Implementing the Water, HEat and Transport model in GEOframe WHETGEO-1D v.1.0: algorithms, informatics, design patterns, open science features, and 1D deployment.

Niccolò Tubini[1] **and Riccardo Rigon**[2]

[1]Department of Civil, Environmental and Mechanical Engineering, University of Trento, Via Mesiano 77, 38123 Trento, Italy
[1]Center Agriculture Food Environment, University of Trento, Via Mesiano 77, 38123 Trento, Italy

Correspondence: Niccolò Tubini (niccolo.tubini@unitn.it)

Abstract. This paper presents WHETGEO and its 1D deployment, a new, physically based model simulating the water and energy budgets in a soil column. The purpose of this contribution is twofold. First, we discuss the mathematical and numerical issues involved in solving the Richardson-Richards equation, conventionally known as Richards' equation, and the heat equation in heterogeneous soils. In particular, for the Richardson-Richards equation (R^2) we take advantage of the nested Newton-Casulli-Zanolli (NCZ) algorithm that ensures the convergence of the numerical solution in any condition. Second, starting from numerical and modelling needs, we present the design of a software that is intended to be the first building block of a new, customisable, land-surface model that is integrated with process-based hydrology. WHETGEO is developed as an open-source code, adopting the Object-Oriented paradigm and a generic programming approach in order to improve its usability and expandability. WHETGEO is fully integrated in the GEOframe/OMS3 system allowing the use of the many ancillary tools it provides. Finally the paper presents the 1D deployment of WHETGEO, WHETGEO-1D, which has been tested against the available analytical solutions presented in Appendix.

Copyright statement. © Author(s) 2020. This work is distributed under the Creative Commons Attribution 4.0 License.

1 Introduction

The Earth's Critical Zone (CZ) is defined as the heterogeneous, near surface environment in which complex interactions involving rock, soil, water, air, and living organism regulate the natural habitat and determine the availability of life-sustaining resources (National Research Council, 2001). Clear interest in studying the CZ is spurred on by ever-increasing pressure due to the growth in human population and climatic changes. Central to simulating the processes in the CZ is the study of soil moisture dynamics (Clark et al., 2015a; Tubini et al., 2020). In the following we suggest that studying the CZ requires tools that are not yet readily available to researchers; then we propose one of our own. These tools should be flexible enough to allow the quick embedding of advancements in science.

1.1 Setting Up the Water Budget

It is generally accepted in the field that the motion of water in soil is regulated by a Darcian-Stokesian flow, meaning that any force is immediately dissipated and water under a gradient of generalised forces acquires a velocity (the Darcy velocity) which is linearly proportional to the gradient of the generalised force. This law is known as the Darcy-Buckingham law and reads

$$\boldsymbol{J} = K(\theta)\boldsymbol{\nabla}\left(\psi + z\right) \tag{1}$$

where the forces acting are gravity (z [L]) and the matric potential ψ [L], and where: J [$\mathrm{L\,T^{-1}}$] is the Darcian flux; K [$\mathrm{L\,T^{-1}}$] is the hydraulic conductivity; θ [$\mathrm{L^3\,L^{-3}}$] is the dimensioneless volumetric water content; $\boldsymbol{\nabla}$ [$\mathrm{L^{-1}}$] is the gradient operator; and z [L] is the vertical coordinate, pos-

itive upward. The assumptions under which such a law is derived from Newton's law are presented in (Whitaker, 1986; Di Nucci, 2014). The hydraulic conductivity depends on soil type (texture and structure) and water content, while the thermodynamic forces must be understood as gradients of the chemical potential of water, which, in turn, can be split as matric potential, osmotic potential and other potentials (Nobel et al., 1999, Chapter 2). However, in Eq. (1) we consider only the action of the matric potential. On the basis of the law of motion in Eq. (1), the mass conservation reads:

$$\frac{\partial \theta}{\partial t} = \nabla \cdot (K(\theta) \boldsymbol{\nabla} (\psi + z)) \tag{2}$$

where $\nabla\cdot$ $[\mathrm{L}^{-1}]$ is the divergence operator. Equation (2) is usually known as the Richards' equation (Richards, 1931), but was previously formulated by Richardson (1922). Therefore, in the following we call it the R^2 equation to remind of this double origin. There are very informative reviews that cover its general, historical and numerical aspects, such as: (Paniconi and Putti, 2015; Farthing and Ogden, 2017; Zha et al., 2019). Therefore, it is not deemed necessary to further summarize the matter here. The R^2 equation is a function of two variables, θ and ψ, and its resolution requires another relation between these two quantities. This relation is known as soil-water retention curves (SWRC), written in the plural because we have many SWRC depending on soil characteristics. The reader may be aware that the R^2 is an exact description of unsaturated flow only if we assume that soil is a bundle of capillaries and that largest capillaries drain first and are filled last (Mualem, 1976). In fact, in this case a relation can be obtained between the radius of the capillaries and the suction, which was fully derived (Kosugi, 1999). However, there are various reasons to take the capillary bundle concept as a rough approximation of natural soil. To enumerate some of the issues:

1. Firstly, pores in soils are not bundles of well defined capillaries of a single diameter; in fact, they can have quite random structures, as revealed, for instance, by tomography (Yang et al., 2018).

2. Secondly, logic and pore scale simulations, as in Tomin and Lunati (2016) for example, indicate that fluids fill the cavities where they fall, and only eventually are they redistributed according to the microstructure of the soil; that is to say, fluids do not move instantaneously from the largest pores to the smallest ones.

3. A set of relatively large pores can, in certain conditions, preferentially drive the flow of water in a short time scale according to laminar viscous flow driven by gravity, before any redistribution happens (Germann and Beven, 1981).

4. The role of living matter, such as bacteria, animals, fungi, vegetation, and roots, is usually eliminated from the hydrological picture but it should have a relevant place (Benard et al., 2019).

Besides,

5. Capillary forces are not the only ones acting at the microscale (Lu, 2016). In fact, measured suction values are far below pressures that can be sustained by capillarity alone.

6. Temperature affects water viscosity; infiltration is faster at warm temperatures and slower at cold ones (Constantz and Murphy, 1991).

7. In high-latitude and high-elevation environments, soils may be subject to freezing and thawing processes which affect pore volume and water dynamics (Dall'Amico et al., 2011).

These facts certainly do not threaten the nature of mass conservation in Eq. (2). However, they can certainly alter the statistics which generate the closure equations, i.e. the SWRC we currently use.

– Requirement I - Without entering into further details, we can observe that the aforementioned issues have consequences that would require a new software to include the possibility of adopting new parameterizations of SWRC and hydraulic conductivity quickly, easily and neatly.

1.2 The Three or Four Worlds

The flow of water obeys the general laws of physics for conservation of mass and momentum but, since the seminal works of Freeze and Harlan in 1969, the scientific community has split up (Furman, 2008) into three groups: groundwater people, vadose zone scientists, and surface water hydrologists. This compartmentalisation of the scientific community was fostered to deepen the knowledge within single branches, with the interactions between the different parts have been governed in models by assigning boundary conditions (Furman, 2008). However, these boundary conditions are intrinsically inadequate and inappropriate in representing the physics of interactions between different domains whose interactions depend strictly on the state of the system. When these conditions are prescribed a priori (Furman, 2008), the proper dynamics of the CZ fluxes cannot be obtained. There is the need to overcome this situation and, therefore:

– Requirement II - the boundary conditions hard-wired into algorithm implementation should be removed in favour of a simultaneous treatment of the three compartments (surface waters, vadose zone and groundwater).

Fortunately, Šimůnek et al. (2012) found the way to smoothly extend Richards equation into the groundwater equation. This and other similar approaches are now used

in various codes, such as Hydrus, ParFlow (Ashby and Falgout, 1996; Jones and Woodward, 2001; Kollet and Maxwell, 2006), CATHY (Paniconi and Wood, 1993; Paniconi and Putti, 1994), and GEOtop 2.0 (Rigon et al., 2006a; Endrizzi et al., 2014). To extend the R^2 equation into the saturated domain it is necessary to include the contribution of groundwater storativity due matrix and fluid compressibility. The common approach is to write the R^2 equation as:

$$\frac{\partial \theta}{\partial t} + S_s \frac{\theta}{\theta_s} \frac{\partial \theta}{\partial t} = \nabla \cdot (K(\psi) \boldsymbol{\nabla} (\psi + z)) \tag{3}$$

where S_s [L^{-1}] is the specific storage coefficient, defined as

$$S_s = \rho g (n\beta + \alpha), \tag{4}$$

with ρ [$\mathrm{M\,L}^{-3}$] being the water density, g [$\mathrm{L\,T}^{-2}$] gravitational acceleration, n [$\mathrm{L}^3\,\mathrm{L}^{-3}$] the soil porosity, β [$\mathrm{L\,T}^2\,\mathrm{M}^{-1}$] the liquid compressibility, and α [$\mathrm{L\,T}^2\,\mathrm{M}^{-1}$] the matrix compressibility. In the left-hand-side of Eq. (3), the first term accounts for changes in liquid saturation, while the second term accounts for the compression or expansion of the porous medium and the water. The left-hand-side term in Eq. (3) can be rewritten as

$$\left(c + S_s \frac{\theta}{\theta_s} \right) \frac{\partial \psi}{\partial t} \tag{5}$$

where c [L^{-1}] is the water retention capacity. Comparing the two terms in brackets, we can see that for $\psi < 0$, then $c >> S_s \frac{\theta}{\theta_s}$; this means that under unsaturated conditions, the contribution of the specific storage is negligible. Whereas when the soil is saturated and $\psi > 0$, then $c = 0$ and therefore what counts is the specific storage. Because of this, it is possible to account for groundwater specific storage simply by modifying the SWRC as:

$$\theta(\psi) = \begin{cases} \theta(\psi) & \text{if } \psi < 0 \\ \theta_s + S_s \psi & \text{if } \psi > 0 \end{cases} \tag{6}$$

Furthermore, switching from Richards to shallow water was made possible in the equation writing thanks to, for example, (Casulli, 2017; Gugole et al., 2018). Therefore, switching to a fully integrated, simultaneous treatment of the three domains is now possible.

1.3 The Necessary Coupling with the Energy Budget

As remarked in point 6 above, temperature affects water viscosity, which effectively doubles in passing from 5 to 20 [°C] (Eisenberg et al., 2005), with a positive feedback on the infiltration process. This has been clearly observed in natural systems (Ronan et al., 1998; Eisenberg et al., 2005; Engeler et al., 2011) where infiltration rates follow diurnal and seasonal temperature-cycles. In fact, according to Muskat and Meres (1936), the unsaturated hydraulic conductivity can be expressed as

$$K(\theta) = \kappa_r(\theta)\, \kappa \frac{\rho g}{\nu} \tag{7}$$

where $\kappa_r(\theta)$ [$-$] is the relative permeability, κ [L^2] is the intrinsic permeability, ρ [$\mathrm{L}^3\,\mathrm{M}^{-1}$] is the liquid density, g is the acceleration of gravity, and ν [$\mathrm{L}^2\,\mathrm{T}^{-1}$] is the kinematic viscosity of the liquid. Thus, for constant θ, variations in $K(\theta)$ due to temperature can be accounted as (Constantz and Murphy, 1991):

$$K(\theta, T_2) = K(\theta, T_1) \frac{\nu(T_1)}{\nu(T_2)} \tag{8}$$

In Eq. (8), T_1 is a reference temperature, while T_2 is the soil water temperature. Temperature is also responsible for the phase change of water, point 7, and because of pore ice, as well as excess ice, infiltration rates and subsurface flows are significantly modified Walvoord et al. (2012).

- Requirement III - To account for thermodynamic effects, temperature should be at least present in the R^2 equation as a parameter, as in Eq. (8). However, for a more accurate approximation of the water dynamics, the option to solve the water and energy budgets simultaneously must be present.

Soil thermal properties are important physical parameters in modelling land surface processes (Dai et al., 2019) since they control the partitioning of energy at the soil surface and its redistribution within the soil (Ochsner et al., 2001). For a multi-phase material, like soil, their definition is always problematic since they depend on the physical properties of each phase and their variations (Dong et al., 2015; Dai et al., 2019; Nicolsky and Romanovsky, 2018). In literature different models have been proposed with such a scope, and further studies on it are recommended (Dai et al., 2019): nonetheless, when considering the phase change of water, the estimation of the unfrozen and frozen water fraction is still an unresolved issue for which different models, usually referred to as SFCC, Soil Freezing Characteristics Curve, have been proposed (Kurylyk and Watanabe, 2013). Thus, it is clear that the aspects related to the estimation of soil thermal properties fall fully within Requirement I too. Moreover, there are other reasons for which the equations of the water and energy budgets should be solved in a coupled manner (Rigon et al., 2006a): for instance, this makes it possible to include an appropriate treatment of evaporation and transpiration processes (Bonan, 2019; Bisht and Riley, 2019), as well as of the heat advection (Frampton et al., 2013; Walvoord and Kurylyk, 2016; Wierenga et al., 1970; Ronan et al., 1998; Engeler et al., 2011; Zhang et al., 2019).

Finally, there is a great urge to model solutes transport according to the water movements. The range of applications for solute/tracer/pollutants spans from agriculture to industry to research itself. In fact, in recent years there has been a

tumultuous growth of studies using tracers to asses the various pathways of water (Hrachowitz et al., 2016). However, so far these studies have mostly used lumped models with limited capability to investigate water age selection processes, processes that became very important in the most recent literature e.g. (Penna et al., 2018). Using more complex modelling can benefit both the investigation of the processes and the construction of more refined water budget closures. Even though in this paper we do not detail the work on tracers, they must be kept in mind in software design so that the modules to be implemented eventually.

1.4 Heat Transport

Under the conditions defined above, the governing equation for the transport of energy in variably saturated porous media is given by the following energy conservation equation:

$$\frac{\partial h(\psi,T)}{\partial t} = \nabla \cdot [\lambda(\psi)\nabla T - \rho_w c_w (T - T_{ref}) J_w] \tag{9}$$

where h is the specific enthalpy of the medium [$\mathrm{L}^2\,\mathrm{T}^{-2}$], λ [$\mathrm{M\,L\,\Theta}^{-1}\,\mathrm{T}^{-3}$] is the thermal conductivity of the soil, T [Θ] is the temperature, ρ_w [$\mathrm{M\,L}^{-3}$] is the water density, c_w [$\mathrm{L}^2\,\Theta^{-1}\,\mathrm{T}^{-2}$] is the specific heat capacity of water, T_{ref} [Θ] is a reference temperature used to define the enthalpy, and J_w is the water flux [$\mathrm{L\,T}^{-1}$]. The first term on the right-hand-side is the heat conduction flux, described by the Fourier's law, and the second term is the sensible heat advection of liquid water. The specific enthalpy of a control volume of soil V_c [L^3] can be calculated as the sum of the enthalpy of the soil particles and liquid water:

$$h = \rho_{sp} c_{sp} (1 - \theta_s)(T - T_{ref}) + \rho_w c_w \theta(\psi)(T - T_{ref}) \tag{10}$$

where ρ_{sp} and ρ_w are the densities of the soil particles and the water, c_{sp} and c_w are the specific heat capacities of the soil particles and the water. Equation (9) is the so-called conservative form.

1.5 Informatics

Amateurism in software (model) building may reflect in an incorrect analysis of the hydrological processes, threatening the development of science from the offset. Besides, in recent years a new paradigm in evaluating scientific productivity, and therefore software productivity, has gained in importance (e.g. see project (IDEAS)), with the aim of reducing effort, time and cost of software development, maintenance and support. More in general, the scientific community has recognised the need to transition to open, accessible, reusable, and reproducible research, which requires the adoption of new software engineering practices. Our intention, therefore, is to build robust and reliable tools that are without unknown side effects and designed to be easily inspected by third parties.

