Documentation: Use of Richards Equation to Model Soil Moisture

Saurabh Kumar

1. INTRODUCTION

In the unsaturated zone, flow is governed by Richards' equation which is a combination of Darcy-Buckingham and the continuity equations [1]. In the current study, 1-D mixed form of the Richards Equation is used. It included both moisture content and pressure head as unknown. The mixed form of Richards equation is more mass conservative as compared to pressure form and it can be solved in a computational efficient manner [2]. The present documentation describes both the mathematical formulation and the numerical implementation used in the Python code base (`Richards_Equation.py`, `vg_models.py`, `re_model_function_files.py`, `FeddesRootModel.py`) that solves the Richards equation to simulate soil moisture dynamics in an agricultural field. The current documentation is almost entrirely generated using CHATGPT and I will be editing it further.

2. Mathematical Formulation

2.1 Governing equation

The vertical flow of water in the unsaturated soil is described by the mixed-form Richards equation. A general 1D Richards equation is given below:

$$[C(h) + S_e S_s] \frac{\partial h}{\partial t} = \nabla \cdot [K(h) \nabla (h + z)] \pm S(h, z, t), \tag{1}$$

where h [m] is the pressure head, t [T] is time, z [L] is the vertical coordinate (positive upwards), $K(h)$ [L T^{-1}] is the unsaturated hydraulic conductivity, $C(h)$ [1/L] is the specific moisture capacity, and $S(h, z, t)$ [T^{-1}] is the sink term representing root water uptake. S_s [L^{-1}] is the specific storage coefficient, and S_e [-] is the effective saturation (degree of saturation). In the present implementation, S_s is set to zero so that $C(h)$ controls storage changes.

The soil profile extends from the lower boundary $z = z_\mathrm{b}$ (bottom of the domain) to the upper boundary $z = z_\mathrm{s}$ (soil surface).

a) Initial condition

At the beginning of the simulation, the pressure head distribution is prescribed as

$$h(z, 0) = h_0(z), \qquad z_\mathrm{b} \le z \le z_\mathrm{s}, \tag{2}$$

where $h_0(z)$ is obtained from observed soil tension data interpolated onto the numerical grid.

b) Upper boundary condition (soil surface)

At the soil surface ($z = z_\mathrm{s}$), a flux-type boundary condition is imposed based on rainfall and evaporation:

$$-K(h) \left(\frac{\partial h}{\partial z} - 1 \right) \bigg|_{z=z_\mathrm{s}} = q_\mathrm{top}(t), \tag{3}$$

where $q_\mathrm{top}(t)$ is the net surface flux (positive downward), typically defined as

$$q_\mathrm{top}(t) = R(t) - E_a(t), \tag{4}$$

with $R(t)$ the rainfall rate and $E_a(t)$ the actual soil evaporation (and/or transpiration component acting at the surface).

When rainfall exceeds the infiltration capacity and surface ponding occurs, the flux boundary in (3) is replaced by a head-type boundary condition

$$h(z_\mathrm{s}, t) = h_\mathrm{pond}(t), \tag{5}$$

where $h_{\text{pond}}(t)$ is a near-saturated pressure head corresponding to a prescribed effective saturation at the surface.

c) Lower boundary condition (free drainage)

At the lower boundary ($z = z_{\text{b}}$), a free-drainage condition is applied, which corresponds to a unit hydraulic gradient:

$$\left.\frac{\partial(h+z)}{\partial z}\right|_{z=z_{\text{b}}} = 0, \tag{6}$$

or, equivalently, in terms of the Darcy flux

$$q_{\text{bottom}}(t) = -K\big(h(z_{\text{b}},t)\big)\left.\left(\frac{\partial h}{\partial z} - 1\right)\right|_{z=z_{\text{b}}}. \tag{7}$$

In the implementation, this condition is enforced by the `bottom_free_drainage` routine in the numerical solver

2.2 Soil water retention curve

The volumetric water content $\theta(h)$ is described by the van Genuchten model:

$$\theta(h) = \begin{cases} \theta_r + (\theta_s - \theta_r)\left[1 + (\alpha|h|)^N\right]^{-m}, & h \le 0, \\ \theta_s, & h > 0, \end{cases} \tag{8}$$

where θ_r is the residual water content, θ_s is the saturated water content, α [L^{-1}] and N [-] are shape parameters, and

$$m = 1 - \frac{1}{N}. \tag{9}$$

The effective saturation (degree of saturation) is

$$S_e(h) = \frac{\theta(h) - \theta_r}{\theta_s - \theta_r}. \tag{10}$$

2.3 Hydraulic conductivity

The unsaturated hydraulic conductivity $K(h)$ is given by a van Genuchten–Mualem type relationship:

$$K(h) = \begin{cases} K_{\text{sat}}\,(1+\beta)^{-\frac{5m}{2}}\left[(1+\beta)^m - \beta^m\right]^2, & h < 0, \\ K_{\text{sat}}, & h \ge 0, \end{cases} \tag{11}$$

with

$$\beta = (\alpha|h|)^N \tag{12}$$

where K_{sat} is the saturated hydraulic conductivity.

