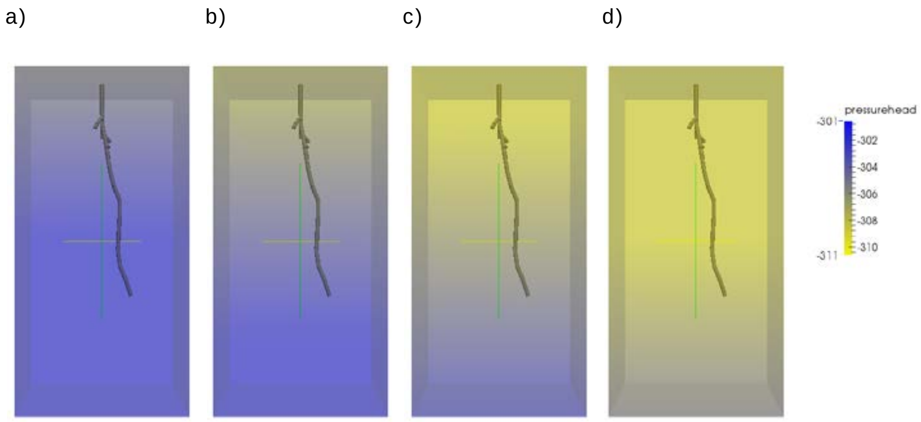


Figure 24: 3D distribution of pressure head for different simulation times: a) 5 days, b) 6 days, c) 7 days, d) 8 days). Visualization with Paraview.

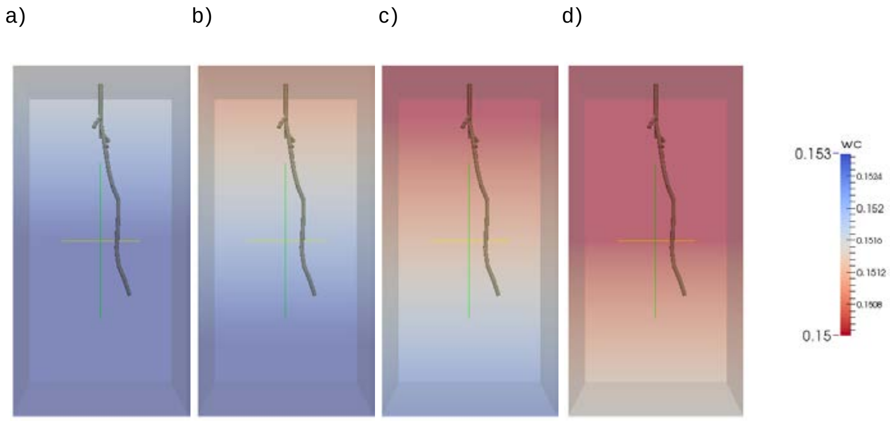


Figure 25: 3D distribution of volumetric soil water content for different simulation times: a) 5 days, b) 6 days, c) 7 days, d) 8 days. Visualization with Paraview.

Example 3: Complex root water uptake for 3 days

Scenario description

The third example simulates the behavior of a more complex root system whose architecture is made of 523 segments in a large loamy soil domain. The hydraulic properties of the root segment are now dependent of the root order and segment age. The transpiration changes versus time: we apply a sinusoidal potential transpiration whose peak is reached at midday and equals zero at noon. The soil domain is considered continuous, i.e. any root reaching a x or y boundary cross the frontier and reappears at the other side.

Input/output

Input folder: in_RWU2

- Total runtime (control.in): 3 days (discretization: 0.1 days, starting at day 20)
- Continuous domain for the side boundaries (control.in)
- Initial root system (control.in)
- No root growth (control.in)
- Soil domain (mesh.in): 2500 cm^3 (10*5*50 blocks of 1*1*1 cm side length)
- Soil initial conditions (mesh.in): pressure head = -300 cm at the top; equipotential distribution
- Soil boundary conditions (bc.in): no flow at top and bottom
- Root boundary conditions (BCroot.in): sinusoidal flux bc
- Non-uniform radial and axial root conductivities (CondRoot.in)
- Isohydric stomatal beahvior (CondRoot.in)

Mainoutputs:

- Root fluxes, pressures, uptakes (OutRoo.XX)
- Soil fluxes and pressures (Outfem.XX → same structure as nodes.in)
- Actual transpiration rate, pressure at root collar (log1)

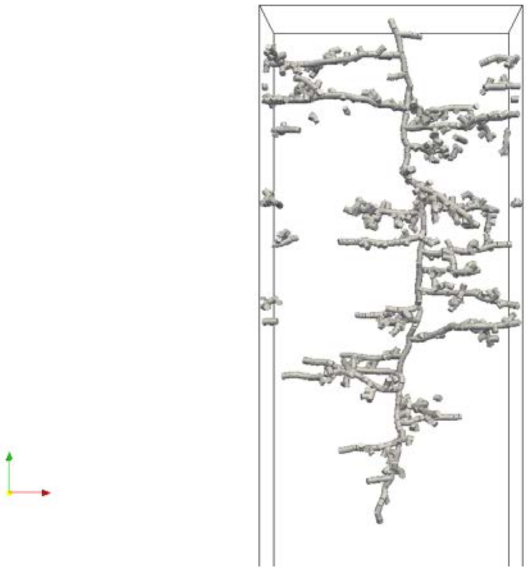


Figure 26: Static root system in a continuous soil domain, visualization with Paraview

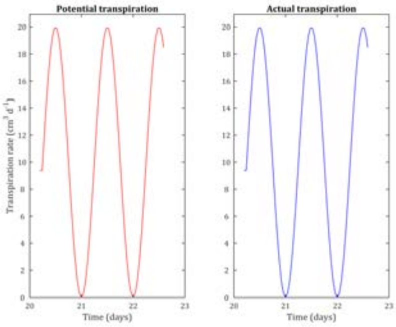


Figure 27: Potential and actual transpiration sinoidal flux during the simulation period

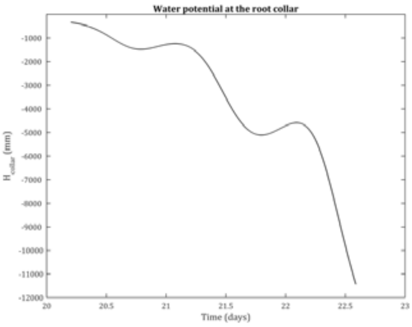


Figure 28: Water potential ant the root collar

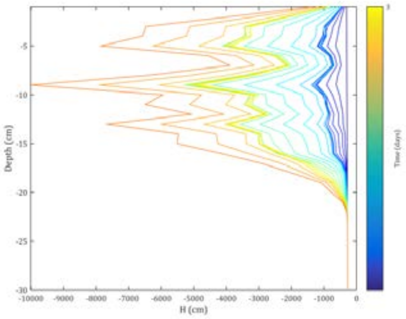


Figure 29: Water potential in the soil domain during the imsulation time

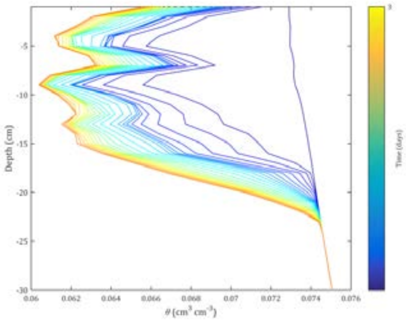
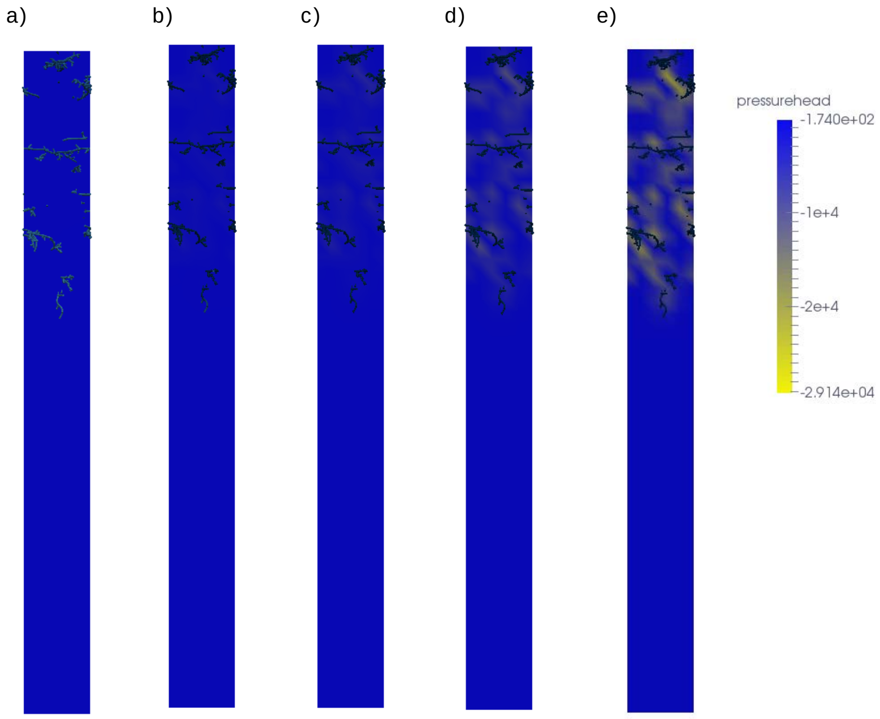


Figure 30: Water content in the soil domain during the simulation time



T=20.21 T=20.9 T=21.4 T=21.9 T=22.4

Figure 31: Pressure head in the soil column

Example 4: Complex root water uptake and an architecture generated with RootTyp

Scenario description

The root system is now generated before the simulation. The roottyp input files are first read (param.txt, soil.txt, limites.txt) to create a fifty days-old root system submitted to a three days drought stress episode. The potential transpiration is once again sinusoidal.