As discussed in Serafin (2019), computing programs have been usually developed as monolithic code; this has serious drawbacks for maintainability and development, as has been shown by our own experience with the GEOtop model (Rigon et al., 2006a; Endrizzi et al., 2014), and by experiences with other modelling frameworks (Clark et al., 2015b, 2021; Bisht and Riley, 2019). The main limit of the monolithic approach can be identified in its absence of *separation of concerns* (Martin, 2009; Newman, 2015). This, results in huge, unreadable code bases that mix several different scientific/mathematical concepts (David et al., 2013), characteristics that impede an easy reviews of models. Being aware of these past experiences, therefore the WHETGEO-1D code has been developed by adopting Object-Oriented-Programming (OOP) and a generic programming strategy whose contents are explained below. In this manner it is possible to split up the code into smaller structures each one implementing a specific scientific/mathematical concept and, besides, to decouple the algorithms from concrete data representation with gains in code maintainability and algorithms inspection.

Moreover, WHETGEO-1D has been integrated into the Object Modelling System v3 (OMS3) framework (David et al., 2013), see Appendix B. OMS3 is a component-based environmental modelling framework that allows the developers to create a component for each single modelling concept, thus having a second level of separation of concerns. Because of the modularity of OMS3, the WHETGEO-1D components can be used seamlessly at run time by connecting them with the OMS3 DSL language based on Groovy (https://groovy-lang.org). OMS3 provides the basic services and, among these, tools for calibration and implicit parallelization of component runs.

WHETGEO is also part of GEOframe (Formetta et al., 2014a; Bancheri, 2017; Bancheri et al., 2020), a system of OMS3 components, see Appendix A. GEOframe provides various components for precipitation treatment, radiation estimation in complex terrain, and evaporation and transpiration. Conversely, with WHETGEO-1D the GEOframe system is enriched with new components, now able to cope with different spatial resolutions and configurations compared to what was offered by the existing tools (Bancheri et al., 2020).

1.6 Organisation and Scope

This paper describes the implementation and content of the WHETGEO-1D (Water, HEat and Transport in GEOframe) software, in observance of requirements I to III and aware of the hydrologic facts described in points 1 to 7 of Section 1.1. Actually, we do not seek to address all the issues listed but focus instead on getting the proper software design that their implementation requires. The first objective is to not have to break and rewrite existing codes and, then, being able to code incrementally, in step with the progressing research in hydrological processes.

Further requirements derive from Numerics and Mathematics. In fact, translating the R^2 and the heat equations into discretized numerical codes implies overcoming some challenges that are well known in the Hydrology community, as illustrated in the next sections. Our contribution aims to make available to the public for the first time integration algorithms that ensure the conservation of mass and energy for any time step size and for a great variety of conditions. It is interesting to note that some of these algorithm were presented almost a decade ago in the Math community, remaining a little concealed, however. Although WHETGEO is meant to deal with the transport of solutes and tracers, this topic will be covered in a dedicated paper since its physics and numerics are slightly different from those of the water and energy budgets.

Moreover, WHETGEO-1D represents the first brick of a new expandable land surface model (Bisht and Riley, 2019): the forthcoming development of WHETGEO-1D its inclusion in the GEO-SPACE (Soil Plant Atmosphere Continuum Estimator) model that aims to simulate the soil-plant continuum (D'Amato et al., 2022).

In the following section we discuss the mathematical issues and the numerics of our implementation. In the subsequent one, we describe the informatics and software engineering issues, and how we solved them. Finally, we discuss WHETGEO-1D implementation by testing it against some analytical solutions available for simplified parameterizations and boundary conditions, for details of these see Appendix C.

2 Mathematical Issues and Numerics

Translating these equations into numerical discretized codes implies to overcome some challenges, as we illustrate in the next sections by exploring, at first, the issues posed by each one of the equation.

2.1 General Issues of the R^2 Equation

Equation (2) is said to be written in "mixed form" because it is expressed in term of θ and ψ (and uses the SWRC to connect the two variables).

The "ψ-based form" is derived from Eq. (2) by applying the chain rule for derivatives:

$$c(\psi)\frac{\partial \psi}{\partial t} = \nabla \cdot [K(\psi)\nabla(\psi + z)] \tag{11}$$

where

$$c(\psi) = \frac{\partial \theta(\psi)}{\partial \psi} \tag{12}$$

with dimension $[L^{-1}]$, is the specific moisture capacity, also called hydraulic capacity. Even though Eq. (2) and Eq. (11) are analytically equivalent under the assumption that the water content is a differentiable variable, this is not generally true in the discrete domain where the derivative chain rule is not always valid (Farthing and Ogden, 2017). Because of this the ψ-based form may suffer from large balance errors in the presence of big nonlinearities and strong gradients, as discussed in (Casulli and Zanolli, 2010; Farthing and Ogden, 2017; Zha et al., 2019) and literature therein. The specific moisture capacity c, which appears in the storage term, itself depends on ψ and so is not constant over a discrete time interval during which ψ changes value. Let us discretise the time derivative in Eq. (11) by using the backward Euler scheme and obtain:

$$\tilde{c}_i \frac{\psi_i^{n+1} - \psi_i^n}{\Delta t} \tag{13}$$

where $\tilde{c}_i$ is the discrete operator of the soil moisture capacity, $c(\psi)$. In order to preserve the chain rule of derivatives at the discrete level, i.e. the equality $c\partial\psi/\partial t = \partial\theta(\psi)/\partial t$, $\tilde{c}_i$ has to satisfy the requirement (Roe, 1981)

$$\tilde{c}_i(\psi_i^{n+1} - \psi_i^n) = \theta(\psi_i^{n+1}) - \theta(\psi_i^n) \tag{14}$$

As can be seen from the above equation, the right definition of $\tilde{c}_i$ depends on the solution itself. To overcome this problem, in literature different techniques have been presented to improve the evaluation of $\tilde{c}_i$, but none ensures mass conservation (Farthing and Ogden, 2017).

There is a third form of Eq. (2), the so-called "θ-based form", that reads as

$$\frac{\partial \theta}{\partial t} = \nabla \cdot [D(\theta)\nabla\theta + K(\theta)] \tag{15}$$

where all is made explicit by inverting the SWRC and $D(\theta) := K(\theta)\partial\psi/\partial\theta$ is the soil-water diffusivity $[L^2\,T^{-1}]$. The first term on the right-hand-side represents the water flow due to capillary forces, while the second term is the contribution due to gravity (Farthing and Ogden, 2017). The θ-based form is mass-conserving and it can be solved perfectly by mass conservative methods (Casulli and Zanolli, 2010). However, it applicability is limited to the unsaturated zone since water content varies between θ_r and θ_s, whereas water suction is not bounded. This formulation is intrinsically not suited to fulfilling our requirement III. Moreover, water content is discontinuous across layered soil since the SWRCs are soil specific, whereas water suction is continuous even in inhomogeneous soils (Farthing and Ogden, 2017; Bonan, 2019).

In WHETGEO, we directly use the conservative form of the R^2 equation, Eq. (2), which seems the easiest way to deal with both the mass conservation issues and the extension of the equation to the saturated case.

2.1.1 The Discretization of the R^2 Equation

The implicit finite volume discretization of Eq. (2) reads as:

$$\theta_i(\psi_i^{n+1}) = \theta_i(\psi_i^n) + \Delta t \Big[K_{i+\frac{1}{2}}^{n+1} \frac{\psi_{i+1}^{n+1} - \psi_i^{n+1}}{\Delta z_{i+\frac{1}{2}}} + K_{i+\frac{1}{2}}^{n+1} - K_{i-\frac{1}{2}}^{n+1} \frac{\psi_i^{n+1} - \psi_{i-1}^{n+1}}{\Delta z_{i-\frac{1}{2}}} - K_{i-\frac{1}{2}}^{n+1} + S_i^n \Big] \quad (16)$$

where Δt is the time step size,

$$S_i = \int_{\Omega_i} S \, d\Omega \quad (17)$$

is an optional source/sink term in volume, and $\theta_i(\psi)$ is the ith water volume given by

$$\theta_i(\psi) = \int_{\Omega_i} \theta(\psi) \, d\Omega. \quad (18)$$

Equation (16) can be written in matrix form as

$$\boldsymbol{\theta}(\boldsymbol{\psi}) + \mathbf{T}\boldsymbol{\psi} = \boldsymbol{b} \quad (19)$$

where $\boldsymbol{\psi} = \{\psi_i\}$ is the tuple of unknowns, $\boldsymbol{\theta}(\boldsymbol{\psi}) = \theta_i(\psi_i)$ is a tuple-function representing the discrete water volume, $\mathbf{T}$ is the flux matrix, and $\boldsymbol{b}$ is the right-hand-side vector of Eq. (16), which is properly augmented by the known Dirichlet boundary condition when necessary. For a given initial condition ψ_i^0, at any time step $n = 1, 2, \ldots$, Eq. (16) constitutes a nonlinear system for ψ_i^{n+1}, with the nonlinearity affecting only the diagonal of the system and being represented by the water volume $\theta_i(\psi_i^{n+1})$. This set of equations is a consistent and conservative discretization of Eq. (2). Therefore, regardless of the chosen spatial and temporal resolution, ψ_i^{n+1} is a conservative approximation of the new water suction.

2.1.2 Surface Boundary Condition

The definition of the type of surface boundary condition (Neumann vs. Dirchlet) for the R^2 equation is a non trivial task since it can depend on the state of the system. In literature several approaches are used (Furman, 2008). These approaches are mainly based on a switch of the type of the boundary condition from a prescribed head to prescribed flux and viceversa. This switching often causes numerical difficulties that need to be addressed (Furman, 2008).

To overcome this problem we have included an additional computational node at the soil surface. As will be made clear in the following, for it we prescribe an "equation state" like that presented in (Casulli, 2009):

$$H(\psi) = \begin{cases} \psi & \text{if } \psi > 0 \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

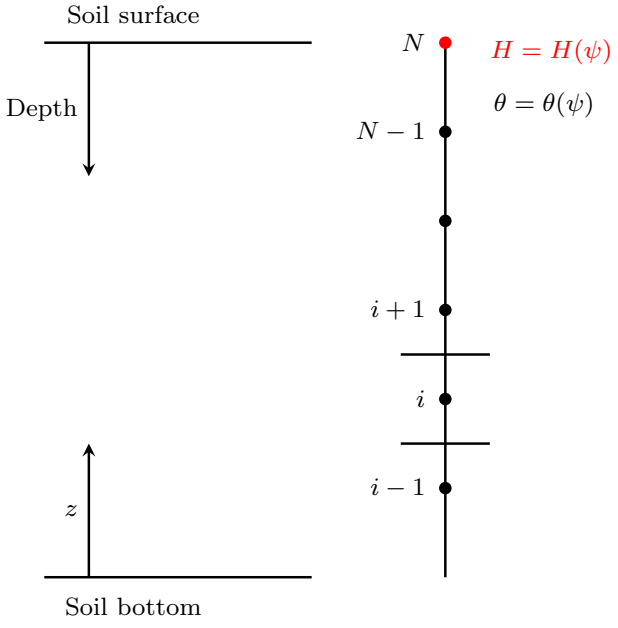

Figure 1. Scheme of the computational domain to solve the R^2 equation in 1D. The uppermost node represents the water depth at the soil surface. By considering this additional computational node the boundary condition does not change its nature depending on the solution.

where H [L] is the water depth, which also represents the pressure if the ponding is assumed to happen in hydrostatic conditions. By doing so, it is possible to prescribe as the surface boundary condition the rainfall intensity (Neumann type) without resorting to any switching techniques to reproduce infiltration excess or saturation excess processes. In this case the system, Eq. (19), must to be modified to account for the additional computational node describing the state of the soil surface:

$$\begin{cases} H_N(\psi_N^{n+1}) - \Delta t \left[0 - K_{N-\frac{1}{2}}^n \dfrac{\psi_N^{n+1} - \psi_{N-1}^{n+1}}{\Delta z_{N-\frac{1}{2}}} \right] \\ \quad = H_N(\psi_N^n) + \Delta t [J^n - K_{N-\frac{1}{2}}^n] \quad \text{if } i == N \\ \theta_i(\psi_i^{n+1}) - \Delta t \left[K_{i+\frac{1}{2}}^n \dfrac{\psi_{i+1}^{n+1} - \psi_i^{n+1}}{\Delta z_{i+\frac{1}{2}}} - K_{i-\frac{1}{2}}^n \dfrac{\psi_i^{n+1} - \psi_{i-1}^{n+1}}{\Delta z_{i-\frac{1}{2}}} \right] \\ \quad = \theta_i(\psi_i^n) + \Delta t [K_{i+\frac{1}{2}}^n - K_{i-\frac{1}{2}}^n] \quad \text{if } i = 1, 2, ..., N-1 \end{cases} \quad (21)$$

where J^n is the rainfall intensity and represents the Neumann boundary condition used to drive the system at the soil surface. For any time step, Eq. (21) can be written in matrix form, similar to (Eq. (19)), as:

$$\boldsymbol{V}(\boldsymbol{\psi}) + \mathbf{T}\boldsymbol{\psi} = \boldsymbol{b} \quad (22)$$

where $\boldsymbol{\psi} = \{\psi_i\}$ is the tuple of unknowns, $\boldsymbol{V}(\boldsymbol{\psi}) = (\theta_i(\psi_i))$ for $i = 1, 2, ..., N-1$ and $V_N(\psi) = H(\psi)$ is a tuple-function

representing the discrete water volume, $\mathbf{T}$ is the flux matrix, and $\boldsymbol{b}$ is the right-hand-side vector of Eq. (16), which is properly augmented by the known Dirichlet boundary condition when necessary. For a given initial condition ψ_i^0, at any time step $n = 1, 2, ...$ Eq. (22) constitutes a nonlinear system for ψ_i^{n+1}, with the nonlinearity affecting only the diagonal of the system and being represented by the water volume $V_i(\psi_i^{n+1})$. Therefore, regardless of the chosen spatial and temporal resolution, ψ_i^{n+1} is a conservative approximation of the new water suction.

2.2 Heat Transport Numerics Issues

Equation (9) is said to be written in conservative form and expresses an important property, which is the conservation of the scalar quantity, in this case the specific enthalpy. It is interesting to note that by making use of the mass conservation equation, Eq. (2), Eq. (9) can be written in an analytically equivalent form, the so called non-conservative form (Sophocleous, 1979; Šimůnek et al., 2005):

$$c_T \frac{\partial T}{\partial t} = \lambda \nabla^2 T - \rho_w c_w J_w \nabla T \tag{23}$$

Equation (23) expresses another important property which is the maximum principle (Casulli and Zanolli, 2005), i.e. the analytical solution is always bounded, above and below, by the maximum and minimum of its initial and boundary values, as shown in (Greenspan and Casulli, 1988, Chapter 7.3).

Although Eq. (9) and Eq. (23) are analytically equivalent, once they are discretised the corresponding numerical solution will, in general, either be conservative or satisfy a discrete max-min property (Casulli and Zanolli, 2005), but not both as would be required.

As in the case of the water flow, thermal budget issues can be subdivided into three aspects: the discretization of the equation, the inclusion of the appropriate boundary conditions, and the implementation of some closure equation for the thermal capacity and conductivity.