2.4 Specific moisture capacity

The specific moisture capacity $C(h)$ is obtained as

$$C(h) = \begin{cases} (N-1)(\theta_s - \theta_r)\dfrac{|h|^{N-1}}{|h_s|^N\,(1+\beta)^{m+1}}, & h \le 0, \\ 0, & h > 0, \end{cases} \tag{13}$$

with

$$h_s = \frac{1}{\alpha},$$

(14)

3. NUMERICAL IMPLEMENTATION

3.1 Coordinate system and vertical grid

In the numerical model, the vertical coordinate z is taken as positive upwards, with the lower boundary at $z = 0$ and the soil surface at $z = z_{\max}$. For convenience, a *depth* coordinate is defined as

$$\text{depth} = z_{\max} - z, \tag{15}$$

such that depth increases downward from the soil surface.

The one-dimensional soil profile is discretised into a set of cell-centred control volumes:

- uniform grid spacing dz is used throughout the profile;
- cell centres z_i and cell thicknesses dz_i are stored in the `ProfileGridSpec` data structure;
- the corresponding depths depth_i are used to assign soil horizons and root distribution.

3.2 Spatial discretisation

The mixed-form Richards equation (1) is discretised using a finite-volume/finite-difference approach with cell-centred unknowns. For each interior node, a discrete balance is written over the control volume:

$$[C(h_i) + S_e(h_i)S_s]\frac{h_i^{n+1} - h_i^n}{\Delta t} = \frac{q_{i-\frac{1}{2}}^{n+1} - q_{i+\frac{1}{2}}^{n+1}}{dz_i} - S_i^{n+1}, \tag{16}$$

where $q_{i\pm\frac{1}{2}}$ are the Darcy fluxes at the cell faces, n denotes the time level, and S_i is the root water uptake sink at node i.

Hydraulic conductivities at cell faces are obtained from the arithmetic mean of neighbouring cell conductivities, and the fluxes are computed as

$$q_{i+\frac{1}{2}} = -K_{i+\frac{1}{2}}\left(\frac{h_{i+1} - h_i}{dz_{i+\frac{1}{2}}} - 1\right), \tag{17}$$

consistent with the sign convention adopted in (**??**).

The assembly of the spatial operator is implemented in

- `central_zone(...)` for interior nodes,
- `bottom_free_drainage(...)` for the lower boundary,
- `top_flux_boundary(...)` and `fixed_headBC(...)` for the upper boundary.

These routines construct the coefficient matrix **A** and right-hand side vector **RHS** of the algebraic system

$$\mathbf{AX} = \mathbf{RHS}, \tag{18}$$

with unknown vector **X** containing the total hydraulic head $H = h + z$ at all grid nodes.

3.3 Time discretisation and nonlinear solution

Time integration is performed using an implicit scheme with adaptive time stepping. At each time step, a Picard-type fixed-point iteration is used to handle the nonlinearity in $C(h)$ and $K(h)$:

1) Start from the solution at time level n as an initial guess for h^{n+1}.
2) Evaluate $C(h)$, $K(h)$, $\theta(h)$ and $S_e(h)$ via the van Genuchten model implemented in `vgModel(...)` from `vg_models.py`.
3) Compute the root water uptake distribution $S(h, z, t)$ using `RootUptakeModel(...)` from `FeddesRootModel.py`.
4) Assemble **A** and **RHS** using the current hydraulic properties and boundary conditions.
5) Solve the linear system for the updated total head $H^{(k+1)}$ (and hence $h^{(k+1)}$).
6) Evaluate the root-mean-square error between successive iterates and stop when the convergence tolerance is satisfied.

The time step Δt is adapted based on convergence behaviour using minimum/maximum limits (`dt_min`, `dt_max`) and multiplicative increase/decrease factors (`plus_factor`, `dec_factor`) in the `TimeSpec` data structure.