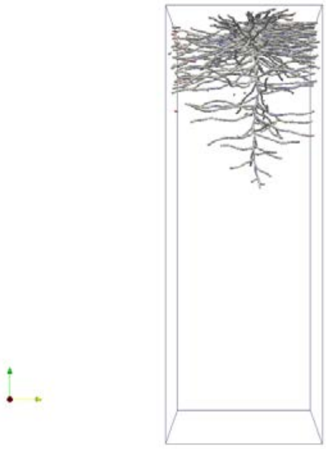


Figure 32: RootTyp root system in continous soil domain

Input/output

Input folder: in_RWU2

- Total runtime (control.in): 3 days (discretization: 0.1 days, start at day 60)
- Root generated by Roottyp (control.in) with the parameters located in param.txt, sol.txt and limites.txt
- No root growth (control.in)
- Continuous domain for the side boundaries (control.in)
- Soil domain (mesh.in): 255 150 cm^3 (discretization: 30*30*84 blocks of 1.5*1.5*1.5 cm side length)
- Soil initial conditions (mesh.in): pressure head = -300 cm at the top; equipotential distribution
- Soil boundary conditions (bc.in): no flow at top and bottom
- Root boundary conditions (BCroot.in): sinusoidal flux bc
- Non-uniform radial and axial root conductivities (CondRoot.in)
- Isohydric stomatal behavior (CondRoot.in)

Mainoutputs:

- RootSys generated (named according to its age)
- Root fluxes, pressures, uptakes (OutRoo.XX)
- Soil fluxes and pressures (Outfem.XX → same structure as nodes.in)
- Actual transpiration rate, pressure at root collar (log1)

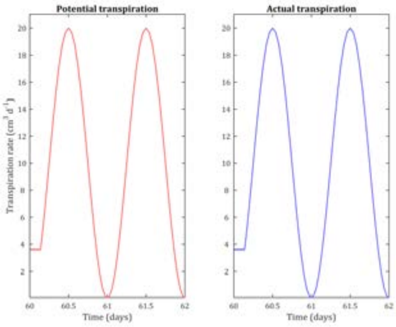


Figure 33: Potential and actual transpiration in the simulation time, with a daily sinoidal flux

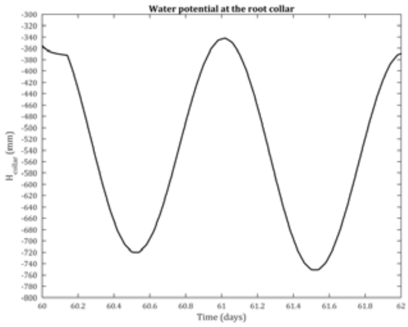


Figure 34: Water potential at the root collar

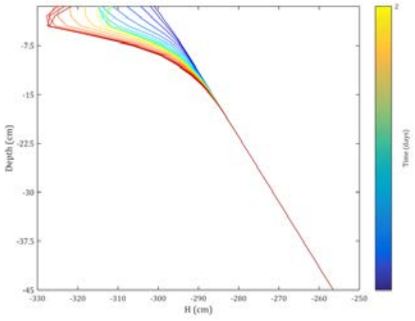


Figure 35: Water potential versus depth

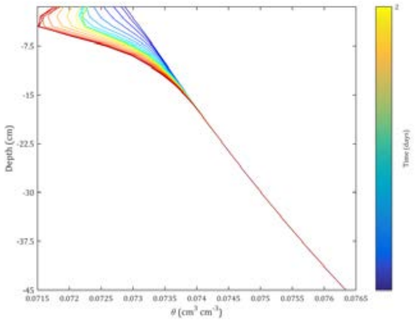


Figure 36: Soil water content versus depth

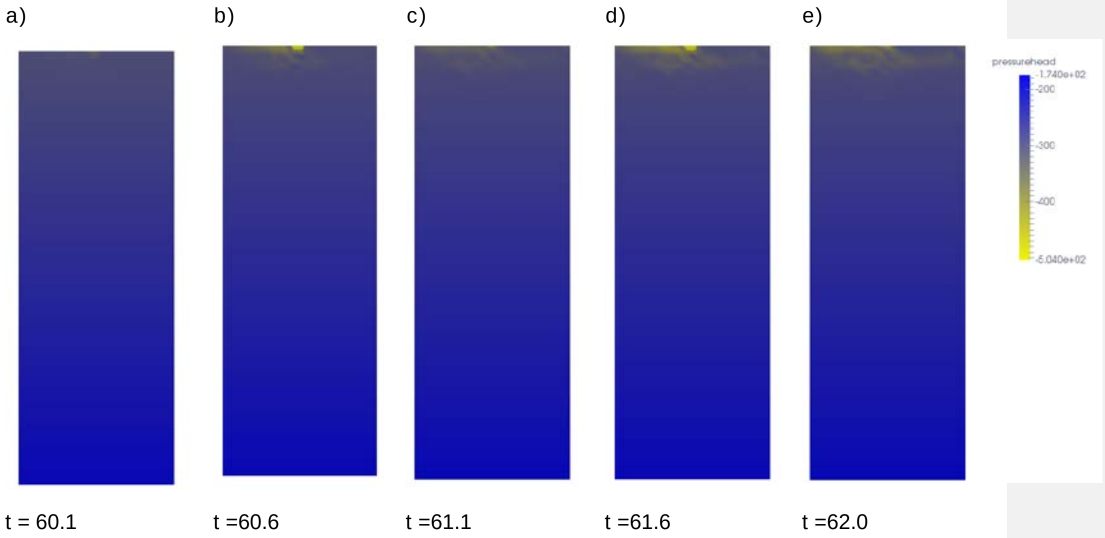


Figure 37 a – e: Cross section ofthe domain showing the pressure head at five points in time.

Example 5: Simulation of a growing root that takes up water

Scenario description

This example simulates a root system which starts to grow from a seed.While growing, the root simultaneously takes up water (simulated by the Doussan root water uptake function).

Input/output

In the file control.in the root growth time step (dt = 0.05 days) and the total runtime of the simulation (total runtime = 5 days, discretization= 0.2 days, start = day 0)are defined. Doussan water uptake and the use of the RootSys file for the definition of the initial root architecture need to be set to true.

```
***** APPLICATION CONTROL PARAMETERS *****

***UNITS***********************************************
[L]   [T]      [M]  Concentration    [P]:
'cm'  'days'  'g'  'microMole/cm3'   'MPa'

***BASIC INFORMATION***
itMax itMaxRoot
 30      20

RelEps factorRelEps
f       1000

errH    errWC   errRootPH  errSink
0.01     .02      0.01      0.0001

dt     dtMin    dtMax   FacInc  FacDec  dtRoot
0.01  .000001   0.05     1.1    0.9     0.05

lRetry | last_out
f             10

***OUTPUT INFORMATION*************************************
nOUT
26

tOut
0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0 2.0 2.2 2.4 2.6 2.8 3.0 3.2 3.4 3.6 3.8
```

```
4.0 4.2 4.4 4.6 4.8 5.0

lvtk   lOutpartrace   profOK
t           f         f

dtprof
0.2

***ROOT INFORMATION*****************************************
Root Water Uptake model                              | Sink type
lno_RWU         lFeddes      lDoussan   lCouvreur    | lSinkCube
f         f            t           f               t

Initial architecture type
lno_Archi         lRootSys(lrrs) lRootTyp(lrrt)
f                 t              f

Root growth model
lno_root_growth       lRootTyp_growth   lSomma_growth   lUpdate_growth
f                         f                   t              f
```

In the root.in file, the maximum root elongation is specified. Taking into account that the roots should not grow more than 1 voxel length within one time step it is set to 1.2 cm/day.