2.2.1 The Discretization of the Heat Equation

The key feature (Casulli and Zanolli, 2005) to obtaining a numerical solution for the heat transport equation that is both conservative and possesses the max-min property is to solve the conservative form of the heat equation by using the velocity field obtained in solving the continuity equation Eq. (2). By making use of the up-wind scheme for the advection part and the centered difference scheme for the diffusion part we have:

$$\begin{aligned} C_{T_i}^{n+1} T_i^{n+1} = C_{T_i}^n T_i^n - \rho_w c_w \Delta t \Bigg[& \frac{1}{2} J_{1+\frac{1}{2}}^{n+1} \left(T_{i+1}^{n+1} + T_i^{n+1} \right) - \\ & \frac{1}{2} \left| J_{1+\frac{1}{2}}^{n+1} \right| \left(T_{i+1}^{n+1} + T_i^{n+1} \right) \\ & - \frac{1}{2} J_{1-\frac{1}{2}}^{n+1} \left(T_i^{n+1} + T_{i-1}^{n+1} \right) + \frac{1}{2} \left| J_{1-\frac{1}{2}}^{n+1} \right| \left(T_{i+1}^{n+1} + T_i^{n+1} \right) \Bigg] \\ & + \Delta t \left[\lambda_{i+\frac{1}{2}}^n \frac{T_{i+1}^{n+1} - T_i^{n+1}}{\Delta z} - \lambda_{i-\frac{1}{2}}^n \frac{T_i^{n+1} - T_{i-1}^{n+1}}{\Delta z} \right] \end{aligned} \tag{24}$$

where

$$C_{T_i} = \int_{\Omega_i} \rho_{sp} c_{sp} (1 - \theta_s) + \rho_w c_w \theta(\psi) \, d\Omega \tag{25}$$

When heat equation does not consider water phase changes, it is decoupled from the R^2 equation and the finite volume discretisation leads to a linear algebraic system of equations. However, once freezing and thawing processes are considered, the heat equation is fully coupled with the R^2 equation, as in (Dall'Amico et al., 2011) for instance, and the enthalpy function becomes nonlinear. At this point, since the enthalpy function is nonlinear the NCZ algorithm is required to linearise it, as shown in (Tubini et al., 2020). So far, we have not considered the problem of water flow in freezing soils, however being aware of this issue is important for the future developments and code design.

2.2.2 Driving the Heat Equation with the Surface Energy Budget

At the soil surface the heat equation is driven by the surface energy balance. The heat flux exchanged between the soil and the atmosphere, the surface heat flux, $\mathcal{F}$ [$\mathrm{M\,T^{-3}}$], is given as:

$$\mathcal{F} = S_{in} - S_{out} + L_{in} - L_{out} + H - LE \tag{26}$$

where S_{in} is the incoming short wave radiation, S_{out} is the outgoing short wave radiation, L_{in} is the incoming longwave radiation, L_{out} is the outgoing longwave radiation, and H and LE are respectively the turbulent fluxes of sensible heat and latent heat. Fluxes are positive when directed toward the soil surface and all have the dimension of an energy per unit area per unit time [$\mathrm{M\,T^{-3}}$].

Similarly to the definition of the surface boundary condition for the water flow, the surface boundary condition for the energy equation is also system dependent. In fact, in (26) the only fluxes that do not depend on the soil temperature and/or moisture are the incoming shortwave and longwave radiation fluxes, S_{in} and L_{in}. The outgoing shortwave radiation flux is usually parameterized as:

$$S_{out} = \alpha S_{in} \tag{27}$$

where the surface albedo α $[-]$ can be assumed to vary with the soil moisture content (Saito et al., 2006) and radiation wavelength. The outgoing longwave surface radiation is:

$$L_{out} = (1-\epsilon)L_{in} + \epsilon\sigma T_s^4 \tag{28}$$

where T_s $[\Theta]$ the temperature of the topmost layer of soil, ϵ is the soil emissivity, and σ is the Stefan-Boltzmann constant. The sensible heat flux H is taken as:

$$H = \frac{\rho_a c_a}{r_H}(T_a - T_s) \tag{29}$$

where ρ_a is the air density $[\mathrm{M\,L^{-3}}]$, and c_a is the thermal capacity of air per unit mass $[\mathrm{L^2\,T^{-2}\,\Theta^{-1}}]$. Regarding the aerodynamic resistances r_H $[\mathrm{T\,L^{-1}}]$, it should be noted that it can be evaluated with different degrees of approximation and may require a specific modelling solution. For instance, the aerodynamic resistance r_H can be evaluated with models ranging from semi-empirical models to the the Monin-Obukov similarity (Liu et al., 2007), or even by solving the turbulent dynamics with direct methods (Raupach and Thom, 1981; Mcdonough, 2004).

The latent heat flux is taken here as given by a formula of the type:

$$LE = l\rho_a E_P \frac{r_H r_v}{r_H + r_v} \tag{30}$$

where l $[\mathrm{M\,L^2\,T^{-2}}]$ is the specific latent heat of vaporisation of water, E_P is the potential evapotranspiration, r_H and r_v $[\mathrm{T\,L^{-1}}]$ are respectively the aerodynamic resistance and the soil surface resistance to water vapour flow. The latent heat flux it is the sum of two distinct processes evaporation and transpiration. Compared to the other fluxes, latent heat flux presents further complications because evaporation is both an energy and a water limited process, and transpiration depends also on the physiology of trees (as well as root distribution/growth and leaf cover). The latent heat flux is associated to the water flux that must be accounted in the R^2 equation. Here we present a simplified treatment of the latent heat flux as an external driving force/ prescribed boundary condition. A more exhaustive and physically based treatment of the latent heat flux, and the related water flux, is addressed in the ongoing development of the GEO-SPACE model (D'Amato).

Including the surface energy budget boundary condition requires the computation of additional quantities such as the incoming radiation fluxes, the shortwave radiation and the longwave radiation, and the potential evapotranspiration flux. These quantities can be easily computed within the GEOframe system in which WHETGEO-1D is embedded. The proper estimation of the incoming radiation fluxes is far from being a simple task and it is often oversimplified in hydrological problems. Our approach is to use the tools already developed inside the system GEOframe which were tested independently and accurately (Formetta et al., 2013, 2014a). Similarly the evapotranspiration can be computed with other GEOframe components (Bottazzi, 2020; Bottazzi et al., 2021).

2.3 Algorithms

By using a numerical method, here the finite volume method, a partial differential equation is transformed into a system of nonlinear algebraic equations, as has already been shown. The system has to be solved with iterative methods and, at their core, these reduce the problem to using a linear systems solver. The solver can be of various types, according to the dimension of the problem. For instance in 1D, the final system of finite volume problems we present is tridiagonal and can be conveniently solved with the Thomas algorithm (Quarteroni et al., 2010), which is a fast direct method. In 2D or 3D, the final matrix is not tridiagonal and a different solution method must be used, for instance the conjugate gradient (Shewchuk et al., 1994). These algorithms are well known and do not need to be explained here.

However, the reduction of a nonlinear system to a linear one is not trivial. We illustrate the issues by taking the R^2 equation as an example. As discussed in depth in Zha et al. (2019) and Farthing and Ogden (2017), and references therein, the linearisation of the R^2 equation is challenging. Following the work of Celia et al. (1990), a lot of advancements have been made in this direction: Hydrus, CATHY, and ParFlow use variants of the Newton and Picard iteration methods (Zha et al., 2019; Paniconi and Putti, 1994), while GEOtop 2.0 implements a suitable globally convergent Newton method (Kelley, 2003). Although current algorithms are relatively stable, they may fail to converge or require a considerable computational cost (Zha et al., 2019). This has a significant impacts both on the reliability of the solution, which can have mass-balance errors, and on the computational cost to produce it (Farthing and Ogden, 2017; Zha et al., 2019). Since Casulli and Zanolli (2010) and Brugnano and Casulli (2008), a new method was found, called nested Newton by the authors and NCZ in the following, that guarantees convergence in any situation, even with the use of large time steps and grid sizes.

As clearly pointed out by Casulli and Zanolli (2010), what makes the linearisation of the R^2 equation difficult is the non-monothonic behaviour of the soil moisture capacity. A mathematical proof of convergence for NCZ exists (Brugnano and Casulli, 2008, 2009; Casulli and Zanolli, 2010, 2012), which is not repeated here. However, we take the time to illustrate this new algorithm with care.

Let us start again from the nonlinear system (Casulli and Zanolli, 2012):

$$\boldsymbol{V}(\boldsymbol{\psi}) + \mathbf{T}\boldsymbol{\psi} = \boldsymbol{b} \tag{20}$$

where $\boldsymbol{\psi} = (\psi_i)$ is the tuple of unknowns, $\boldsymbol{V}(\boldsymbol{\psi}) = (V_i(\psi_i))$ is a nonnegative vectorial function and where the $V_i(\psi_i)$ are defined for all $\psi_i \in \mathbb{R}$ and can be expressed as:

$$V_i(\psi_i) = \int_{-\infty}^{\psi_i} a_i(\xi)\, d\xi \tag{31}$$

For all $i = 1, 2, ..., N$, the following assumptions are made on the functions $a_i(\psi)$ (we are here quite literally following (Casulli and Zanolli, 2010)):

A1 : $a_i(\psi)$ is defined for all $\psi \in \mathbb{R}$ and is a nonnegative function with bounded variations;

A2 : There exists $\psi_i^* \in \mathbb{R}$ such that $a_i(\psi)$ is strictly positive and nondecreasing in $(-\infty, \psi_i^*)$ and nonincreasing in $(\psi_i^*, +\infty)$.

$\mathbf{T}$ in eq (Eq. (20)) is the so-called matrix flux and it is a symmetric and (at least) positive semidefinite matrix satisfying one of the following properties:

T1 : $\mathbf{T}$ is a Stieltjes matrix, i.e., a symmetric M-matrix, or

T2 : $\mathbf{T}$ is irreducible, null$(\mathbf{T}) \equiv$ span$(\boldsymbol{v})$, with $\boldsymbol{v} > \mathbf{0}$ (componentwise), and $\mathbf{T}+\mathbf{D}$ is a Stieltjes matrix for all diagonal matrices $\mathbf{D} \gneqq \mathrm{O}$, with O denoting the null matrix.

Finally $\boldsymbol{b}$ is the vector of the known terms. When $\mathbf{T}$ satisfies property T2, then for Eq. (20) to be physically and mathematically compatible, the following assumption about $\boldsymbol{b}$ is required:

$$0 < \boldsymbol{v}^\intercal \boldsymbol{b} < \boldsymbol{v}^\intercal \boldsymbol{V}^{Max} \tag{32}$$

where $\boldsymbol{V}^{Max} = \int_{-\infty}^{+\infty} a_i(\xi) d\xi$.

Having assumed that the $a_i(\psi)$ are non-negative functions of bounded variations, they are differentiable almost everywhere, admit only discontinuities of the first kind, and can be expressed as the difference of two non-negative, bounded, and non-decreasing functions, say $p_i(\psi)$ and $q_i(\psi)$, such that:

$$\begin{aligned} &a_i(\psi) = p_i(\psi) - q_i(\psi) \geq 0 \\ &0 \leq q(\psi) \leq p(\psi) \end{aligned} \tag{33}$$

for all $\psi \in \mathbb{R}$. When $a(\psi)$ satisfy assumptions A1 and A2, the corresponding decomposition (known as the Jordan decomposition (Chistyakov, 1997) and presented in Figure 2), is given by:

$$\begin{aligned} &p_i(\psi) = a_i(\psi) \quad q_i(\psi) = 0 \quad &&\textit{if } \psi \leq \psi_i^* \\ &p_i(\psi) = a_i(\psi_i^*) \quad q_i(\psi) = p_i(\psi) - a_i(\psi) \quad &&\textit{if } \psi > \psi_i^* \end{aligned} \tag{34}$$

where ψ_i^* is the position of the maximum of p_i. Thereafter, $\boldsymbol{V}(\boldsymbol{\psi})$ can be expressed as

$$\boldsymbol{V}(\boldsymbol{\psi}) = \boldsymbol{V}_1(\boldsymbol{\psi}) - \boldsymbol{V}_2(\boldsymbol{\psi}) \tag{35}$$

where the i-th component of $\boldsymbol{V}_1(\boldsymbol{\psi})$ and $\boldsymbol{V}_2(\boldsymbol{\psi})$ are defined as

$$V_{1,i}(\psi_i) = \int_{-\infty}^{\psi_i} p_i(\xi)\, d\xi \qquad V_{2,i}(\psi_i) = \int_{-\infty}^{\psi_i} q_i(\xi)\, d\xi \tag{36}$$

By making use of Eq. (35) the algebraic system in Eq. (20) can be written as

$$\boldsymbol{V}_1(\boldsymbol{\psi}) - \boldsymbol{V}_2(\boldsymbol{\psi}) + \mathbf{T}\boldsymbol{\psi} = \boldsymbol{b} \tag{37}$$

It is necessary here to point out exactly how the nonlinear system, Eq. (20), reads when considering only the R^2 equation and when the water depth function is used to properly define the surface boundary condition. In the first case, i.e. when Neumann or Dirichlet boundary conditions are used, the vectorial function is defined as $\boldsymbol{V}(\boldsymbol{\psi}) = (\theta_i(\psi_i))$ for $i = 1, 2, ..., N$.

Instead, when we consider the water depth function to describe the computational node at the soil surface, the vectorial function is defined as $\boldsymbol{V}(\boldsymbol{\psi}) = (\theta_i(\psi_i))$ for $i = 1, 2, ..., N-1$ and $V_N(\psi) = H(\psi)$. Therefore, the nonlinear system in Eq. (37) is valid to describe both the subsurface and surface waters when the symbols are appropriately understood.

This aspect, the use of two different equation states, and the fact that the NCZ algorithm can be successfully reused to solve other problems (Casulli and Zanolli, 2012; Tubini et al., 2020), requires a careful design of its implementation, as discussed in the following sections.

3 Design and deployment of WHETGEO-1D

The concepts and requirements previously illustrated must be cast into a software whose usability, expandability and inspectability are demanded by good software design, which adds further requirements. The software design requirements and the software deployment are explained below.

3.1 Design Requirements

One of the major difficulties encountered by a research group concerns the development and reuse of scientific software (Berti, 2000) and the writing of structurally clean code, i.e. a code that is easily readable and understandable, with objects that have a specified, and possibly unique, responsibility (Martin, 2009).

An Object-Oriented-Programming approach, with the adoption of standard design patterns (DPs) (Gamma et al., 1995; Freeman et al., 2008) and the creation of new ones, has been adopted for the internal classes design and hierarchy.

The design principles followed by the WHETGEO-1D software can be summarised as follows:

A. The software should be open source to allow inspection and improvements by third parties;

B. For the same reason it should be organised into parts, each with a clear functional meaning and possibly a single responsibility.