3.4 Boundary conditions and surface ponding

At the lower boundary, a free-drainage condition is used, which corresponds to a unit hydraulic gradient:

$$\frac{\partial(h+z)}{\partial z} = 0 \quad \Rightarrow \quad q_{\text{bottom}} = -K(h_{\text{bottom}}). \tag{19}$$

This is implemented in `bottom_free_drainage(...)`.

At the soil surface, a flux boundary condition is applied based on rainfall and evaporation:

$$q_{\text{top}} = R(t) - E_a(t), \tag{20}$$

where $R(t)$ is the rainfall rate and $E_a(t)$ is the actual soil evaporation. The routine `top_flux_boundary(...)` assembles this boundary condition.

When rainfall exceeds the infiltration capacity, a surface ponding algorithm is invoked via `ponding_flux(...)` in `re_model_function_files.py`. This routine:

- estimates a near-saturated pressure head h_{top} based on a chosen effective saturation $S_e \approx 1$;
- computes the corresponding hydraulic conductivity and maximum admissible surface flux;
- if the applied rainfall flux exceeds this limit, switches the top boundary to a fixed-head condition using `fixed_headBC(...)`.

4. CODE STRUCTURE

The implementation is organised into four main Python modules.

4.1 `Richards_Equation.py` (driver script)

This is the main driver that orchestrates the simulation:

- imports numerical and plotting libraries;
- reads meteorological forcing and soil hydraulic parameter tables from Excel files;
- constructs instances of:
  - `ProfileGridSpec` (soil profile and grid),
  - `TimeSpec` (time discretisation),
  - `RWUSpec` (root water uptake parameters);
- specifies the initial pressure head profile $h(z, t = 0)$ from observed tension data;
- calls `Richards_Solver(...)` in `re_model_function_files.py`;
- performs post-processing, including spatial interpolation and integrated water storage calculations;
- generates plots comparing simulated and observed soil water tension and soil moisture.

4.2 `vg_models.py` (soil hydraulic model)

This module implements the van Genuchten–Mualem relationships:

- `VG_theta(h, vgData)`: water retention curve $\theta(h)$;
- `VG_K(h, vgData)`: unsaturated hydraulic conductivity $K(h)$;
- `Van_Genuchten_specific_moisure_capacity(h, vgData)`: specific moisture capacity $C(h)$;
- `Van_Genuchten_Ss(h, vgData)`: specific storage S_s (set to zero in the current application);
- `vgModel(h, vgData)`: convenience wrapper returning (θ, K, C, S_s) for a given pressure head vector.

4.3 `re_model_function_files.py` (Richards solver utilities)

This module contains the numerical utilities required by the main solver:

- `soil_grid(...)`: generates the vertical grid and depth arrays;
- `assign_vg(depth, df, extend_to)`: assigns van Genuchten parameters to each grid node based on depth and horizon definitions;
- zone assembly routines: `central_zone(...)` for interior nodes, `bottom_free_drainage(...)`, `top_flux_boundary(...)`, `fixed_headBC(...)` for boundary nodes;
- `ponding_flux(...)`: evaluates maximum infiltration flux and handles surface ponding;
- `Richards_Solver(profileData, timeData, vgData, MetData, RWUData, h_ini)`: core time-stepping routine solving the mixed-form Richards equation;

- post-processing routines:
  - `spatial_interpolation(...)`: interpolates $h(z,t)$ to a user-defined vertical resolution and output times;
  - `prcoessed_results(...)`: computes θ, S_e and integrated water storage in the top soil layers.

4.4 `FeddesRootModel.py` (root uptake model)

This module provides the Feddes-type root water uptake model:

- `f2(psi, psi_a, psi_d, psi_w)`: pressure-head-dependent stress reduction function;
- `f1(depth, a, Lr, zmax)`: vertical root density distribution function;
- `RootUptakeModel(h, RWUData, profileData, Ep_current)`: combines f_1 and f_2 with potential evapotranspiration to obtain the sink term $S(h,z,t)$.

5. ROOT WATER UPTAKE MODEL

5.1 Feddes stress reduction function

Root water uptake is reduced under both very wet and very dry conditions. The Feddes-type reduction function $f_2(\psi)$ is defined in `f2(...)` as:

- $f_2 = 0$ for $\psi \geq \psi_a$ (anaerobiosis due to waterlogging);
- $f_2 = 1$ for $\psi_d < \psi < \psi_a$ (optimal pressure head range);
- f_2 decreases linearly from 1 to 0 between ψ_d and ψ_w;
- $f_2 = 0$ for $\psi \leq \psi_w$ (wilting point).