In the mesh.in file the domain geometry, the mesh size and the initial water pressure are defined. In this example, the length of the block sides in x direction is 0.25 cm, in y and z direction it is 0.5 cm.The number of blocks in x - and z - direction is n=20 and in y - direction n=10. The size of the whole soil domain is thus 5x5x10=250 cm³.

It is also necessary to define node 1 which is the node in the top left corner and which in our example has the coordinates -2.5 -2.5 0. The initial water potential is set to -100 cm at the top of the soil domain. The pressure head distribution over the depth is equipotential.

```
*** Mesh generator parameters ***

dx     dy     dz     nex    ney    nez    xmin   ymin   zmax
0.25   0.5    0.5     20    10     20     -2.5   -2.5     0

ICtype htop
2      -100

geom
```

```
2
param_geom
4

conctr
0.
```

In the bc.in file, the soil boundary conditions are defined. In our example there is no flow at the top and at the sides of the soil domain; the bottom boundary is set to free drainage (lFrdr flag set to true)

```
***** BC INFORMATION *****
* Water flux boundary conditions *********************
  *UNIFORM*
  qfun (0-4)
    0
  if qfun=0,1: ---; if qfun=2: xqmin,xqmax ; if qfun=4: xqmin1,xqmax1,xqmin2,xqmax2
----
nQbcCh
    0
tQbcCh  Qbcrec
    0       0

  *IRRIGATORS*
nIrrig(nI)
0
nIrrig>0: xIrrig    yIrrig zIrrig
    ----
   nIBcCh
    0
nIBcCh>0: tIbcCh    Ibc
    ----

* Head boundary conditions *****************************
  hfun (0-2)  ltop
    0        t
  if hfun>1: xhmin, xhmax
   ----
  nhBcCh
    1
  nhBcCh>0: thbcCh  hbc
    0.0 0.0

* Free drainage****************************************
  lFrdr
    t
```

In the BCRoot.in file, the root boundary conditions are defined. In this example root fluxes for two points in time are defined. The root fluxes are thought to decrease with time.

```
**********UPPER BOUNDARY CONDITIONS FOR ROOT*************

funBC
1
nBCr
```

```
2
tBCr typeBCr BCroot
0      2       0.5
1      2       0.2
```

Main outputs:

- Root systems (named via their age, e.g. 0.200) → same structure as RootSys
- Soil water content and soil water potential (Outfem.XX → same structure as nodes.in)
- Potential and actual transpiration rate, pressure head at the root collar (log.1)
- Root fluxes and pressure heads in the root (OutRoo.XX)

The below figures show some of the main outputs. The soil water potential ranged between -90 and -100 cm over all depths during the whole simulation period. The volumetric soil water content varied between 0.11 and 0.14 cm^3/cm^3. The actual transpiration rate reflects the input values from the file BCRoot.in, where the root fluxes were defined to decrease with time.

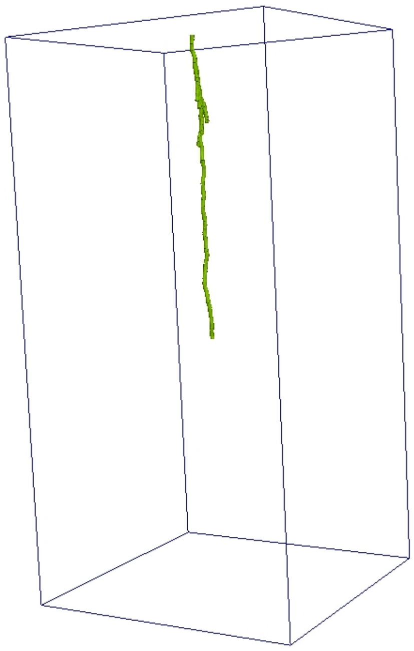


Figure 38: 5 - day old root, visualization with ParaView

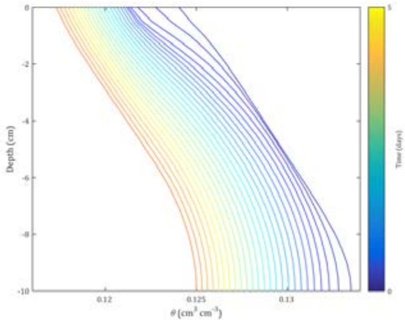


Figure 39: volumetric soil water content over the simulation period

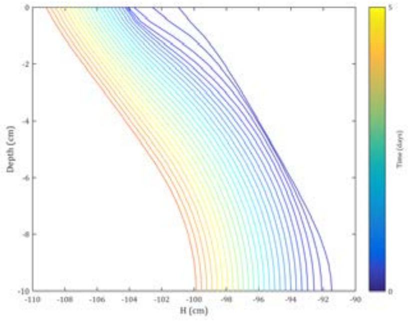


Figure 40: soil water potential over the simulation period

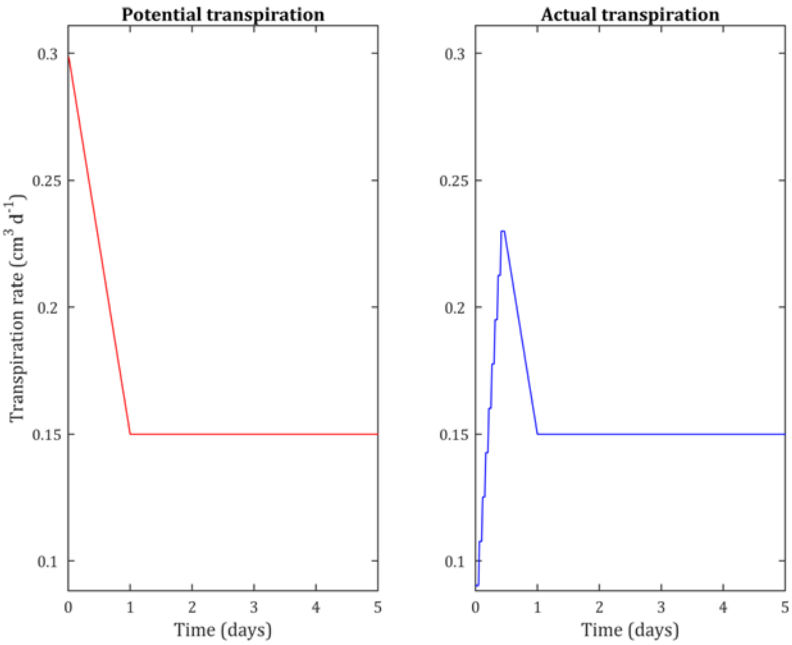


Figure 41: potential and actual transpiration rate during the simulation period

Example 6: Stomatal Regulation by Hormonal signaling

Scenario description

In the previous simulations transpiration rate was either imposed or a constant head boundary condition was applied. In this example transpiration rate is regulated by controlling the stomatal aperture as a function of foliar hormone concentration. Hormones are generated at each root segment as a function of the local xylem pressure in this segment and then transported with the xylem stream towards the root collar. Transpiration is regulated as a function of hormone concentration.