C. The software can be extended with minimal effort and without modifications (according to the "open to exten-

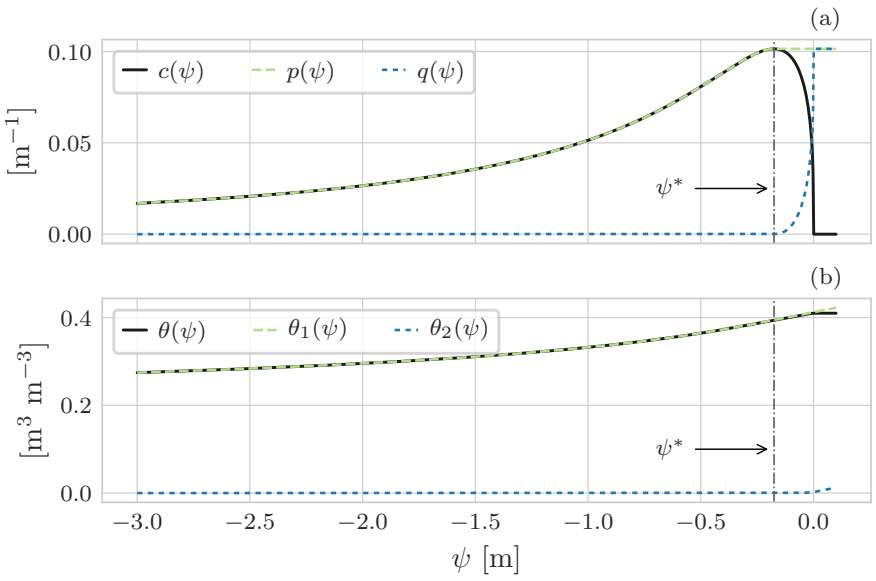

Figure 2. Graphical representation of the Jordan decomposition for soil water content using the SWRC model by Van Genuchten (Van Genuchten, 1980) for a clay loam soil (Bonan, 2019). (a) shows the Jordan decomposition of $c(\psi)$, (34). For $\psi = \psi^*$, $c(\psi)$ presents a maximum: for $\psi < \psi^*$ it is increasing, and for $\psi > \psi^*$ it is decreasing. This non monothonic behaviour causes problems when solving the nonlinear system. $c(\psi)$ is thus replaced by $p(\psi)$ (in green) and $q(\psi)$, in blue, two monothonic functions whose difference is the original function c. Consequently, (b), $\theta(\psi)$ is replaced by $\theta_1(\psi)$ and $\theta_2(\psi)$, Eq. (35).

sions, closed to modifications" principle Freeman et al. (2008)). In particular, the parts to be modified are those that, according to the discussion of the previous sections, could be changed to try new closures, i.e. the SWRC and the hydraulic conductivities in the case of the R^2 equation, and the thermal capacity and thermal conductivity in the case of the energy budget. Adding a new SWRC type or a new conductivity functions should be easy.

D. The largest set of boundary conditions should be smoothly manageable

E. The implementation of equations should be abstract, according to the principle of “programming to interfaces and not to concrete classes”, which is the core of contemporary OOP (Gamma et al., 1995). The different equations describing the processes should be implemented within the set of classes by implementing a common interface.

F. The implementation of algorithms should not depend on the data formats of inputs and outputs.

Another requirement has to do with the user experience. In fact, solvers of PDEs (Menard et al., 2020) tend to be complex to understand and run when features are added. In particular, the number of inputs grows exponentially when features are added, and the user has to overcome a steep learning curve before being able to use these software packages to appreciate all the cases implemented and their physics.

G. To simplify this situation, WHETGEO-1D has to be implemented is such a way that any of the alternative implementations must come only with their own parameters and variables, and appear to the user as simple as possible, though not too simple.

There is finally a last requirement to consider:

H. For computational and research purposes, there will be one,two and three dimensional (1D, 2D, 3D) implementations of the aforementioned equations. Therefore, as much as possible of the code should be shared across these. In particular, the NCZ and Newton algorithms should be shareable across the various applications.

This requirement implies that the geometry of the domain, as well as the topology, be specified in an abstract manner to cope with the specifics of each dimensionality.

The rest of this section is organised to respond to points from A to H. A is actually responded to in the next section describing where the software can be downloaded and with which open license. Points B and C are accomplished by studying an appropriate design of classes and the use of design patterns (Gamma et al., 1995; Freeman et al., 2008). For D, generic programming is used and specific classes are implemented. E is resolved by deploying a set of classes that implement a common interface with an extension that allows it to obtain the required functionalities.

To respond to the issues raised in F and G, WHETGEO-1D is implemented as various components of OMS3, as shown below, each one with its own inputs. Therefore, the number of flags to check and the number of unused inputs are reduced to the minimum required by the solvers and parameterizations that the user chooses. If users want to solve the R^2 equation alone, for instance, they can pick out the appropriate component and do not need to know about the inputs and details of the energy budget. The separation into components has two other advantages. First, it eases the testing of a single process against available analytical solutions and against other models results (Bisht and Riley, 2019). Second, it improves the model structure, facilitating the representation of new processes (Clark et al., 2021).

Point H is solved by deploying new components for the 1D, 2D, and 3D cases. In the following section we mainly deal with points B to E. Before discussing details of some classes, a few general choices have to be reported. Data of any type are stored internally in vectors of doubles, in turn encapsulated in appropriate Java objects. OOP good practice would suggest that an object should be immutable (Bloch, 2001), but we decided that the main classes have to be mutable and allocated once forever as singletons (Gamma et al., 1995; Freeman et al., 2008). This potentially exposes the software to side effects but frees it to allocate new objects at any time step and decreases the computational burden and memory occupancy generated by deallocating unused obsolete objects at runtime. This approach may be considered a specific design pattern for partial differential equation solvers.

3.2 The Software Organisation

The more visible effect of our choices is that we have built various OMS3 components:

- whetgeo1d-1.0-beta
- netcdf-1.0-beta
- closureequation-1.0-beta
- buffer-1.0-beta
- numerical-1.0-beta

Internally, the classes are assembled by using some interfaces and abstract classes, since WHETGEO-1D is coded using the Java language.

In order to improve the re-usability of the Java code we adopted a generic programming approach (Berti, 2000) that consists in decoupling of algorithm implementations from the concrete data representation while preserving efficiency. The generic approach has been balanced with domain-specific ones that can improve the computational efficiency of the software, as is the case of the previously mentioned Thomas algorithm used in 1-D implementations.

Another requirement regards the division of software classes into three main groups, as the lack of a proper separation between the parameterisation of physical processes and their numerical solutions has been recognised as one of the weak points of existing land surface models (Clark et al., 2015b, 2021). One group describes the mathematical-physical problem, the second one implements the numerical solution (Berti, 2000), and the third one contains the growing group of concrete classes. The first group contains the SWRC, and hydraulic and thermal conductivity, and it forms a stand-alone library since its content can eventually be reused in the 2D and 3D version of WHETGEO-1D. Similarly, we grouped the classes that solve linear and nonlinear algebraic systems, containing the Thomas, conjugate gradient, and the various Netwon types of algorithms, in a second stand-alone library. The third group of classes gathers the concrete implementations and the variety of OMS3 components that are deployed.

The classes used, and their repository for third parties inspection, are illustrated in the 00_Notebooks referred in section 4 and in the supplemental material. However, there are three pivotal groups of classes that we want to mention here: These contain the description of the geometry of the integration domain, the closure equation, and the state equation.

3.2.1 Computational domain, i.e. the `Geometry` class

One of the key aspects to have in a generic solver regards the management of the grid and, in particular, the definition of its topology, or how grid elements are connected to each other. In the 1D case the description of the topology is quite simple since it can be implicitly contained in the vector representation: each element of the vector corresponds to a control volume of the grid and it is only connected with the elements preceding and following it. It is worth noting that this approach is peculiar to 1D problems and cannot be adopted for the 2D and 3D domains, where, especially when unstructured grids are used, the grid topology requires a smart implementation of the incidence and adjacency matrices.

For each control volume it is necessary to store its geometrical quantities, their position and dimension, its variables, its parameter set, and the form of the equation to be solved there, referred to in the following as “equation state". The appropriate arrangement of information together with the internal design of the classes allows us to create a generic finite volume solver.

3.2.2 Closure equations, i.e. the `ClosureEquation` abstract class

The `ClosureEquation` class is shown in the UML diagram of Fig. (3). As explained in the Introduction, one of the core concepts of modelling water and heat transport in soils is the SWRC. Soil is a multi-phase material, thus knowledge of its composition is of crucial importance in defining its un-

saturated hydraulic conductivity and its thermal properties, specific internal energy and thermal conductivity.

An abstract class `ClosureEquation` is defined to contain only abstract methods that would be overwritten by the concrete classes implementing it. The `ClosureEquation` class essentially defines a new data-type. A closer inspection of Fig. (3) reveals that the `ClosureEquation` is composed by aggregation with the `Parameter` class, which contains all the physical parameters of the model. Moreover, the `Parameter` class is implemented by using the Singleton pattern (Freeman et al., 2008). This design pattern is functional to have only one instance of the `Parameter` class shared by all the `ClosureEquation` objects and to provide a global point of access to the `Parameter` instance. The Simple Factory pattern `SoilWaterRetentionCurvesFactory` accomplishes the task of implementing the concrete classes. By preferring polymorphism to inheritance and using the Factory pattern (Gamma et al., 1995; Freeman et al., 2008), the developers can easily include and extend existing code or new formulations or parametrisations of SWRC. Besides, the Simple Factory fulfils the dependency inversion principle (Eckel, 2003), thus new extensions cannot affect the functioning of existing code. The same closure equation, for instance a particular SWRC, can be used to compute the soil water volume, when solving the R^2 equation, and the specific enthalpy of the soil, when solving the heat equation.

3.2.3 Equation state, i.e. the `EquationState` class

The `EquationState` class in Fig. (4) contains the implementation of the discretised form of the equation state of the PDE under scrutiny. It contains a reference each to the `ClosureEquation` object, to the `Geometry` and `ProblemVariables` objects.

Notably, the solver of a PDE problem can refer to the abstract class and to its abstract object without implying the specific concrete equation to be solved or its concrete parameterisations. Moreover, the compositionality of the `EquationState` allows the creation of new solvers from existing closures without the need to add new subclasses. As shown in the UML of Figure 4, the `EquationState` class defines methods used to linearise the PDE when it is nonlinear. For instance, it computes the first and second derivatives, and the functions necessary to define the Jordan decomposition as required by the NCZ algorithm. Specifically, these methods are `p`, `q`, `pIntegral`, `qIntegral`, and `computeXStar`.

In our code design the `ClosureEquation` class is limited to computing a physical parameterisation, whereas the `EquationState` class is used to discretise the equation state of the PDE, and whenever required to properly linearise it. Any new concrete `EquationState` subclass can either have the same physics of another with a different solver or a different physics with the same solver.

3.3 Generic Programming at work

As explained in Section 2.1.2, the definition of the surface boundary condition for the R^2 equation can require the introduction of an additional computation node at the soil surface to simulate the water depth. This means that we have two different equation states, one for the soil water content and one for the water depth. On the other hand, when considering the Neumann or Dirichlet boundary condition we have only one equation state, for the soil water content. The nonlinear solver of the NCZ algorithm must work seamlessly with any type of boundary condition used, and with any type of equation involved in the problem.

Let us consider, for example, the case of the R^2 equation with water depth as the surface boundary condition. Figure (5) reports the pseudocode for computing the nonlinear functions in the NCZ algorithm using a procedural approach. When traversing the computational grid we need to resort to an `if-else` statement to compute the nonlinear function of each control volume: the control volume N represents the control volume for the surface water and in this case we need to use the water depth equation state, $H(\psi)$ in Figure (5). The remaining control volumes represent the soil moisture and for them we need to use the SWRC equation state, $\theta(\psi)$ in Figure (5). The main limit of this approach is that the computation of the nonlinear functions, $H(\psi)$ and $\theta(\psi)$, is hard-wired into the code of the NCZ algorithm. This presents a shortcoming for the reusability of the code since, as the boundary condition changes and the physical problem changes, it is necessary to modify the `for` loop and the name of the objects computing the nonlinear functions.

Adopting the OOP and generic programming approach, Fig. (6), it is possible to implement the NCZ algorithm in such a way that enhances its reusability. The key feature is the decoupling of the computational grid from the algorithm(data) (Berti, 2000). This is achieved through two elements. The first consists in creating a container of the objects that deal with the equation states of the problem, `equationState`, eS in Fig. (6). Second, we use a label `equationStateID`, eSID in Fig. (6), to specify the behaviour of each control volume. So, the behaviour of each control volume is determined by this label and not by the position of the element in the grid. Specifically, when we traverse the grid we use the `equationStateID` to determine which object inside the container `equationState` to use.

The NCZ algorithm has been implemented in the `NestedNewtonThomas` class. The `NestedNewtonThomas` contains a reference to the `Thomas` object, whose task is to solve a linear system, and to a list of `EquationState` objects, Fig. (7). Considering the ubiquity of nonlinear problems in Hydrology and the robustness of the NCZ algorithm, the NCZ algorithm has been encapsulated in a stand-alone library.

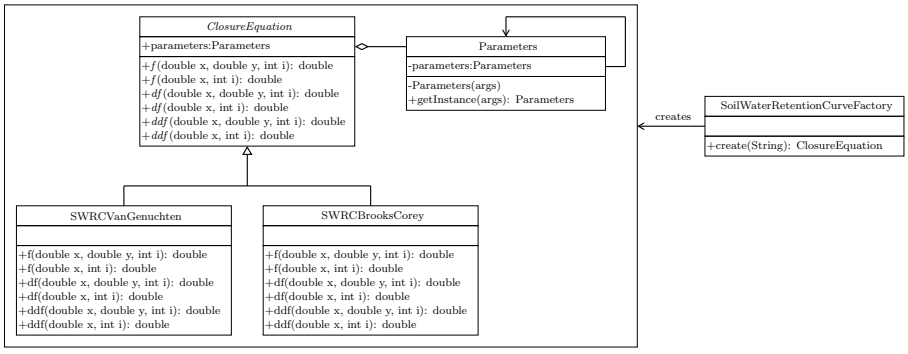

Figure 3. UML class diagram for the Java Simple Factory applied for the choice of the SWRC model. The `ClosureEquation` defines the interface that is implemented by the concrete classes `SWRCVanGenucthen` and `SWRCBrooksCorey`. The `ClosureEquation` is aggregated with the class `Parameters` containing the physical parameters of the model. The `Parameters` is implemented by using the Singleton pattern and its instance is inherited by the concrete classes `SWRCVanGenucthen` and `SWRCBrooksCorey`.

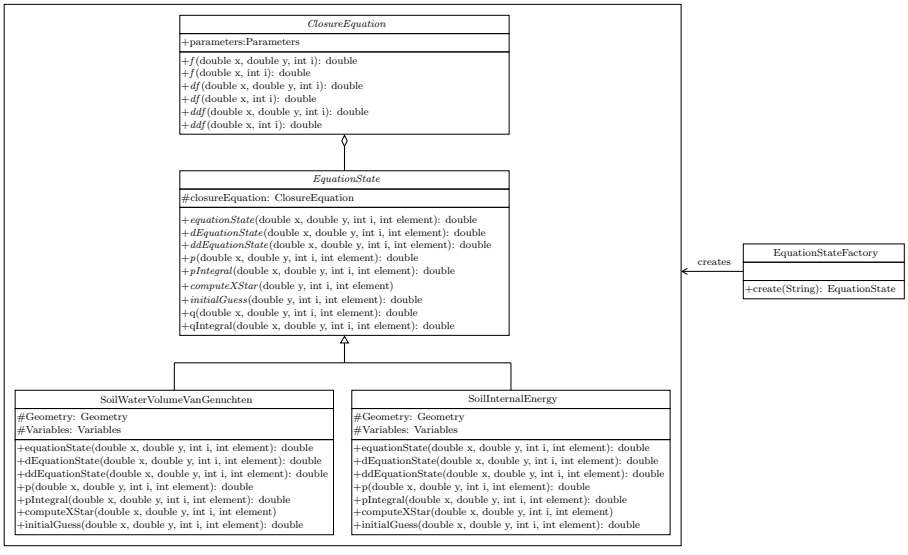

Figure 4. UML class diagram for the Java Simple Factory applied for the choice of the `EquationState` model. The `EquationState` defines the interface that is implemented by the concrete classes `SoilWaterVolumeVanGenucthen` and `SoilInternalEnergy`. The `EquationState` contains a reference to a `ClosureEquation` object.