Here ψ_a, ψ_d and ψ_w are crop-specific parameters collected in the `RWUSpec` structure.

5.2 Root distribution with depth

The vertical distribution of roots is represented by the function f_1(depth) implemented in `f1(...)`. An exponential profile is used:

- the parameter a controls how quickly root density decays with depth;
- the function is non-zero only within the root zone of thickness L_r;
- f_1 is normalised so that its integral over depth equals unity.

5.3 Sink term for Richards equation

The local root water uptake sink $S(h,z,t)$ is modelled as

$$S(h,z,t) = f_1(z)\, f_2(h(z,t))\, E_p(t), \tag{21}$$

where $E_p(t)$ is the potential transpiration or evapotranspiration at time t. The routine `RootUptake-Model(...)`:

- evaluates f_1 at all grid nodes based on depth and root zone parameters;
- computes f_2 at each node from the current pressure head $h(z,t)$;
- scales the distribution so that the vertically integrated sink equals the actual transpiration rate used in the surface boundary condition.

6. INPUT DATA AND MODEL CONFIGURATION

6.1 Meteorological forcing

Meteorological data are read from an Excel file (e.g. `data_johnstown.xlsx`) with a sheet containing:

- date;
- daily rainfall depth (e.g. $\frac{\mathrm{mm}}{\mathrm{d}}$);
- potential evapotranspiration or its components (crop transpiration, soil evaporation).

These quantities are converted to consistent units (typically $\frac{\mathrm{m}}{\mathrm{d}}$) inside the driver script. The array of meteorological time points is stored as `time_given` in the `TimeSpec` structure and used for step-wise interpolation to the current model time.

6.2 Soil hydraulic parameters

Soil hydraulic parameters for each horizon are provided in a separate Excel sheet (e.g. `depthAvgV-GRosetta`) that includes:

- horizon identifier and upper/lower depth bounds;
- van Genuchten parameters θ_r, θ_s, α, N;
- saturated hydraulic conductivity K_{sat}.

The routine `assign_vg(depth, df, extend_to)` maps these horizon-based values to each numerical node. If the numerical domain extends deeper than the deepest horizon, the properties of the deepest horizon are extended downwards.

6.3 Initial conditions

Initial conditions for pressure head are constructed from observed soil tension data at discrete depths:

- measured tensions (e.g. in hPa) are converted to pressure head h in metres;
- a vertical interpolation is used to obtain $h(z, t = 0)$ at each grid node;
- this field provides the initial guess for the first time step of the Richards solver.

6.4 Data classes for configuration

The main configuration structures are:

- **`ProfileGridSpec`**: stores $z_{\min}$, $z_{\max}$, dz, arrays of node locations z, cell thicknesses dz_all, and depths;
- **`TimeSpec`**: defines $t_{\min}$, $t_{\max}$, Δt limits (`dt_min`, `dt_max`), initial time step, and time-stepping control factors;
- **`RWUSpec`**: holds root uptake parameters (ψ_a, ψ_d, ψ_w, L_r, and shape parameter a for the root distribution).

7. MODEL OUTPUTS AND POST-PROCESSING

The main outputs of the Richards solver and subsequent post-processing are:

- time series of $h(z, t)$, $\theta(z, t)$, $K(z, t)$ and effective saturation $S_e(z, t)$ on the numerical grid;
- root water uptake profiles $S(h, z, t)$ and time series of actual transpiration $T_a(t)$;
- time series of actual soil evaporation $E_a(t)$ and surface fluxes (rainfall, infiltration, ponded outflow);
- vertically integrated water storage in specified layers (e.g. 0–10 cm and 0–1 m);
- simulated soil water tension and soil moisture at depths corresponding to measurement locations for model evaluation.

The functions `spatial_interpolation(...)` and `prcoessed_results(...)` provide smoothed spatial profiles and aggregated metrics that are convenient for plotting and comparison with observations. The driver script produces figures showing:

- simulated vs. observed soil water tension at selected depths;
- simulated vs. observed volumetric water content at selected depths;
- temporal evolution of S_e or θ at several depths across the root zone.

References

[1] L. A. Richards, "Capillary conduction of liquids through porous mediums," *Physics*, vol. 1, no. 5, pp. 318–333, 11 1931. [Online]. Available: https://doi.org/10.1063/1.1745010

[2] M. A. Celia, E. T. Bouloutas, and R. L. Zarba, "A general mass-conservative numerical solution for the unsaturated flow equation," *Water Resources Research*, vol. 26, no. 7, pp. 1483–1496, 1990.