Input/output

Input:in_Signaling

- Total runtime (control.in): 0.5 days (discretization: 0.05 days, starting at day 4.0)
- Soil domain (mesh.in): 5.0 x 5.0 x 10 cm^3 (discretization: 0.125 cm^3)
- Soil initial conditions (mesh.in): pressure head = -300 cm constant
- Soil boundary conditions (bc.in): no flow at all sides
- Initial root system (RootSys): 4 days old - static root system, no growth
- Root boundary conditions (RootBC.in): flux bc, constant time
- Parameters for hormone signalling (CondRoot.in)

Main Outputs:

- Root fluxes and pressures (OutRoo.XX)
- Soil fluxes and pressures (Outfem.XX → same structure as nodes.in)
- Transpiration rate, pressure at root collar, hormone concentrations (log.1)

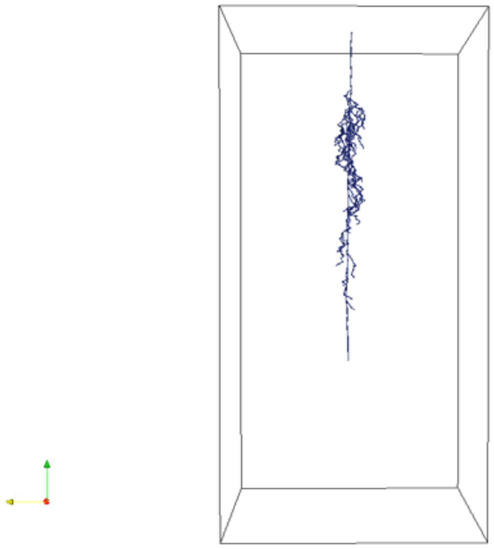


Figure 42: Four days old root system illustrated in Paraview

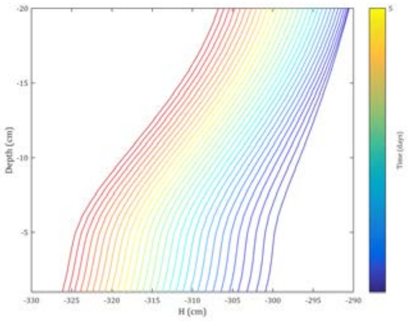


Figure 43: Hydraulic potential in the soil column

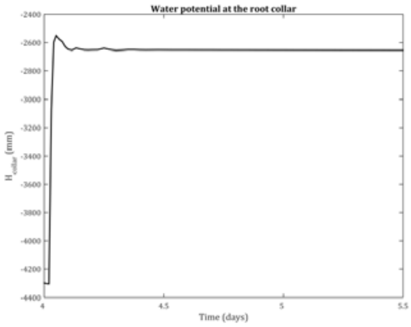


Figure 44: Water potantial at the root collar in the simulation period

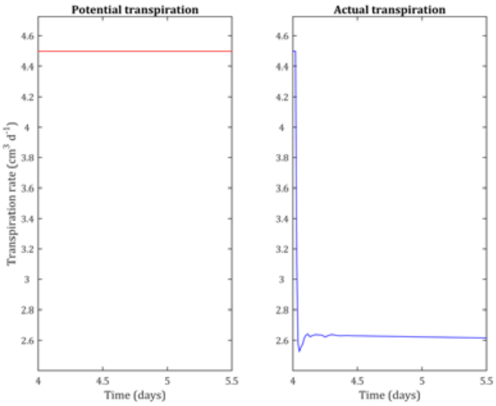


Figure 45: Potential transpiration and actual transpiration regulated by hormonal signaling

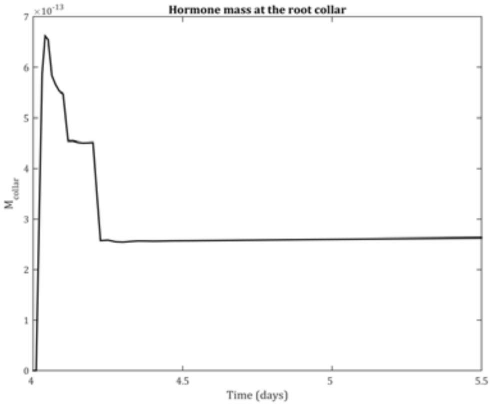
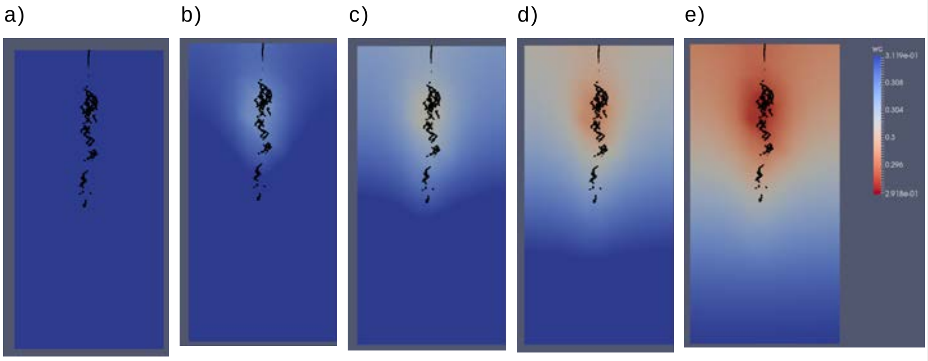


Figure 46: Hormone mass at the root collar in time



T=4.05 T = 4.40 T = 4.75 T = 5.10 T = 5.50

Figure 47 a - e: Water content at the domain cross section for five output times

Example 7: Solute transport with ParTrace

Description

This example simulates solute transport with the coupled R-SWMS ParTrace version. Initially a certain amount of particles is injected at the top soil surface and then distributed within the soil according to the water velocity field and a random diffusion component.

Input/output

Input: in_ParTrace

To run this example, the program first has to be recompiled. In Makefile.defs, Line 2, the flag for Partrace has to be set to 1.

Furthermore, in this example we are not going to consider the effect of osmotic potential, produced by the solute concentration, on root water uptake. This implies that our reflection coefficient, σ, needs to be set to 0. This can be done in Modules.f90, in MODULE RootData, on the variable named 'sigma'.

Then type 'make clean + ENTER' and 'make + ENTER'. Simulates solute injection into and flow of solutes in the soil, enrichment of solutes around the roots.

Partrace uses some extra input files that are found in the PartraceInOut folder and in the Input_PARTRACE.pTraceInpV12 file. It is important to note that the Partrace input files have to be adjusted to the current grid size!

- Total runtime (control.in): 0.5 days (discretisation: 0.05 days, starting at day 4.0)
- Initial root system is RootSys file (control.in): 4 days old - static root system, no growth
- Soil domain (mesh.in): 4.5 x 4.5 x 12 cm^3 (discretisation: 0.5 cm^3)
- Soil initial conditions (mesh.in): pressure head = -295 cm constant
- Soil boundary conditions (bc.in): no flow at all sides except by some water flow for the first 0.1 days in order to inject particles on the top soil.
- Root boundary conditions (BCRoot.in): flux bc, increasing with time

Main Outputs:

- Root fluxes and pressures (OutRoo.XX)
- Soil fluxes and pressures (Outfem.XX → same structure as nodes.in)
- Transpiration rate, pressure at root collar, hormone concentrations (log.1)

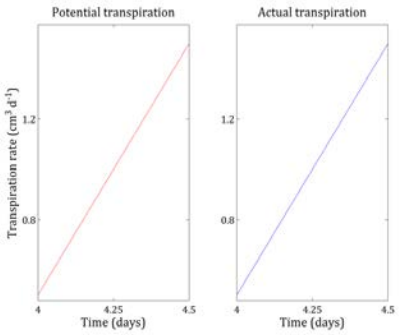


Figure 48: Potential and actual transpiration rates over the simulation preiod.

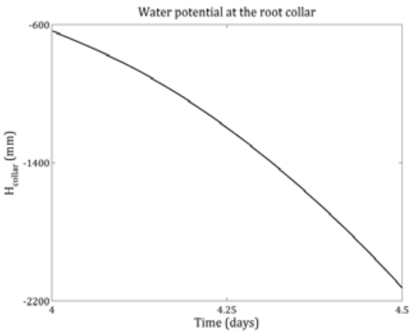


Figure 49: Water potential at the root collar over the simulation period.

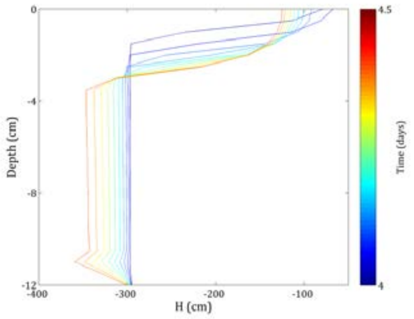


Figure 50: Soil water potential over depth and for all the simulation period.

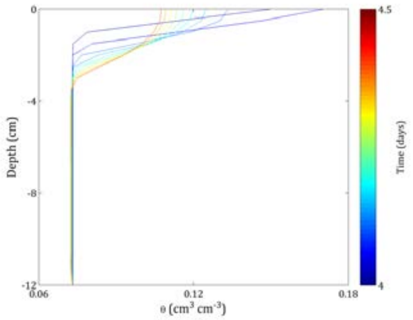


Figure 51: Volumetric soil water content over depth and for all the simulation period.

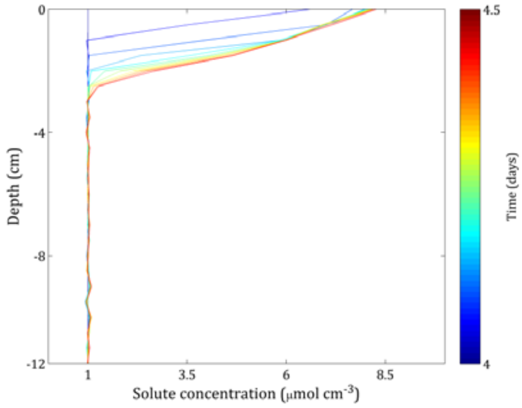


Figure 52: Solute concentration over depth and for all simulation period.