The example has been illustrated in 1D but it becomes even more effective when working on 2D or 3D, especially with an unstructured grid.

4 Information for Users and Developers

While most of the what written so far is of general application, the deployment shown here is 1D. Information on WHETGEO-1D for users and developers is provided in the supplemental material, where there is a Jupyter Notebook that contains the guidelines for executing the codes for any of the components. Its name starts with "00_" and we call it "Notebook Zero" of the components. The latest executable code can be downloaded from

– https://github.com/geoframecomponents/WHETGEO-1D

and can be compiled by following the instructions therein. The version of the OMS3 compiled project can be found here https://github.com/GEOframeOMSProjects/OMS_Project_WHETGEO1D. The code can be executed in the OMS3 console, which can be downloaded and installed according to the instructions given at:

– https://geoframe.blogspot.com/2020/01/the-winter-school-on-geoframe-system-is.html

Some brief information about GEOframe can be found in Appendix B, and more comprehensive information is at:

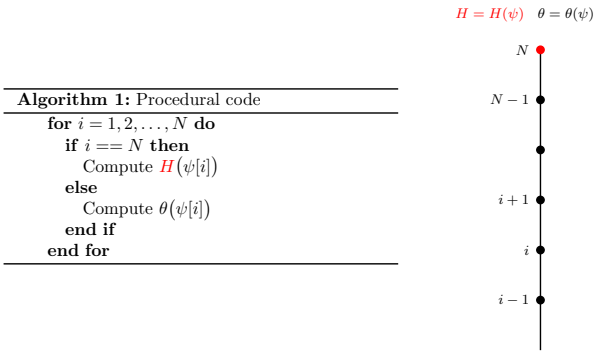

Figure 5. Adopting a procedural approach, the computation of the equation states is hard-wired into the code. The behaviour of each control volume is determined by an `if-else` statement according to the position of the element in the grid. In this case the properties of the grid, here the equation state, are joined with the topology. Here the non linear function $V(\psi)$ is replaced with either $H(\psi)$ or $\theta(\psi)$ according with the position of the node. To keep the pseudocode short, $H(\psi)$ and $\theta(\psi)$ stand for all the nonlinear function used in the NCZ algorithm, and the method `f` stands for one of the methods defined in the `EquationState` class.

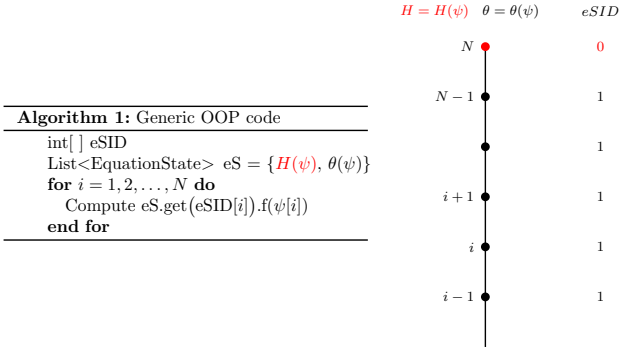

Figure 6. Adopting OOP with a generic programming approach, the computation of the equation states is independent from the grid. In fact, the behaviour of each control volume is determined by the vector eSID - `equationStateID` in the code - that determines which object belonging to the class `EquationState` of eS - `equationState` in the code - must be used to compute the equation state. In this manner it is possible to traverse the computation domain without resorting to the `if-else` statement. Here the non linear function $V(\psi)$ is consistently replaced with either $H(\psi)$ or $\theta(\psi)$ according to the position of the node. To keep the pseudocode short, $H(\psi)$ and $\theta(\psi)$ stand for all the nonlinear function used in the NCZ algorithm, and the method `f` stands for one of the methods defined in the class `EquationState`.

- https://abouthydrology.blogspot.com/2015/03/jgrass-newage-essentials.html
- https://geoframe.blogspot.com/2020/01/gsw2020-photos-and-material.html

To run the tests, please follow the instructions on the Github repository of the GEOframe components. If a user wants to compile the code themselves, they can use the appropriate Gradle script that guarantees independence from any IDE. For further information about input and output formats for WHETGEO-1D, please see the Notebook 00_WHETGEO1D_Richards.ipynb in the folder Documentation of the Zenodo distribution.

4.1 Workflow for Users

Examples of uses of WHETGEO-1D can be found in the form of Python Notebooks in the directory Notebooks/Jupyter. Documentation can be found in form of Python Notebooks in the directory Documentation. Simulations with WHETGEO-1D are run as OMS3 simulations. Therefore, the first operation to accomplish is to prepare the appropriate .sim files. For new users, many simulation files are available in the directory simulation of the Zenodo distribution. As shown in Fig. (8), in the modelling solutions that involve WHETGEO-1D, there is always a "Main" component that is in charge of running the core code for solving the PDE. The inputs and the outputs are treated by other OMS3 components. They are tied together by a Domain Specific Language (DSL) based on Groovy. This allows for great flexibility in using various input and output formats.

4.2 Inputs and Outputs

Input data can be broadly classified into time series, computational grid data, and simulation parameters. Time series are used to specify the boundary conditions of the problem. Time series are contained in .csv files with a specific format that is OMS3 compliant (David). With computational grid data we refer to the domain discretisation, initial condition, and soil parameters. All these data are stored in a netCDF file. Time series and computational grid data are elaborated with dedicated Python modules distributed under the geoframepy package (Tubini and Rigon, 2021). The simulation parameters, such as the start date and end date of the simulation, time step size, and file paths, are specified by the user in the OMS3 .sim file.

For the design of the output workflow we took advantage of the OMS3 system that allows the user to connect stand-alone components. Figure (8) shows the output workflow for saving output data and where `Main`, `Buffer`, and `netCDF writer` are the stand-alone OMS3 components. `Main` stands for the generic component having the responsibility of solving the PDE. `Buffer` has the responsibility of temporarily storing output data, and `Writer` handles the saving of data to the disk. The `Buffer` component has the sole purpose of storing data and this has two important advantages. The first is that it limits the number of accesses to the disk to save output, i.e. reducing the computational time. The second is that it introduces a layer separating the `Main` component from the `netCDF writer`. This increases the flexibility of the modelling solution, as future developers can

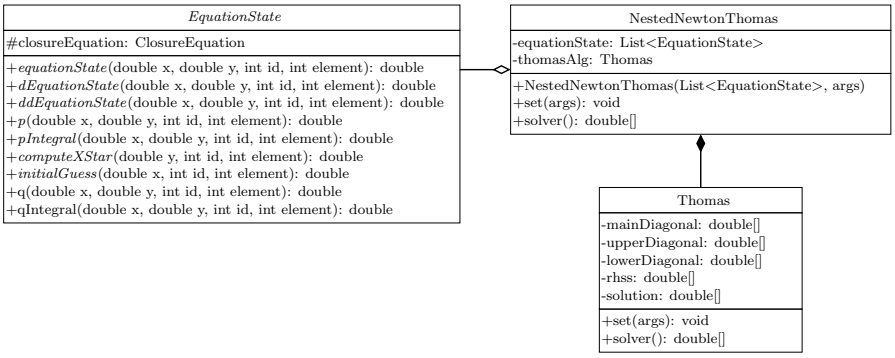

Figure 7. UML class diagram of the `NestedNewtonThomas` class. This class deals with the solution of the nonlinear system. The `NestedNewtonThomas` contains a reference each to the `Thomas` object, whose task it is to solve a linear system, and to a list of `EquationState` objects.

adopt different file formats, or develop different writer components that, instead of saving all the outputs, can save discrete outputs or aggregated outputs. The advantage is that developers need only know the legacy of the `Buffer` component and customise the both output file format and memory optimisation strategy, such as chuncking, according to their need. Currently all outputs are stored in a netCDF-3 format (Unidata, 2021). netCDF is a self-describing portable data format developed and maintained by UCAR Unidata. netCDF is commonly used by the Geo-science community and there is an ever-growing number of tools for processing and visualisation.

The choice of including the `Buffer` component in the workflow of the modelling solution is motivated by earlier experiences of the GEOtop community with the GEOtop model, and by more recent experience with the FreeThaw1D model (Tubini et al., 2020), where long spin-ups of the model ($\sim$ 1500 years) were required with consequent large output files ($\sim$ GB). Furthermore, in anticipation of the 2D and 3D developments, the netCDF-3 format is probably not the most appropriate and could be abandoned in favour of more performing file formats Unidata (a, b), such as netCDF-4 or HDF5.

WHETGEO-1D can be integrated with the built-in calibration component `LUCA` (Hay et al., 2006; Formetta et al., 2014a) and the `Verification` component, as shown in Fig. (9). The former is used to calibrate optimal parameters, the latter to compute the indices of goodness of simulated data versus measured data. Besides the `LUCA` component and the `Verification` component, it is necessary to add two more components, specifically the `Buffer calibration parameters` and the `Measurement point data`. The `Buffer` calibration parameters is needed to interface the `Main` component with the `LUCA` component. In fact, in the WHETGEO-1D `Main` component, physical parameters are stored as vectors, whilst `LUCA` handles calibration parameters as scalars (single value). The `Buffer calibration` component receives the optimal parameters set from the `LUCA` component and returns them packed in appropriate vectors. The `Verification` component receives as input two, OMS3-compliant time series: one for measured data and one for simulated data. In this case it is necessary to extract from the simulation output only the simulated data at the measurement points of the variable used to calibrate the model. These data are then saved as OMS3 time series. It interesting that the integration of WHETGEO-1D with the OMS3 built-in calibration components is achieved by adding two new, stand-alone components without modifying the source code of the existing components, i.e. the `Main` component, the `Buffer` component, and the `netCDF writer`.

4.3 Workflow for Developers

Here, as an example, we present how to add the Brooks-Corey (Brooks and Corey, 1964) model as an extension of the code base. The constitutive relationships are given by:

$$\theta(\psi) = \begin{cases} \theta_r + (\theta_s - \theta_r)\left(\dfrac{\psi_d}{\psi}\right)^n & \text{if } \psi \leq \psi_d \\ \theta_s & \text{if } \psi > \psi_d \end{cases} \tag{38}$$

$$K(\psi) = K_s \left[\frac{\theta(\psi) - \theta_r}{\theta_s - \theta_r}\right]^{3+\frac{2}{n}} \tag{39}$$

where θ_s and θ_r are, respectively, the saturated and residual values of the volumetric water content, ψ_d is the air-entry water suction value, n is the pore size distribution index, and K_s is the saturated hydraulic conductivity at saturation.

The standard approach to adding a new SWRC parametrization, here the Brooks and Corey model, requires the definition a new class that extends the abstract class `ClosureEquation`. This new class, `SWRCBrooksCorey`, provides the implementation

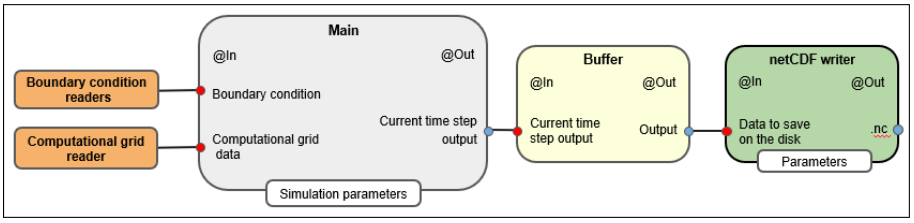

Figure 8. Workflow of WHETGEO-1D. The boundary condition readers, computational grid reader, `Main`, `Buffer`, and `netCDF writer` are stand-alone OMS3 components. `Main` stands for the generic component with the responsibility of solving the PDE, the `Buffer` temporarily stores output data that later are passed to the `netCDF writer` component, which in turn saves data to the disk. The `Buffer` component has the sole purpose of storing data. This has two adavantages: the first one is to limit the number of accesses to the disk to save output, i.e. reducing the computational time, and the second one is to introduce a layer separating the `Main` component, which handles the numerical solution of the PDE, and the component responsible for saving outputs.

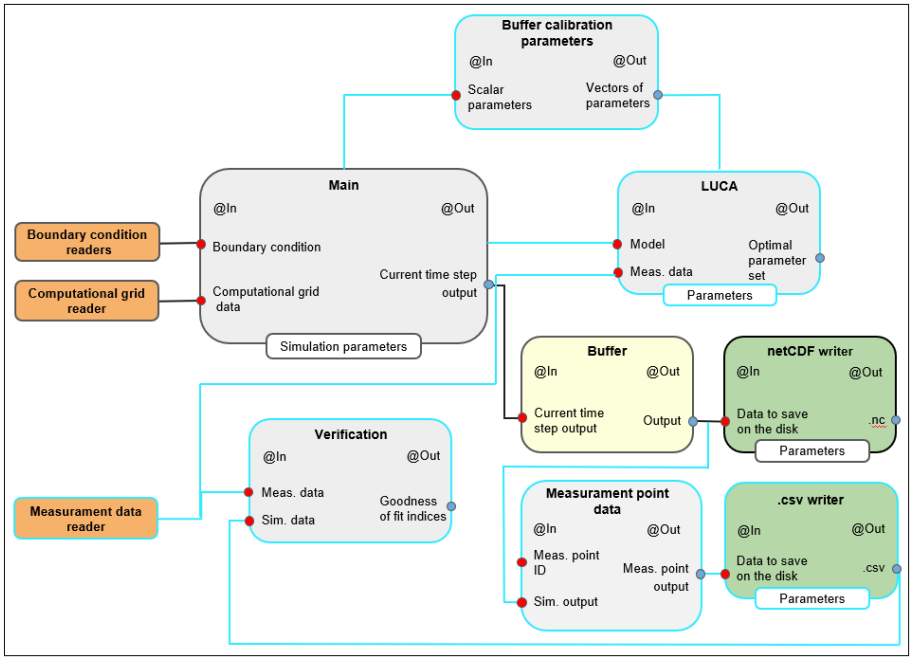

Figure 9. Workflow of WHETGEO-1D integrated with LUCA. The cyan lines identify those components required to integrate WHETGEO-1D with LUCA. The LUCA and Verification components are built-in OMS3 components. The former is used to calibrate optimal parameters, the latter to compute the indices of goodness of simulated data versus measured data. To integrate WHETGEO-1D with the OMS3 calibration components it is necessary to add two other components into the workflow, specifically the Buffer calibration parameters component and the measurement point data component. LUCA handles scalar parameters whilst WHETGEO 1D uses vectorial parameters. The `Buffer calibration parameters` creates an interface between these two components, simply creating vectorial parameters from scalar parameters. The `Verification` component requires the input time series as a .csv file of simulated data at the measurement points. To accomplish this requirement, it is necessary to modify the output strategy: the `Buffer` component passes output data to the `Measurement points data` component. This component extracts only the simulated variables at the measurement points and passes them to the OMS3 time series writer, `.csv writer`, which save the simulated time series as a .csv file. Notably, if the user is not interested in saving all the output of each run of the model during calibration, it is possible to remove the `netCDF writer` component.

of the abstract methods defined in the super class `ClosureEquation`, and inherits the association with the Parameters class. Specifically, the `SWRCBrooksCorey` class overrides the following methods:

- `f` calculates the water content for a given water suction value and set of parameters, Eq. (38);
- `df` calculates the first derivative of Eq. (38);
- `ddf` calculates the second derivative of Eq. (38).