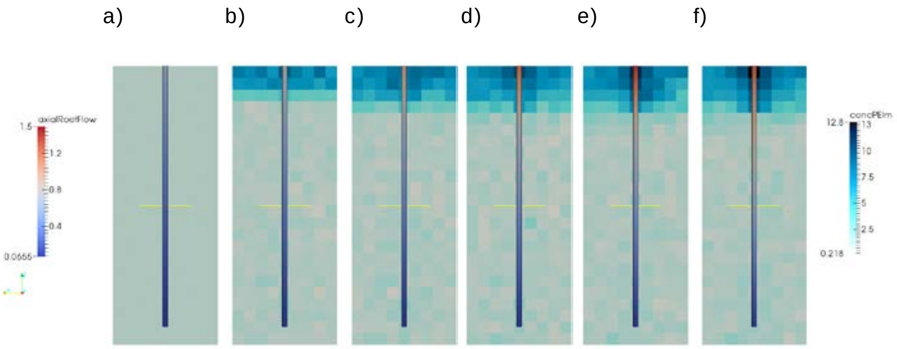


Figure 53: 3D distribution of solute concentration for yz plane for different simulation times (a) 4 days, b) 4.1 days, c) 4.2 days, d) 4.3 days, e) 4.4 days and f) 4.5 days). Visualization with Paraview.

Example 8: Solving the Doussan matrix

Description

This example simulates water flow in a simple root that consists of 4 nodes (0-3) that are connected by 3 segments as depicted in the figure.

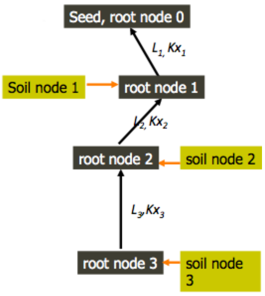


Figure 54.Simple root system consisting of 4 nodes connected by 3 segments.

Theoretical solution

Assumptions:

The boundary condition at the root collar (node 0) switches between a flux boundary condition (given by the transpiration flux) and a prescribed root xylem water potential if the root xylem reaches a certain threshold value. The soil matrix potential is constant and read in as an input file.

For flux boundary condition at the root collar, mass balance for each node is given by

At node 0: $J_{01}^x = T_{pot}$

At node 1: $J_{12}^x + J_1^r = J_{01}^x$

At node 2: $J_{23}^x + J_2^r = J_{12}^x$

At node 3: $J_3^r = J_{23}^x$

Inserting into the radial and axial fluxes gives

At node 0: $\frac{K_{x1}}{l}(h_0 - h_1) = T_{pot}$

At node 1: $\frac{K_{x2}}{l}(h_1 - h_2) + K_{rs}^1 s_1 (h_1 - h_1^s) = \frac{K_{x1}}{l}(h_0 - h_1)$

At node 2: $\frac{K_{x3}}{l}(h_2 - h_3) + K_{rs}^2 s_2 (h_2 - h_2^s) = \frac{K_{x2}}{l}(h_1 - h_2)$

At node 3: $K_{rs}^3 s_3 (h_3 - h_3^s) = \frac{K_{x3}}{l}(h_2 - h_3)$

This can be written in matrix form:

$$\begin{pmatrix} \frac{K_{x1}}{l} & -\frac{K_{x1}}{l} & 0 & 0 \\ -\frac{K_{x1}}{l} & \frac{K_{x2}}{l}+\frac{K_{x1}}{l}+K_{rs}^{1}s_1 & -\frac{K_{x2}}{l} & 0 \\ 0 & -\frac{K_{x2}}{l} & \frac{K_{x2}}{l}+\frac{K_{x3}}{l}+K_{rs}^{2}s_2 & -\frac{K_{x3}}{l} \\ 0 & 0 & -\frac{K_{x3}}{l} & \frac{K_{x3}}{l}+K_{rs}^{3}s_3 \end{pmatrix}\begin{pmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{pmatrix}=\begin{pmatrix} T_{pot} \\ K_{rs}^{1}s_1h_1^{s} \\ K_{rs}^{2}s_2h_2^{s} \\ K_{rs}^{3}s_3h_3^{s} \end{pmatrix}$$

or

$C \cdot H_x = Q_d$,

where***C*** is the conductance matrix, ***H***$_x$is the vector of root xylem total water potentials, and ***Q*** is a vector containing soil factors and the boundary condition at the root collar.

Q$_d$can be written in matrix form as

$Q_d = Q_i \cdot H_s + Q_{bc}$,

Where***Q***$_i$ is the matrix containing the root radial conductivities, ***H***$_s$ is the vector of soil water total potentials, and ***Q***$_{bc}$ contains the flux boundary condition,

$$Q_i = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & K_{rs}^{1}s_1 & 0 & 0 \\ 0 & 0 & K_{rs}^{2}s_2 & 0 \\ 0 & 0 & 0 & K_{rs}^{3}s_3 \end{pmatrix},$$

$$Q_d = \begin{pmatrix} 0 \\ h_1^{s} \\ h_2^{s} \\ h_3^{s} \end{pmatrix},$$

$$Q_{bc} = \begin{pmatrix} T_{pot} \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

For root xylem potential boundary condition$h_0 = h_{imp}$ at the root collar, mass balance for each node is given by

At node 0: $h_0 = h_{imp}$

At node 1: $J_{12}^{x} + J_1^{r} = J_{01}^{x}$

At node 2: $J_{23}^{x} + J_2^{r} = J_{12}^{x}$

At node 3: $J_3^{r} = J_{23}^{x}$

Inserting into the radial and axial fluxes gives

At node 0: $h_0 = h_{imp}$

At node 1: $\frac{K_{x2}}{l}(h_1 - h_2) + K_{rs}^{1}s_1(h_1 - h_1^{s}) = \frac{K_{x1}}{l}(h_{imp} - h_1)$

At node 2: $\frac{K_{x3}}{l}(h_2 - h_3) + K_{rs}^{2}s_2(h_2 - h_2^{s}) = \frac{K_{x2}}{l}(h_1 - h_2)$

At node 3: $K_{rs}^{3}s_3(h_3 - h_3^{s}) = \frac{K_{x3}}{l}(h_2 - h_3)$

This can be written in matrix form:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{K_{x2}}{l}+\frac{K_{x1}}{l}+K_{rs}^{1}s_1 & -\frac{K_{x2}}{l} & 0 \\ 0 & -\frac{K_{x2}}{l} & \frac{K_{x2}}{l}+\frac{K_{x3}}{l}+K_{rs}^{2}s_2 & -\frac{K_{x3}}{l} \\ 0 & 0 & -\frac{K_{x3}}{l} & \frac{K_{x3}}{l}+K_{rs}^{3}s_3 \end{pmatrix}\begin{pmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \end{pmatrix}=\begin{pmatrix} h_{imp} \\ K_{rs}^{1}s_1h_1^{s}+\frac{K_{x1}}{l}h_{imp} \\ K_{rs}^{2}s_2h_2^{s} \\ K_{rs}^{3}s_3h_3^{s} \end{pmatrix}$$

or

$C \cdot H_x = Q_d$.

Qd can again be written in matrix form as

$\boldsymbol{Q_d} = Q_i \cdot H_s + Q_{bc}$,

with Q_i and H_s as above, and

$$Q_{bc} = \begin{pmatrix} h_{imp} \\ \frac{K_{x1}}{l}h_{imp} \\ 0 \\ 0 \end{pmatrix}.$$

Given the soil water potentials, root axial conductances, root radial conductivities and boundary condition at the root collar, the root xylem water potential and water fluxes inside the root can be calculated. The routines in R-SWMS are as follows:

The complete matrix ***Q*** is estimated in the subroutine SetSnk. ***Q***ᵢ is calculated one time at each root growth time step, in the subroutine setupDou. The matrix of total water potential ***Qd*** is calculated in SetSnk at each time step. The matrix ***Qbc*** is estimated first in SetupDou and then updated in the subroutine SetSnk only if new boundary conditions at the collar have been defined in the file **Bcroot.in**. The subroutine solveroot is used to solve the Doussan matrices.