In order to use the Brooks-Corey model in the Richards' equation it is necessary to define a new class, `SoilWaterVolumeBrooksCorey`, that extends the abstract class `StateEquation`. Specifically the

`SoilWaterVolumeBrooksCorey` class overrides the following methods:

- `equationState` calculates the water volume using the Brook-Corey model for a given water suction value and set of parameters, Eq. (18).
- `dEquationState` calculates the first derivative of the `equationState` function, in this the moisture capacity function. This method is used within the linearisation algorithm.
- `ddEquationState` calculates the second derivative of the `equationState` function. This method is relevant for those models where the ψ^* cannot be computed analytically but requires the application of a root finding method such as the bisection method. An example is the soil internal energy function when considering the phase change of water Tubini et al. (2020).
- `p` calculates the p function of the Jordan decomposition.
- `pIntegral` calculates the V_1 function of the Jordan decomposition.
- `computeXStar` calculates the ψ^* value to properly define the functions p and V_1.
- `initialGuess` calculates the initial guess for the linearisation algorithm.

5 Conclusions

In this paper we discussed the issues raised by implementing a new expandable system to model the Earth's CZ, WHETGEO, the core of which is a reliable and robust integrator of the Richards-Richardson equation and the associated energy budget.

We would like to stress that:

- WHETGEO makes available the NCZ algorithm which has a priori convergence assured for any choice of time step and for a great variety of boundary and initial conditions.
- The coupling between the R^2 equation and the heat advection diffusion equation is done using a recent algorithm (Casulli and Zanolli, 2005) that, to our knowledge has not been applied before for the same scope.
- The application of our methods could benefit a large range of Earth Science models, which at present require complicated workarounds (Regenass et al., 2021).
- The adoption of the mixed form of the equation allow the joint modelling of the saturated and unsaturated soil/groundwater conditions.
- Moreover, the coupling strategy with ponding water makes WHETGEO naturally suitable to investigate the partitioning of infiltration and surface run-off without any of the issues documented for other softwares (e.g. (Šimůnek et al., 2005)).

The implementation of WHETGEO has been shown to solve 3 software requirements, and the A to H design specifications. Each of these was analysed and thus influenced the choice of algorithms and code implementation. Of the hydrological issues presented in 7 observations, the issues 1 to 4 were also solved, and, the code provided some answers for observations 5 to 7 because:

- The code allows for the easy changing soil water retention curves; therefore, for instance, it would be easy to implement new ones taking more accurate account for the different processes in the various ranges of suction. A description on how to add a new soil water retention curve is presented in Section 4.3.
- The effect of temperature on viscosity has been included and, in the supplemental material, we have shown that temperature changes in soil have relevant effects on infiltration and runoff production.
- Freezing and thawing processes coupled with the R^2 equation will be inserted. As an intermediate step, the heat conduction with phase change has been implemented in the energy budget solver.

The flexibility of the code was obtained with the creation of the `StateEquation` and the `ClosureEquation` classes, illustrated in Section 3. Their organisation can be considered a new programming pattern that can be reused by programmers to implement the solution of partial differential equations, independently of our use of the Java language. Furthermore, we also chose to implement the objects containing data adopting the Singleton pattern. This choice violates the safety prescription that objects should preferably be immutable but, by avoiding the continuous allocation and deallocation (through the Java garbage collector) of new objects, we aimed at saving computational time. Also, this scheme can be considered a design choice that can be replicated by anyone interested in scientific computing.

The first deployment of the concepts was the 1D standalone water budget and the coupled water and energy budgets WHETGEO versions. The water budget was tested against analytical solutions presented in (Srivastava and Yeh, 1991) and (Vanderborght et al., 2005). Some behavioural simulations were also performed to show some features of the code, such as the ability to deal with switching boundary conditions. Also, we showed that WHETGEO can be easily extended to simulate the thermal regime in frozen soils, as described in Tubini et al. (2020), by merely adding the SFC model presented by (Dall'Amico et al., 2011).

The model is an entirely new code that was accurately designed to allow easy inspection and to support code literacy and classes reuse. It is open-source and built with open-source tools. It and its documentation fulfil the requirements of Open Science (Hall et al., 2021). In a science where a lot of daily research activities are based on computer simulations, this is a requirement that most of the existing models do not fulfil but that the Authors believe is profitable for the progress of science. Besides, WHETGEO is built with a chain of open-source tool, making it available to all researchers who wants to peruse it without constraint. Its documentation was produced using only open source tools and is available together with the distribution of the main code.

Code availability. The source code is written in Java using the object-oriented programming paradigm. The source code can be found at https://github.com/geoframecomponents/WHETGEO-1D and a frozen version at 10.5281/zenodo.5112727. The OMS3 project can be found at https://github.com/GEOframeOMSProjects/OMS_Project_WHETGEO1D and a frozen version is available on Zenodo at 10.5281/zenodo.5112727. The source code of external dependencies are provided at in the README of the github page https://github.com/geoframecomponents/WHETGEO-1D. WHETGEO-1D is deployed as an open source code to work alone or within the Object Modelling System version 3 framework (David et al., 2013). In the latter case it can be connected at run time with the many other components developed in the GEOframe system to provide hydrometeorological forcings and other fluxes, such as evapotranspiration. The simulations presented here can be found at 10.5281/zenodo.4749319. The code must be run within the OMS3 Console or using the Dockerized version of OMS3. To set up the environment please follow the steps described in the README file present in the Github repository https://github.com/GEOframeOMSProjects/OMS_Project_WHETGEO1D and in the GEOframe pages at https://geoframe.blogspot.com/2021/05/whetgeo-1d.html (last accessed 18 May 2021). Once you have installed OMS3, please follow the instructions contained in the Documentation folder. They contain all the details about simulations inputs and parameters.For those interested in using WHETGEO-1D and apply it in their research it is available the material of the first GEOframe Summer School held in Trento on 4-7 October, 2021, http://geoframe.blogspot.com/2021/06/geoframe-summer-school-2021-gss2021.html. The material of the School is available here https://osf.io/z53c6/ (DOI 10.17605/OSF.IO/Z53C6). The Fair Use & Publication Policy of the GEOframe group is available here https://osf.io/wgdyq/.

Appendix A: GEOframe

WHETGEO-1D was implemented as a Java component within the GEOframe, an open-source, component-based hydrological modelling system. Within GEOframe, each part of the hydrological cycle is implemented in a self-contained building block, an OMS3 component (David et al., 2013). Components can be joined together to obtain multiple modelling solutions that can accomplish from simple to very complicated tasks. GEOframe has proved great flexibility and robustness in several applications (Bancheri et al., 2020; Abera et al., 2017a, b). There are more than 50 components available that can be grouped into the following categories:

- Geomorphic and DEM analyses;
- Spatial extrapolation/interpolation of meteorological variables;
- Estimation of the radiation budget;
- Estimation of evapotranspiration;
- Estimation of runoff production with integral distributed models;
- Channel routing;
- Travel time analysis;
- Calibration algorithms.

Using the components for geomorphic and DEM analyses Rigon et al. (2006b), the basin can be discretised into Hydrological Response Units (HRUs), i.e., hydrologically similar parts, such as a catchment or a hillslope or one of its parts. The meteorological forcing data can be spatially interpolated using a geostatistical approach, such as the Kriging technique (Bancheri et al., 2018). Both shortwave and longwave radiation components are available for the estimation of the radiation budget (Formetta et al., 2013, 2016). Evapotranspiration can be estimated using three different formulations: the FAO Evapotranspiration model (Allen et al., 1998), the Priestley-Taylor model (Priestley and Taylor, 1972), and the Prospero model, (Bottazzi, 2020; Bottazzi et al., 2021). Snow melting and the snow water equivalent can also be simulated with three models, as described in (Formetta et al., 2014b). Runoff production is performed by using the Embedded Reservoir Model (ERM) or a combination of its reservoirs (Bancheri et al., 2020). The discharge generated at each hillslope is routed to the outlet using the Muskingum-Cunge method (Bancheri et al., 2020). Travel time analysis of a generic pollutant within the catchment can be done using the approach proposed in (Rigon et al., 2016b, a). Model parameters can be calibrated using two algorithms, and several objective functions: Let Us CAlibrate (LUCA) (Hay et al., 2006) and Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995). A graph-based structure, called NET3 (Serafin, 2019), is employed for the management of process simulations. NET3 is designed using a river network/graph structure analogy, where each HRU is a node of the graph, and the channel links are the connections between the nodes. In any NET3 node, a different modelling solution can be implemented and nodes (HRUs or channels) can be connected or disconnected at run-time through scripting. GEOframe is open source and helps the reproducibility and replicability of

research (Bancheri, 2017). Developers and users can easily collaborate, share documentation, and archive examples and data within the GEOframe community.

Appendix B: OMS3

The Object Modeling System v.3 (OMS3) is a component-based environmental modelling framework that provides a consistent and efficient way to: 1) create science simulation components; 2) develop, parameterise, and evaluate environmental models, and modify and adjust them as science advances; and 3) re-purpose environmental models for emerging customer requirements (David et al., 2013).

In OMS3 the term component refers to self-contained, separate software units that implement independent functions in a context-independent manner (David et al., 2013). This means that developers and researches can build their model as composition of stand-alone components, moving away from the monolithic approach. The entire GEOframe system, and therefore WHETGEO-1D, is built upon the OMS3 framework.

Compared to other Environmental Modelling Frameworks (EMF), OMS3 is characterised by being a non-invasive and lightweight framework (Lloyd et al., 2011). That is to say, the model code is not tightly coupled with the underlying framework - OMS3 -, i.e. the environmental modeller does not need a deep knowledge of the API, and the modelling components can still function and continue to evolve outside the framework (David et al., 2013). In fact, OMS3 relies on specific annotations to provide meta data for Java code. These annotations describe elements such as classes, fields, and methods, and are used by the framework to interpret the component as a building block of the modelling solution (MS), hence controlling its connectivity and data flow (David et al., 2013). It is worth noting that, being meta data, these annotations do not directly affect the execution of the source code outside the OMS3 - non-invasive and lightweight framework.

Besides the technical aspects, the adoption of a software framework has a positive effect on "non-functional" quality attributes, such as maintainability, portability, reusability and understandability (David et al., 2013). The component-based approach allows the developer to break down the problem into smaller parts, each one tackled by a specific component. Hence, the components are joined together to build the desired modelling solution (point B). This facilitates the construction of new MSs, thanks to the plug-in system of model components (David et al., 2013; Peckham et al., 2013; Serafin, 2019). Thanks to the modularity, the updating of a component with the most recent scientific advances is facilitated and has no side effects on the other components. The other advantage regards the long term development of the code. From past experiences, one of the main limits to model development and maintenance was related to the lack of a proper software architectural design (Rizzoli et al., 2006; David et al., 2013; Formetta et al., 2014a; Bancheri, 2017; Serafin, 2019). Moreover, it is interesting to note that the component-based approach encourages collective model development (Serafin, 2019) and also eases the attribution of authorship since any component is a stand-alone chunk of code and can be authored separately.

Besides, the adoption of an environmental modelling framework promotes the concept of reproducible research, easing third parties inspection and providing consistent and verifiable model results (Formetta et al., 2013; Bancheri, 2017; Serafin, 2019).

Another advantage of using OMS3 is represented by the opportunity to keep the code development transparent to the user.

Appendix C: $\mathbf{R^2}$ test cases

In this section we test the solver of the R^2 equation against the analytical solutions presented by Srivastava and Yeh (1991) and by Vanderborght et al. (2005). Then we present and discuss two "behavioural" test cases to try out WHETGEO-1D in simulating both the infiltration excess and the saturation excess process.

C1 Analytical solution Srivastava and Yeh (1991)

Srivastava and Yeh (1991) derived an analytical solution describing the one-dimensional transient infiltration in an homogeneous and layered soil. The hydraulic properties of the soil are described by the following constitutive relations:

$$K(\psi) = K_s e^{\alpha\,\psi} \tag{C1}$$

$$\theta(\psi) = \theta_r + (\theta_s - \theta_r) e^{\alpha\,\psi} \tag{C2}$$

where Ks is the saturated hydraulic conductivity, θ_r is the residual water content, θ_s is the saturated water content, and α is the soil pore-size distribution parameter, representing the desaturation rate of the SWRC. The lower boundary condition is represented by the water table, $\psi = 0$, while the upper boundary condition is subjected to a constant flux, q. The initial condition corresponds to the steady state profile due to a prescribed initial flux at the soil surface and prescribed pressure at the lower boundary. The analytical solution is derived by linearising Richards' equation and using Laplace's transformation. Details on the analytical solution can be found in (Srivastava and Yeh, 1991).

C1.1 Homogeneous soil

We consider a one-dimension homogeneous soil layer of 1 $[\mathrm{m}]$ depth (TP1). The saturated hydraulic conductivity value is assumed to be 1.0 $[\mathrm{cm\,h^{-1}}]$, with $\theta_s = 0.45$ $[\mathrm{m^3\,m^{-3}}]$,

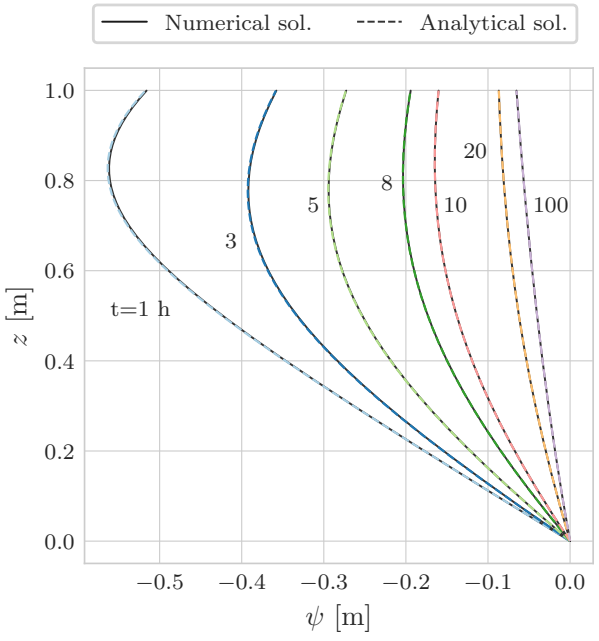

Figure C1. Comparison between the analytical and numerical solutions for the test problem TP1.

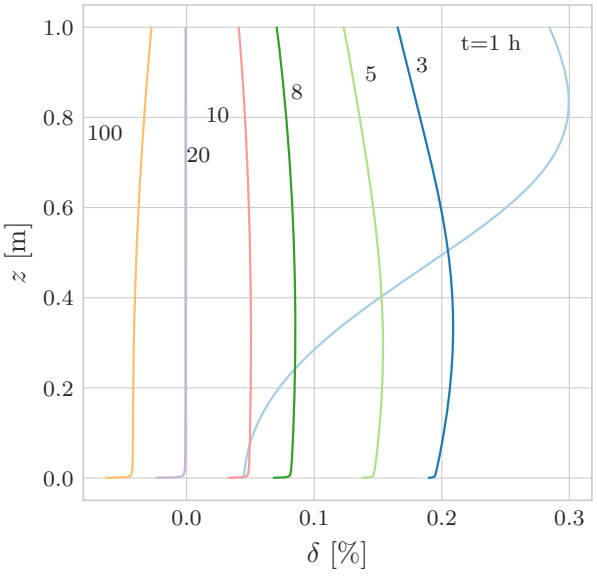

Figure C2. Relative water suction error for the test problem TP1.