Input/output

Inputs:

- **RootSys**: Initial root system, static root system with 4 nodes, 3 segments
- **Control.in**: Total runtime 1 second, Root water uptake model Doussan, no root growth
- **Mesh.in**: htop prescribes the initial pressure head for the soil domain. We only simulate water flow inside the root, and we set the soil pressure head constant, htop = -300 cm, ICtype 1. Mesh.in is used to create the files Nodes.in and Grid.in. Here: Soil initial pressure head is constant, and set to -300 cm
- **BC.in**: All types of boundary conditions are set to 0.
- **BCroot.in**: Root boundary conditions, constant head bc, -1000 cm
- **CondRoot.in**: Root hydraulic properties are set.

Main Outputs:

Doussan matrices, omitting the first column and line that do not belong to the root system,

Diagonal elements of $Q_i = \begin{pmatrix} 6.78 \times 10^{-6} \\ 6.78 \times 10^{-6} \\ 6.78 \times 10^{-6} \end{pmatrix}$

$$Q_d = \begin{pmatrix} -86.64 \\ -2.03 \times 10^{-3} \\ -2.03 \times 10^{-3} \end{pmatrix}$$

$$Q_{bc} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Example 9: Crop cycle

Scenario description

Water uptake over the whole crop cycle

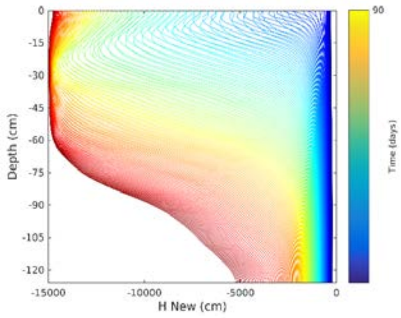
Input/output

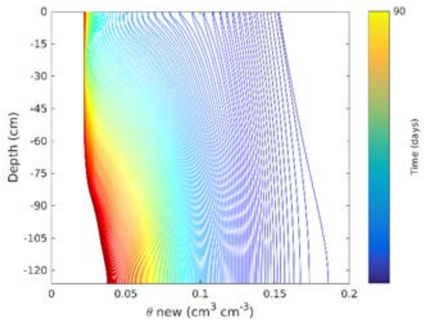
Input

- Total runtime (control.in): 80 days (discretization: 0.5 days, start at day 10)
- Root architecture type: Use of Couvreur model (control.in)
- No root growth (control.in)
- Continuous soil domain (control.in)
- Soil domain (mesh.in): 94500 cm3 (50*10*84 blocks of 1.5*1.5*1.5 cm side length)
- Soil initial conditions (mesh.in): pressure head = -300 cm at the top; equipotential distribution
- Soil boundary conditions (bc.in): no flow at top and sides , free drainage at the bottom
- Root boundary conditions (BCroot.in): sinusoidal flux

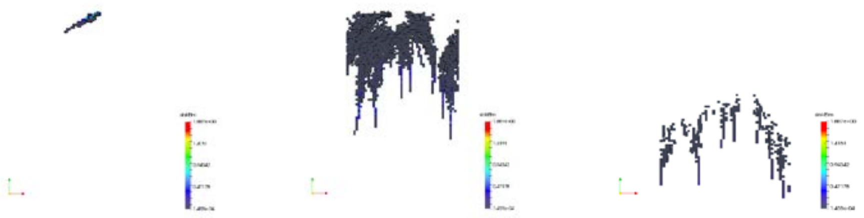
Main Outputs

- Soil fluxes and pressures (Outfem.XX → same structure as nodes.in)
- Sink (veloci.XXX.vtk)





Sink distribution at different days of the cycle



Day 10 Day 50 Day 90

Example 11: Root growth and uptake in a cylindrical soil column

Scenario description

This example simulates the water uptake of a root system in a cylindrical soil column. The root system growth is read in the rootsys file (origination file).

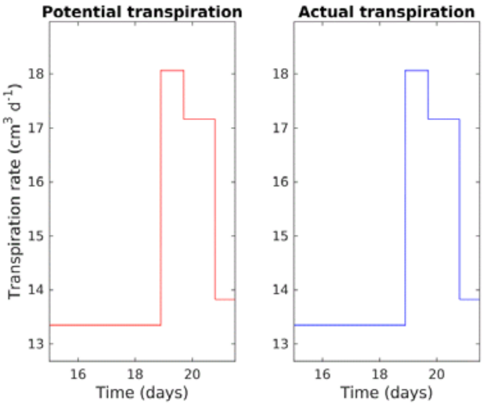
Input/output

Input

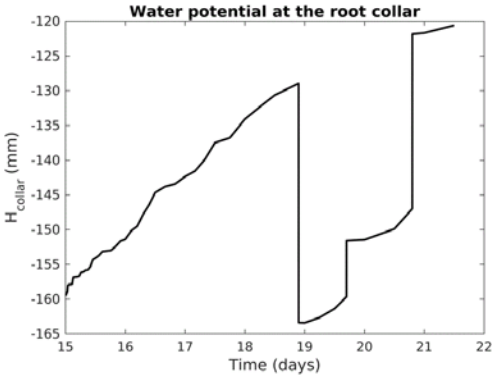
- Total runtime (control.in): 6.5 days (discretization: 0.5 days, start at day 15)
- Initial root architecture type: read in RootSys (control.in)
- Update root growth (control.in)
- Non-continuous soil domain (control.in)
- Soil domain (mesh.in): 250 cm3 (20*20*40 blocks of 0.25*0.25*0.25 cm side length)
- Soil initial conditions (mesh.in): pressure head = -26 cm at the top; equipotential distribution
- Soil domain shape: cylindrical column with a 2.5 cm radius
- Soil boundary conditions (bc.in): irrigation at the top , pressure head condition at the bottom
- Root boundary conditions (BCroot.in): free format flux
- Uniform radial and axial root conductivities (CondRoot.in)

Main Outputs

- Root fluxes, pressures, uptakes (OutRoo.XX)
- Soil fluxes and pressures (Outfem.XX → same structure as nodes.in)
- Actual transpiration rate, pressure at root collar (log1)


Potential transpiration
Actual transpiration
Transpiration rate (cm^3 d^{-1})
18
17
16
15
14
13
16
18
20
Time (days)
Time (days)

[1]


Water potential at the root collar
H_{collar} (mm)
-120
-125
-130
-135
-140
-145
-150
-155
-160
-165
15
16
17
18
19
20
21
22
Time (days)

[1]

Programmers guide

Directories

R-SWMS has different directories:

- All input files are **in** the in folder
- All output files are in the **out** folder
- Fortran source code is in the **src_rswms** folder and contains the main R-SWMS program. It contains also the grid generator and the coupled **RootTyp** c-code files
- **Partrace** source code and the interface to the main Fortran file can be found in the **src_partrace** folder. Partrace is written in C++
- The folder **sparskit** contains the linear solver library sparskit and provides solvers and preconditioners. The library is linked to the R-SWMS during the compiling process
- The folder **PartraceInOut** contains the in- and output files of Partrace

In addition to the different subfolders, the main R-SWMS folder contains several files:

- **Main.f90**: The main file of the R-SWMS program – everything starts from here
- **Makefile.defs**: Defines make variables, compilers, compiler flags, linking flags etc.
- **Makefile**: includes the Makefile.defs and takes care of the compilation and of the building of executables
- **Readme**: small documentation – update it with every svn commit!
- **Input_PARTRACE.pTraceInpV12**: Main input file for Partrace

Guidelines

An important characteristic of source code is that it is written only once but usually it is read manytimes (e.g. when debugging things, adding features, etc.). For this reason, good programming frameworks always aim to be as readable as possible, even if comes with higher effort to write them in first place. The remainder of this section is very similar of the DUNE and DuMuX coding guidelines found at (http://www.dune-project.org/) and (http://www.dumux.org). These guidelines are also recommended for coding with R-SWMS as developer and user.

- Comments: they are helpful! Please document freely what each part of your code does.
- all comments / documentation is in English
- Variables should be named as self-explaining as possible: especially abbreviations should be avoided (saturation instead of S)
- Files: source file names (*.f90) should start with an capital, input and output file names should consist of lower case letters exclusively

Documentation

R-SWMS, as any software project of similar complexity, will stand and fall withthe quality of its documentation. Therefore it is of paramount importance that you documentwell everything you do! We use the Doxygen system to extract easily-readable documentationfrom the source code. Please use its syntax everywhere. In particular, please commentall

- new files
- subroutines
- subroutine Parameters (in / out)
- return values

Here is an example on how to comment a subroutine:

```
!> ### reads main inputs ###

SUBROUTINE Applic(dt)

!> \param nPt total number of nodal points

!> \param nBCpts total number of nodal points with specified BC

!> \param nElm total number of elements

!> \param ne* number of elements (half-cuboids) in '*'-direction

!> \param nel nex*ney (number of elements per horizontal element-layer)
```

Download doxygen: https://sourceforge.net/projects/doxygen/

Doxygen manual: http://www.stack.nl/~dimitri/doxygen/manual/docblocks.html

To run doxygen, it’s necessary to tick

- Wizard/Optimize for fortran output
- Expert/build/extract all instead of extract local classes

To view the doxygen documentation, open any of the html files in the dropbox folder.