$\theta_r = 0.2$ [$m^3\,m^{-3}$], and $\alpha = 0.01$ [cm^{-1}]. The initial condition is determined by imposing as lower boundary condition $\psi = 0$ [m] and a constant water flux at the soil surface $q_A = 0.1$ [$cm\,h^{-1}$]. For times greater than 0 the water flux at the soil surface is $q_B = 0.9$ [$cm\,h^{-1}$]. The domain is discretised with a uniform grid space $\Delta z = 0.001$ [m] and the time step is $\Delta t = 60$ [s]. The model accuracy is enhanced by allowing two Picard iterations per time step. Figure (C1) shows a comparison between the numerical and the analytical solutions.

C1.2 Layered soil

In this numerical problem (TP2) we consider one-dimensional vertical infiltration toward the water table through a layered soil. The initial condition is determined by imposing as lower boundary condition $\psi = 0$ [m] and a constant water flux at the soil surface $q_A = 0.1$ [$cm\,h^{-1}$]. For times greater than 0 the water flux at the soil surface is $q_B = 0.9$ [$cm\,h^{-1}$]. The domain is discretised with a uniform grid space $\Delta z = 0.001$ [m] and the time step is $\Delta t = 60$ [s]. The model accuracy is enhanced by allowing two Picard iterations per time step. The hydraulic conductivity at the interface is computed as the harmonic mean of the neighbours (Romano et al., 1998). Comparison between the numerical and the analytical solution for water suction is shown in Fig. (C3).

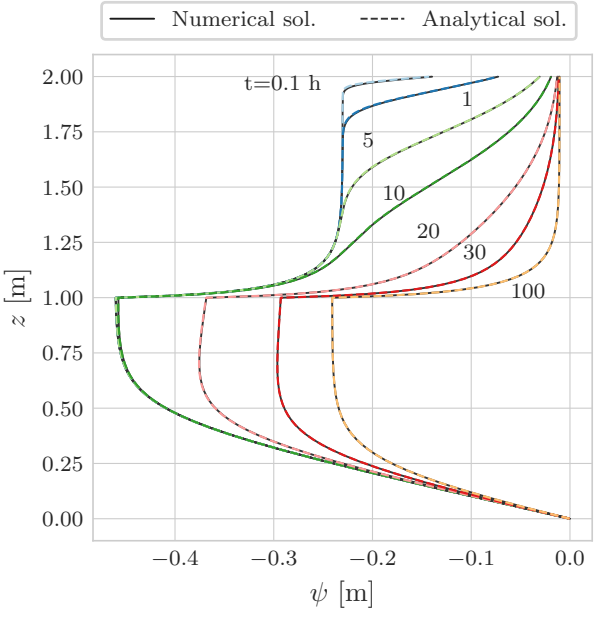

Figure C3. Comparison between the analytical and numerical solutions for the test problem TP2.

C2 Analytical solution Vanderborght et al. (2005)

The next test case was defined by Vanderborght et al. (2005) to evaluate the steady-state flux in layered soil profiles. For this numerical problem (TP3) we consider a soil column of 2 [m] depth with one soil type for depth 0 [m] – 0.5 [m] overlying another soil type for depth 0.5 [m] – 2 [m], specifically for loam over sand, sand over loam, and clay over sand. The

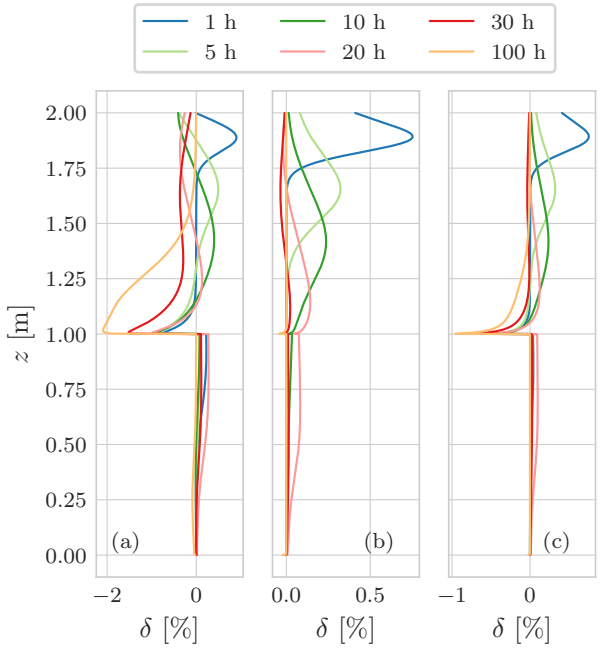

Figure C4. Comparison of relative water suction error δ for the test problem TP2 using different interface hydraulic conductivity algorithms. Panel (a) is computed with max., panel (b) with harmonic mean, and panel (c) with geometric mean. As reported in (Romano et al., 1998), the harmonic mean offers the best agreement with the analytical solution. This is particularly evident at the interface between the two layers.

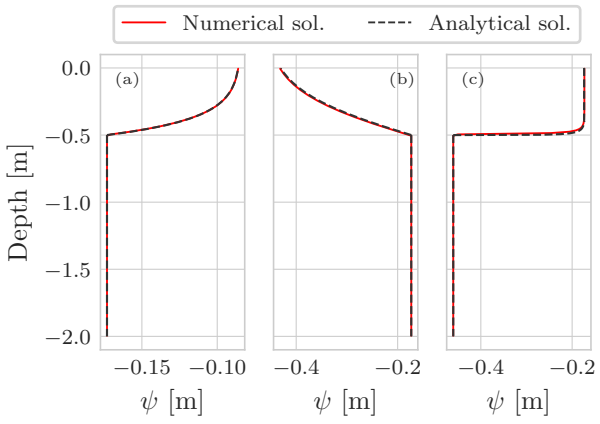

Figure C5. Comparison between the analytical and the numerical solution for the test problem TP3. The three panels show the vertical profile of water suction at steady state for a constant flow rate of $5.79\,10^{-8}$ [$\mathrm{m\,s^{-1}}$]: (a) clay-sand soil profile, (b) loam-sand soil profile, and (c) sand-loam soil profile.

soil parameters are defined in Tab. (C1). The initial condition for water suction is a uniform profile with $\psi = -20$ [m], the surface boundary condition is a constant flux $q = 5.79\,10^{-8}$ [$\mathrm{m\,s^{-1}}$], and at the bottom we impose a free drainage boundary condition. The domain is discretized with a uniform grid space $\Delta z = 0.01$ [m] and the time step is $\Delta t = 3600$ [s]. In order to reach the steady state condition the simulation lasts 2 years. Comparison between the numerical and the analytical solution is shown in Fig. (C5).

C3 Surface boundary condition

The definition of the surface boundary condition is a nontrivial task since it is a system-dependent boundary condition. The infiltration rate through the soil surface depends on precipitation, rainfall intensity J, and on the moisture condition of the soil. Because of this, the surface boundary condition may change from the Dirichlet type - prescribed water suction - to the Neumann type - prescribed flux - and vice-versa. The works by Horton (1933) and Dunne and Black (1970) establish the conceptual framework to explain the runoff generation.

The infiltration excess or Horton runoff occurs when the rainfall intensity is larger than infiltration capacity of the soil:

$$\left|J\right| > \left| -K(\psi)\frac{\partial}{\partial z}(\psi + z)\right|_{z=0} \tag{C3}$$

Infiltration excess is most commonly observed with short-duration, intense rainfall.

The saturation excess or Dunnian runoff occurs when the soil is saturated and additional water exfiltrates at the soil surface. Saturation excess generally occurs with long-duration, moderate rainfall, or with a series of successive precipitation events. In this case the soil depth or the presence of shallow fragipan are determining factors for saturation excess. Another possible cause is the rise of the water table up to the soil surface.

C3.1 Infiltration excess

In this numerical experiment we consider a homogeneous soil of 3 [m] depth. Soil hydraulic properties are described with the Van Genuchten's model, Table (C2).

The initial condition is assumed to be hydrostatic with $\psi = 0$ [m] at the bottom. The surface boundary condition is a synthetic rainfall, as in Fig. (C6) (a), lasting 15 [min] with constant intensity of 0.028 [$\mathrm{mm\,s^{-1}}$]. At the bottom we prescribed a Dirichlet boundary condition with constant $\psi = 0$ [m] so the transient is driven only by the surface boundary condition. In Fig. (C6) panel (a), the time is indicated when it would be necessary to switch from the Neumann type to the Dirichlet type boundary condition.

Figure (C7) shows a comparison of water ponding at the soil surface considering two different initial conditions of the soil, wet and dry. For the wet case, the initial condition is hydrostatic with $\psi = 0$ [m] at the bottom. For the dry case, the initial condition is hydrostatic with $\psi = -100$ [m] at the bot-

Table C1. Hydraulic properties for the three soil types used in the Vanderborght test case (TP3).

Soil type	θ_s [$m^3\,m^{-3}$]	θ_r [$m^3\,m^{-3}$]	α [m^{-1}]	n [$-$]	K_s [$m\,s^{-1}$]
Sand	0.43	0.045	15.0	3	$1.16e-04$
Loam	0.43	0.08	4.0	1.6	$5.79e-06$
Clay	0.4	0.1	1.0	1.1	$1.16e-06$

Table C2. Hydraulic properties of the silty clay loam soil Bonan (2019) for the Horton runoff numerical experiment.

Soil type	θ_s [$m^3\,m^{-3}$]	θ_r [$m^3\,m^{-3}$]	α [m^{-1}]	n [$-$]	K_s [$m\,s^{-1}$]
Silty clay loam	0.43	0.089	1.0	1.23	$1.94e-07$

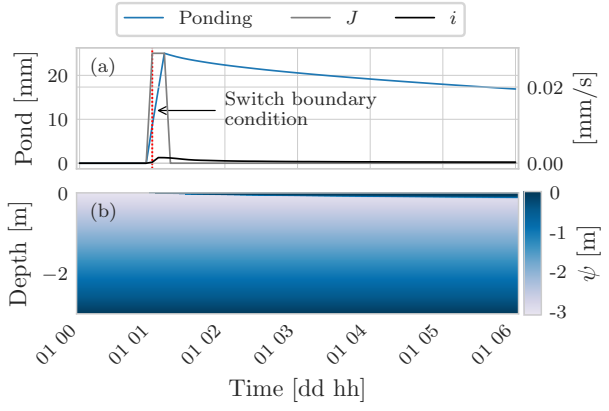

Figure C6. Panel (a) shows a comparison between the rainfall intensity J and actual soil infiltration i. The rainfall intensity exceeds the actual infiltration rate so water builds up at the soil surface (blue line). Panel (b) shows the time evolution of the water suction within the soil. From the numerical point of view, as water builds up at the soil surface it would be necessary to switch the boundary condition, from Neumann type to Dirichlet type.

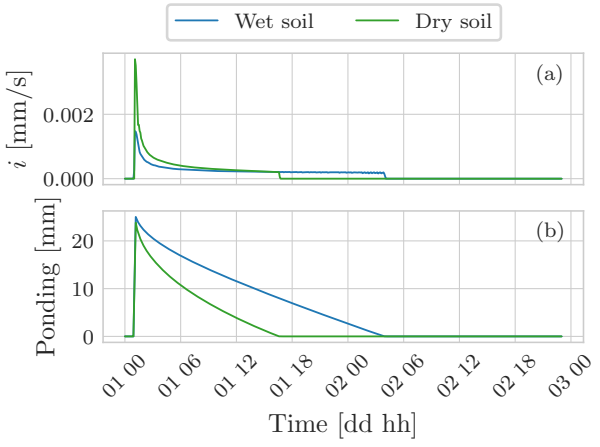

Figure C7. Panel (a) shows a comparison between the infiltration rate for two cases: wet and dry initial condition. In the dry case, soil infiltration is greater than the wet case even though the hydraulic conductivity is smaller. This is due to the higher capillary gradients that develop in the soil. Panel (b) shows the time evolution of the water ponding at the soil surface.

tom. In the wet initial condition the hydraulic conductivity is higher than for the dry initial condition, however, in the dry case the capillary gradient is larger and because of this the soil infiltration capacity is higher, as in Fig. (C7) panel (a). With regards to the water ponding, the maximum value is almost the same in both cases, 1 [mm] higher in the wet case, but the time evolution is different: in the wet case the water only infiltrates completely 13 [h] later than the dry case. This delay may seem counter-intuitive since wetter conditions are associated with higher values of hydraulic conductivity, Fig. (C8), but in the wet soil the capillary gradients are smaller than in the dry soil, Fig. (C9).

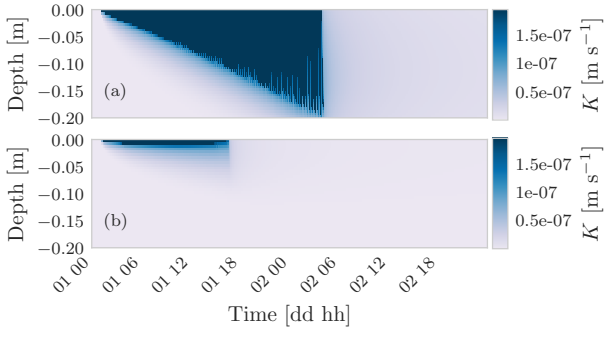

Figure C8. Panel (a) shows the hydraulic conductivity field for the case of wet soil, while panel (b) shows the hydraulic conductivity field for the case of dry soil.

C3.2 Saturation excess

In this section we present two numerical experiments to simulate the saturation excess process. Saturation excess is more critical, in terms of simulation stability, than the infiltration excess (Forums). We consider two cases: one in which the water table reaches the soil surface; and another in which the total rainfall amount is larger than the maximum water hold-

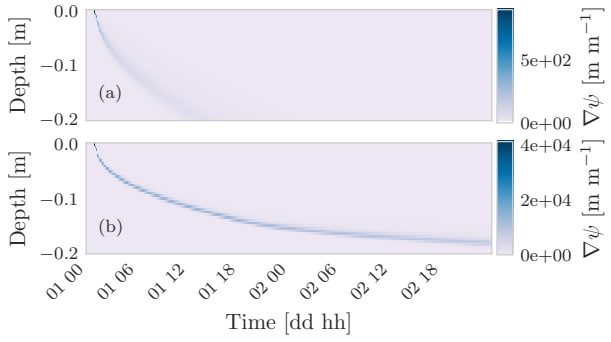

Figure C9. Panel (a) shows the capillary gradient for the case of wet soil, panel (b) shows the the capillary gradient for the case of dry soil. As can be seen, in the dry soil the capillary gradient is two orders of magnitude larger than in the wet soil. Because of this higher gradient water infiltrates faster in the dry soil than in the wet soil.

ing capacity but the rainfall intensity is less than the maximum infiltration rate.

Firstly, we consider a layered soil of 3 [m] depth. The thicknesses of the loamy layer and clay layer are, respectively, 0.5 [m], and 2.5 [m]. The soil hydraulic properties are described with the Van Genuchten's model, Tab. (C3). The initial condition is assumed to be hydrostatic with $\psi = 0$ [m] at the bottom. At the surface boundary condition we prescribe no rainfall, while at the bottom a variable Dirichlet boundary condition is prescribed Fig. (C10) (a). The transient is driven by variation of the water table. In Fig. (C10) panel (a) the time is indicated when it would be necessary to switch the surface boundary condition from the Neumann type to the Dirichlet type boundary condition and vice-versa.