Subversion repository

For those (all of you!!!) who work with Subversion (svn)some basics are given in this section. For further reading please check out the Subversion User Manual (http://subversion.apache.org/) where you will also finda "High Speed Turorial" in the appendix.

The five most important commands are

- svn checkout
- svn status
- svn update
- svn add
- svn commit

The first one (svn checkout) will create acopy of the trunk version from the svn server on your local system. This is something you should have already done during the installation process. You might not do it again. The svn address is available on request.

The second one (svn status) will show you the actual status of your working copy compared to the svn repository, e.g. if you have local changes in your working copy (status M - modified), or if files exist which are not under subversion control (status ?).

Usesvn updateto get the latestchanges in the repository (commits from other users). In order to add a new file to the repositorythe following steps have to be taken:

1) **svn update**
The first step is to update your R-SWMS folders/files.You should execute this command at least in your folder where you want to add a new file (e.g. src_rswms)
2) **svn add** YOURFILES
In your folder: use svn add YOURFILES to add your files
3) **svn commit** YOURFILES
Use svn commit. This uploads all your changes of your file to the svn server. You will be asked to briefly explain the content of your commit in an editor. If you don’t specify any file, every change in that folder will be commited.

The above shows you the necessary steps if you use the command line. There are also other toolsproviding a graphical user interface for using svn like tortoisesvn (http://www.tortoisesvn.net) or eclipse (http://www.eclipse.org/subversive/). The necessary steps for addingand committing stay the same.

How to create branches with svn:

a tutorial but unfortunately in German =/
http://web-rocker.de/2011/01/subversion-howto-branch-merge-reintegrate/

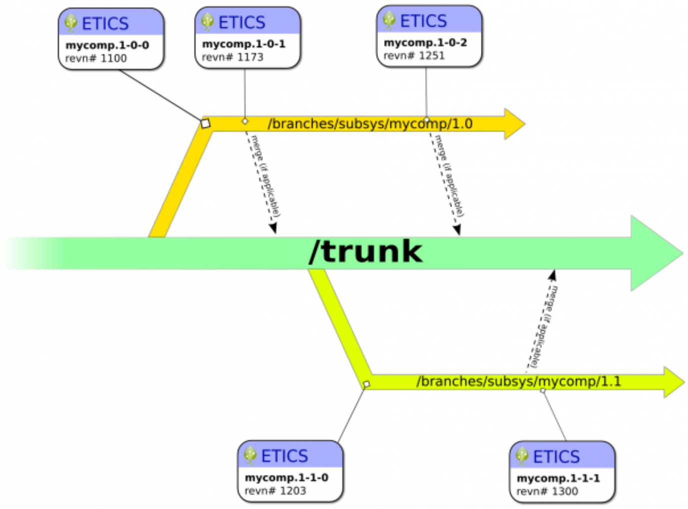


Figure 55 Illustration of trunk and branches

The branches are located at svn://icg4lts.icg.kfa-juelich.de/rswms/Branches

1. create a new directory for your own branch
   svn mkdir svn://icg4lts.icg.kfa-juelich.de/rswms/Branches/YOURNAME
2. copy the current rswms version to your branch (svn copy source sink)
   svn copy svn://icg4lts.icg.kfa-juelich.de/rswms/RSWMS7 svn://icg4lts.icg.kfa-juelich.de/rswms/Branches/YOURNAME –m 'add your comment here'
3. check out your private working copy
   svn co svn://icg4lts.icg.kfa-juelich.de/rswms/Branches/YOURNAME
4. update your changes in the branch
   svn ci svn://icg4lts.icg.kfa-juelich.de/rswms/Branches/YOURNAME –m 'your comment'
5. update changes on the trunk to your branch
   svn merge svn://icg4lts.icg.kfa-juelich.de/rswms/RSWMS7
6. iterate steps 4 and 5 *regularly* until you are finished programming
7. reintegrate the branch to the trunk
   first we have to check out the trunk to a new repository and work from there
   svn co svn://icg4lts.icg.kfa-juelich.de/rswms/RSWMS7
   now some 'magic'
   svn merge --reintegrate svn://icg4lts.icg.kfa-juelich.de/rswms/Branches/YOURNAME
   svn ci –m 'reintegration comment'
8. delete the branch (just in case)
   svn delete svn://icg4lts.icg.kfa-juelich.de/rswms/Branches/YOURNAME –m 'your comment'

Appendices

Summary of variable and parameters (to be completed)

Description of plant hydraulic parameters in soil-plant system

The definitions of conductivity and conductance are sometimes misleading in the literature. Proper definitions depend whether the transfer equation is written in terms of flux density (then it is a conductivity) or of flow (conductance). Units then depend whether the potential difference is explained n water head or in pressure. Finally, transfer equations typically write in potential differences for membranes, while in tubes, potential gradients are used. The table below summarizes most of the definitions found in this manual, and in literature.

Table 9. Overview of conductivity and conductance definitions

	Flow (J) $[L^3T^{-1}]$		**Flux density (q) $[LT^{-1}]$**	
	Resistance	**Conductance**	**Resistivity**	**Conductivity**
Pressure gradient	$(d\psi/dl)/J$ $[TPL^{-4}]$	$J/(d\psi/dl)$ $[L^4T^{-1}P^{-1}]$	$(d\psi/dl)/q$ $[PTL^{-2}]$	$q/(d\psi/dl)$ $[L^2T^{-1}P^{-1}]$
Pressure difference	$\Delta\psi/J$ $[TPL^{-3}]$	$J/\Delta\psi$ $[L^3T^{-1}P^{-1}]$	$\Delta\psi/q$ $[PTL^{-1}]$	$q/\Delta\psi$ $[LT^{-1}P^{-1}]$
Head gradient	$(dH/dl)/J$ $[TL^{-3}]$	$J/(dH/dl)$ $[L^3T^{-1}]$	$(dH/dl)/q$ $[TL^{-1}]$	$q/(dH/dl)$ $[LT^{-1}]$
Head difference	$\Delta H/J$ $[TL^{-2}]$	$J/\Delta H$ $[L^2T^{-1}]$	$\Delta H/q$ $[T]$	$q/\Delta H$ $[T^{-1}]$

References

Barber, S., A. 1995. *Soil Nutrient Bioavailability: A Mechanistic Approach*.

Bear, J. 1972. *Dynamics of Fluids in Porous Media*. American Elsevier Pub. Co.

Bechtold, M., J. Vanderborght, O. Ippisch, and H. Vereecken. 2011. "Efficient Random Walk Particle Tracking Algorithm for Advective-Dispersive Transport in Media with Discontinuous Dispersion Coefficients and Water Contents." *Water Resources Research* 47 (10). doi:10.1029/2010WR010267.

Bengough, A.G., B.M. McKenzie, P.D. Hallett, and T.A. Valentine. 2011. "Root Elongation, Water Stress, and Mechanical Impedance: A Review of Limiting Stresses and Beneficial Root Tip Traits." *Journal of Experimental Botany* 62 (1): 59–68. doi:10.1093/jxb/erq350.

Celia, Michael A., Efthimios T. Bouloutas, and Rebecca L. Zarba. 1990. "A General Mass-Conservative Numerical Solution for the Unsaturated Flow Equation." *Water Resources Research* 26 (7): 1483–96. doi:10.1029/WR026i007p01483.

Clausnitzer, V., and J.W. Hopmans. 1994. "Simultaneous Modeling of Transient Three-Dimensional Root Growth and Soil Water Flow." *Plant and Soil* 164 (2): 299–314.

Couvreur, Valentin. 2013. "Emergent Properties of Plant Hydraulic Architecture: A Modelling Study." Louvain-la-Neuve, Belgique: Université catholique de Louvain.

Couvreur, V., J. Vanderborght, and M. Javaux. 2012. "A Simple Three-Dimensional Macroscopic Root Water Uptake Model Based on the Hydraulic Architecture Approach." *Hydrology and Earth System Sciences* 16 (8): 2957–71. doi:10.5194/hess-16-2957-2012.

Doussan, Claude, Gilles Vercambre, and Loïc Pagè. 1998. "Modelling of the Hydraulic Architecture of Root Systems: An Integrated Approach to Water Absorption—Distribution of Axial and Radial Conductances in Maize." *Annals of Botany* 81 (2): 225–32. doi:10.1006/anbo.1997.0541.

Durner, Wolfgang. 1994. "Hydraulic Conductivity Estimation for Soils with Heterogeneous Pore Structure." *Water Resources Research* 30 (2): 211–23. doi:10.1029/93WR02676.

French, A., S. Ubeda-Tomás, T. J. Holman, M. J. Bennett, and T. Pridmore. 2009. "High-Throughput Quantification of Root Growth Using a Novel Image-Analysis Tool." *Plant Physiology* 150 (4): 1784–95.

Gregory, P.J. 2006. "Roots, Rhizosphere and Soil: The Route to a Better Understanding of Soil Science?" *European Journal of Soil Science* 57 (1): 2–12. doi:10.1111/j.1365-2389.2005.00778.x.

Hamza, M.A., and L.A.G. Aylmore. 1992. "Soil Solute Concentration and Water Uptake by Single Lupin and Radish Plant Roots." *Plant and Soil* 145 (2): 187–96. doi:10.1007/BF00010347.

Hopmans, J.W., and K.L. Bristow. 2001. *Current Capabilities and Future Needs of Root Water and Nutrient Uptake Modeling*. Vol. 77. Advances in Agronomy. http://www.scopus.com/inward/record.url?eid=2-s2.0-35448935307&partnerID=40&md5=d8c48885289b1b97f379c0c7b4120fef.

Huber, Katrin, Jan Vanderborght, Mathieu Javaux, Natalie Schröder, Ian C. Dodd, and Harry Vereecken. 2014. "Modelling the Impact of Heterogeneous Rootzone Water Distribution on the Regulation of Transpiration by Hormone Transport And/or Hydraulic Pressures." *Plant and Soil* 384 (1-2): 93–112. doi:10.1007/s11104-014-2188-4.

Javaux, Mathieu, Tom Schroeder, Jan Vanderborght, and Harry Vereecken. 2008. “Use of a Three-Dimensional Detailed Modeling Approach for Predicting Root Water Uptake.” *Vadose Zone Journal* 7 (3): 1079–88. doi:10.2136/vzj2007.0115.

Kochian, Leon V, and William J Lucas. 1982. “Potassium Transport in Corn Roots : I. Resolution of Kinetics into a Saturable and Linear Component.”*Plant Physiology* 70 (6): 1723–31.

Kolb E, Hartmann C, Genet P 2012 Radial force development during root growth measured by phytoplasticity. . Plant and Soil 360: 19-35.

LaBolle, E.M., J. Quastel, G.E. Fogg, and J. Gravner. 2000. “Diffusion Processes in Composite Porous Media and Their Numerical Integration by Random Walks: Generalized Stochastic Differential Equations with Discontinuous Coefficients.” *Water Resources Research* 36 (3): 651–62. doi:10.1029/1999WR900224.

Leitner, Daniel, Félicien Meunier, Gernot Bodner, Mathieu Javaux, and Andrea Schnepf. 2014. “Impact of Contrasted Maize Root Traits at Flowering on Water Stress Tolerance – A Simulation Study.” *Field Crops Research*, June. doi:10.1016/j.fcr.2014.05.009.

Leitner, D., S. Klepsch, G. Bodner, and A. Schnepf. 2010. “A Dynamic Root System Growth Model Based on L-Systems.” *Plant and Soil* 332 (1): 177–92.

Lichtner, P.C., S. Kelkar, and B. Robinson. 2002. “New Form of Dispersion Tensor for Axisymmetric Porous Media with Implementation in Particle Tracking.” *Water Resources Research* 38 (8): 211–2116.

Millington, R.J., and J.P. Quirk. 1961. “Permeability of Porous Solids.” *Transactions of the Faraday Society* 57: 1200–1207. doi:10.1039/TF9615701200.

Mualem, Yechezkel. 1976. “New Model for Predicting the Hydraulic Conductivity of Unsaturated Porous Media.” *Water Resources Research* 12 (3): 513–22.

Neuendorf, O. 1997. “Numerische 3D-Simulation Des Stofftransports in Einem Heterogenen Aquifer.” Jülich: Forschungszentrum Jülich GmbH.

Pagès, Loïc, Gilles Vercambre, Jean-Louis Drouet, François Lecompte, Catherine Collet, and Jacques Le Bot. 2004. “Root Typ: A Generic Model to Depict and Analyse the Root System Architecture.” *Plant and Soil* 258 (1): 103–19. doi:10.1023/B:PLSO.0000016540.47134.03.

Salamon, P., D. Fernàndez-Garcia, and J.J. Gómez-Hernández. 2006. “Modeling Mass Transfer Processes Using Random Walk Particle Tracking.” *Water Resources Research* 42 (11). doi:10.1029/2006WR004927.

Schröder, N. 2013. “Three-Dimensional Solute Transport Modeling in Coupled Soil and Plant Root Systems.” Université Catholique de Louvain and Forschungszentrum Jülich GmbH.

Schröder, Natalie, Naftali Lazarovitch, Jan Vanderborght, Harry Vereecken, and Mathieu Javaux. 2014. “Linking Transpiration Reduction to Rhizosphere Salinity Using a 3D Coupled Soil-Plant Model.” *Plant and Soil* 377 (1-2): 277–93. doi:10.1007/s11104-013-1990-8.

Shani, Uri, and Alon Ben-Gal. 2005. “Long-Term Response of Grapevines to Salinity: Osmotic Effects and Ion Toxicity.” *American Journal of Enology and Viticulture* 56 (2): 148–54.

Siddiqi, M.Y., A.D.M. Glass, T.J. Ruth, and T.W. Rufty Jr. 1990. “Studies of the Uptake of Nitrate in Barley: I. Kinetics of 13NO3 - Influx.” *Plant Physiology* 93 (4): 1426–32.

Simunek, J., K. Huang, and M. Th Van Genuchten. 1995. “The SWMS_3D Code for Simulating Water Flow and Solute Transport in Three-Dimensional Variably-Saturated Media.” *US Salinity Laboratory Research Report* 139. http://pc-progress.com/Documents/programs/SWMS_3D.pdf.

Somma, F., V. Clausnitzer, and Jan W. Hopmans. 1997. *An Algorithm for the Three-Dimensional, Simultaneous Modeling of Root Growth, Transient Soil Water Flow, and Solute Transport and Uptake*. version 2.1.

Somma, F., J.W. Hopmans, and V. Clausnitzer. 1998. “Transient Three-Dimensional Modeling of Soil Water and Solute Transport with Simultaneous Root Growth, Root Water and Nutrient Uptake.” *Plant and Soil* 202 (2): 281–93. doi:10.1023/A:1004378602378.

Tardieu, F., and W. J. Davies. 1993. “Integration of Hydraulic and Chemical Signalling in the Control of Stomatal Conductance and Water Status of Droughted Plants.” *Plant, Cell & Environment* 16 (4): 341–49. doi:10.1111/j.1365-3040.1993.tb00880.x.

Tardieu, F., J. Zhang, and D. J. G. Gowing. 1993. “Stomatal Control by Both [ABA] in the Xylem Sap and Leaf Water Status: A Test of a Model for Draughted or ABA-Fed Field-Grown Maize.” *Plant, Cell & Environment* 16 (4): 413–20. doi:10.1111/j.1365-3040.1993.tb00887.x.

Tompson, A.F.B., and L.W. Gelhar. 1990. “Numerical Simulation of Solute Transport in Three-Dimensional, Randomly Heterogeneous Porous Media.” *Water Resources Research* 26 (10): 2541–62. doi:10.1029/90WR01127.

Tracy S 2013 The response of root system architecture to soil compaction. Philosophy. Nottingham.

Trapp S 2000 Modelling uptake into roots and subsequent translocation of neutral and ionisable organic compounds. Pest Management Science 56: 767-778.

Trapp S 2004 Plant Uptake and Transport Models for Neutral and Ionic Chemicals. Environmental Science and Pollution Research 11: 33-39.

Trapp S 2007 Fruit tree model for uptake of organic compounds from soil and air. SAR and QSAR in Environmental Research 18: 367-387.

Tuzet, A., Perrier A. and R. Leunin. 2003. A coupled model of stomatal conductance, photosynthesis and transpiration. Plant, cell and environment 26: 1097-1116.

Van Genuchten, M.Th. 1980. “Closed-Form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils.” *Soil Science Society of America Journal* 44 (5): 892–98.

Whalley W, To J, Kay B, Whitmore A 2007 Prediction of the penetrometer resistance of soils with models with few parameters Geoderma 137: 370-377.