Secondly, we consider a layered soil of 3 [m] depth. The thicknesses of the loamy sand layer and the clay layer are, respectively, 0.3 [m], and 2.7 [m]. The soil hydraulic properties are described with the Van Genuchten's model, Tab. (C4). The initial condition is assumed to be hydrostatic with $\psi = -2$ [m] at the bottom. The surface boundary condition is a synthetic rainfall Fig. (C11) (a), at the bottom we prescribed a Dirichlet boundary condition with constant $\psi = -2$ [m] so the transient is driven only by the surface boundary condition. Figure (C11) panel (b) shows the time evolution of the degree of saturation within the soil. Initially water infiltrates in the soil but then the clay layer, which is characterised by a lower conductivity than the loam-sand layer, limits the deep infiltration causing the saturation of the loam-sand layer from below.

Repeating the above numerical experiment with a thicker loam-sandy layer, Fig. (C12), there is no water ponding at the soil surface. In this case all the rainfall can infiltrate into the loam-sandy layer thanks to the increased water storage capacity.

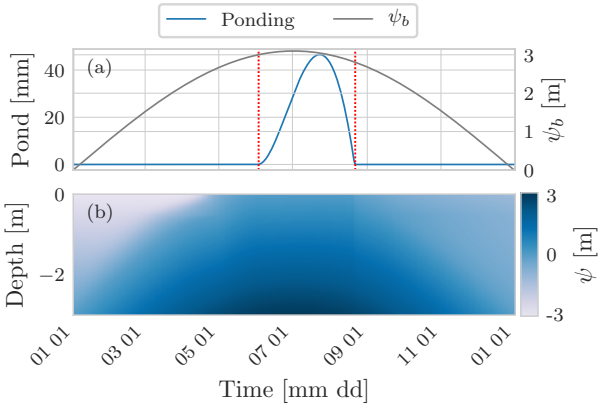

Figure C10. Panel (a) shows the water table position, Dirichlet boundary condition, and the water ponding at the soil surface. The dotted red lines indicate the times at which the boundary condition at the soil surface should be switched. The first line indicates the switch from the Neumann type to the Dirichlet type since water starts building up. The second line indicates the switch form the Dirichlet type to the Neumann type because there is no water at the soil surface. Panel (b) shows the time evolution of the water suction within the soil.

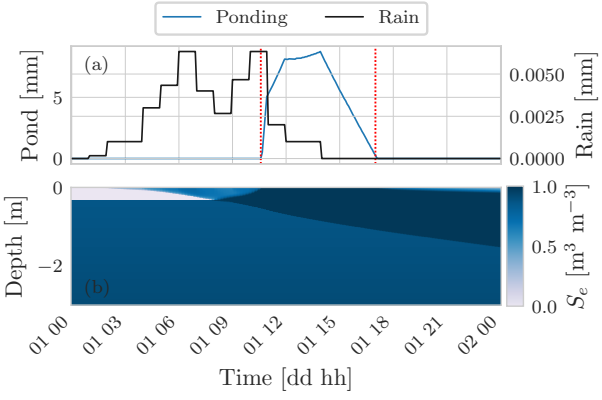

Figure C11. Panel (a) shows the rainfall and the water ponding at the soil surface. Initially rainfall can infiltrate into the soil and no water builds up a the soil surface. As the uppermost layer of the soil saturates water starts ponding at the soil surface. The dotted red lines indicate the times at which the boundary condition at the soil surface should be switched. Panel (b) shows the time evolution of the saturation degree within the soil.

Appendix D: Energy budget

Here we present a behavioural test case on the pure heat conduction considering the surface energy balance. The purpose of this simulation is twofold. First, we show an application of WHETGEO-1D that exploits existing GEOframe components to model the external components of the surface energy budget, specifically: the incoming shortwave radia-

Table C3. Hydraulic properties of the loam layer and clay layer, respectively, Bonan (2019) for the numerical experiment on Dunnian runoff due to water table rising.

Soil type	θ_s [$m^3\ m^{-3}$]	θ_r [$m^3\ m^{-3}$]	α [m^{-1}]	n [$-$]	K_s [$m\ s^{-1}$]
Loam	0.43	0.078	3.6	1.56	$2.8889e-06$
Clay	0.38	0.068	0.8	1.09	$5.5556e-07$

Table C4. Hydraulic properties of the loamy sand layer and clay layer, respectively, Bonan (2019) for the numerical experiment on Dunnian runoff due rainfall.

Soil type	θ_s [$m^3\ m^{-3}$]	θ_r [$m^3\ m^{-3}$]	α [m^{-1}]	n [$-$]	K_s [$m\ s^{-1}$]
Loamy sand	0.41	0.057	12.4	2.28	$4.0528e-05$
Clay	0.38	0.068	0.8	1.09	$5.5556e-07$

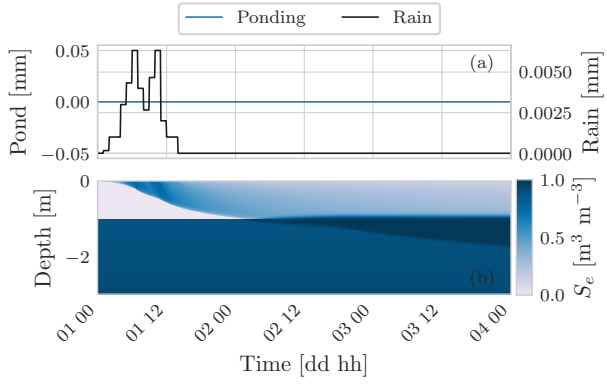

Figure C12. In this numerical experiment there is no saturation excess since the greater thickness of the uppermost layer ensure a sufficient water storage capacity for the forcing rainfall.

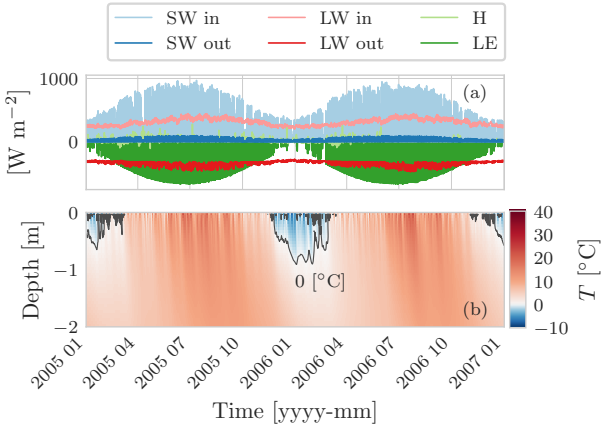

Figure D1. Behavioural test case of pure heat conduction in soil considering the surface energy budget. Panel (a) shows the surface energy fluxes driving the simulation. The external fluxes, incoming shortwave and longwave radiation, and the latent heat flux, are computed with existing GEOframe components. Panel (b) shows the thermal regime of the uppermost 2 [m] of the soil column. As can be seen, during winter the temperature of the uppermost layer goes below 0 [°C], the grey line is the 0 [°C] isotherm, therefore, it is not reasonable to overlook the phase change of water.

tion (Formetta et al., 2013), the incoming longwave radiation (Formetta et al., 2016), and the latent heat flux. Second, we show how to easily include the phase change of water by adding a new closure equation that describes the SFC model presented by Dall'Amico et al. (2011)

The soil column is 30 [m] deep and the initial condition is a constant temperature profile $T = 12$ [°C]. Figure (D1) panel (a) shows the components of the surface energy fluxes and the thermal regime of the uppermost 2 [m] of the soil column. As can be seen in panel (b), the soil temperature falls below 0 [°C]; therefore it is not reasonable to neglect freezing and thawing processes.

By including the closure equation of the SFC presented by Dall'Amico et al. (2011), it is possible to consider the phase change of water. Moreover, compared to the FreeThaw-1D model Tubini et al. (2020), in WHETGEO-1D it is possible to drive the simulation of the soil thermal regime by using the surface energy budget. This aspect represent a novelty with respect to FreeThaw-1D and was obtained with a minimum effort thanks to the code design we adopted. The soil column is 30 [m] depth and the initial condition is a constant temperature profile $T = 12$ [°C]. The parameters of the SFC are presented in Tab. (D1) Figure (D1) shows in panel (a) the components of the surface energy fluxes, and the thermal regime of the uppermost 2 [m] of the soil column. As can be seen in panel (b), the soil temperature fall below the 0 [°C] thus neglecting freezing and thawing processing results in a strong approximation.

Figure (D3) shows a comparison between the two simulations.

Author contributions. Conceptualization, R.R. and N.T.; Methodology: N.T. and R.R. Software engineering: N.T. and R.R.; Code writing N.T. Code Revision: RR and NT; Data curation and simulations, N.T.; Paper Writing, Review, Editing; N.T. and R.R.; Lan-

Table D1. Parameters of the SFC model.

Soil type	θ_s [m^3 m^{-3}]	θ_r [m^3 m^{-3}]	α [m^{-1}]	n [−]
Clay	0.38	0.068	0.8	1.09

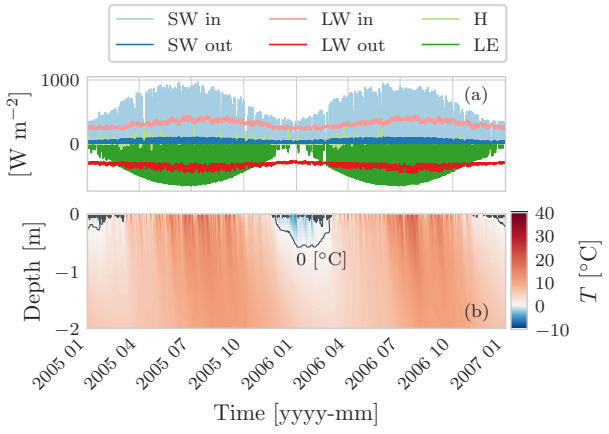

Figure D2. Behavioural test case of the pure heat conduction in soil considering the surface energy budget and the phase change of water. Panel (a) shows the surface energy fluxes driving the simulation. The external fluxes, incoming shortwave and longwave radiation, and the latent heat flux, are computed with existing GEOframe components. Panel (b) shows the thermal regime of the uppermost 2 [m] of the soil column.

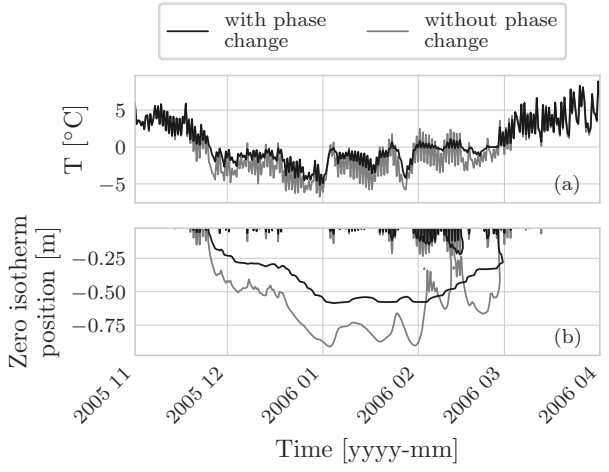

Figure D3. Panel (a) shows a comparison of the soil temperature at 0.05 [m] considering the phase change of water and without. As can be seen, in the latter case the soil temperature reaches lower value and fluctuates more: when considering the phase change of water we include in the problem the latent heat of water that increase the thermal inertia of the soil. Panel (b) shows a comparison of the position of the zero-isotherm in the two simulation. When the phase of water is not included the zero-isotherm does deeper into the ground.

guage revision, Joseph Eamonn Tomasi; Documentation: N.T., R.R. and Concetta D'Amato; Funding: R.R.

Competing interests. The authors declare no conflict of interest.

Acknowledgements. The authors would like to thank Professor Vincenzo Casulli and Professor Michael Dumbser of the Department of Civil, Environmental and Mechanics Engineering at the University of Trento for their fruitful discussions on the numerical aspects of the work. We thank Concetta D'Amato for the help in preparing the supporting material. We would like to thank Joseph Eamonn Tomasi for his invaluable help with the linguistic revision. We especially thank the two anonymous reviewers for their constructive comments that helped to improve our paper.

This work has been partially supported by a Ph.D. grant by the Department of Civil, Environmental and Mechanics Engineering at the University of Trento and by the Italian MIUR Project (PRIN 2017) "WATer mixing in the critical ZONe: observations and predictions under environmental changes-WATZON" (project code: 2017SL7ABC).

References

Abera, W., Formetta, G., Borga, M., and Rigon, R.: Estimating the water budget components and their variability in a pre-alpine basin with JGrass-NewAGE, Advances in water resources, 104, 37–54, https://doi.org/10.1016/j.advwatres.2017.03.010, 2017a.

Abera, W., Formetta, G., Brocca, L., and Rigon, R.: Modeling the water budget of the Upper Blue Nile basin using the JGrass-NewAge model system and satellite data, Hydrology and Earth System Sciences, 21, 3145–3165, https://doi.org/10.5194/hess-21-3145-2017, 2017b.

Allen, R. G., Pereira, L. S., Raes, D., Smith, M., et al.: Crop evapotranspiration-Guidelines for computing crop water requirements-FAO Irrigation and drainage paper 56, Fao, Rome, 300, D05 109, 1998.

Ashby, S. F. and Falgout, R. D.: A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations, Nuclear science and engineering, 124, 145–159, https://doi.org/10.13182/NSE96-A24230, 1996.

Bancheri, M.: A flexible approach to the estimation of water budgets and its connection to the travel time theory, Ph.D. thesis, University of Trento, 2017.

Bancheri, M., Serafin, F., Bottazzi, M., Abera, W., Formetta, G., and Rigon, R.: The design, deployment, and testing of kriging models in GEOframe with SIK-0.9. 8, Geoscientific Model Development, 11, 2189–2207, https://doi.org/10.5194/gmd-11-2189-2018, 2018.

Bancheri, M., Rigon, R., and Manfreda, S.: The GEOframe-NewAge Modelling System Applied in a Data Scarce Environment, Water, 12, 86, https://doi.org/10.3390/w12010086, 2020.

Benard, P., Zarebanadkouki, M., and Carminati, A.: Physics and hydraulics of the rhizosphere network, Journal of Plant Nutrition and Soil Science, 182, 5–8, https://doi.org/10.1002/jpln.201800042, 2019.

Berti, G.: Generic software components for Scientific Computing, Ph.D. thesis, Von der Fakultät für Mathematik, Naturwissenschaften und Informatik der Brandenburgischen Technischen Universität Cottbus, 2000.

Bisht, G. and Riley, W. J.: Development and verification of a numerical library for solving global terrestrial multiphysics problems, Journal of Advances in Modeling Earth Systems, 11, 1516–1542, https://doi.org/10.1029/2018MS001560, 2019.

Bloch, J.: Effective Java: Programming Language Guide, Addison-Wesley Professional, 2001.

Bonan, G.: Climate change and terrestrial ecosystem modeling, Cambridge University Press, 2019.

Bottazzi, M.: Transpiration theory and the Prospero component of GEOframe, Ph.D. thesis, University of Trento, 2020.

Bottazzi, M., Bancheri, M., Mobilia, M., Bertoldi, G., Longobardi, A., and Rigon, R.: Comparing evapotranspiration estimates from the GEOframe-Prospero model and three empirical models under different climate conditions, Water, 13, 1221–1243, https://doi.org/10.3390/w13091221, 2021.

Brooks, R. and Corey, T.: Hydraulic properties of porous media, Hydrology Papers, Colorado State University, 24, 37, 1964.

Brugnano, L. and Casulli, V.: Iterative solution of piecewise linear systems, SIAM Journal on Scientific Computing, 30, 463–472, https://doi.org/10.1137/070681867, 2008.

Brugnano, L. and Casulli, V.: Iterative solution of piecewise linear systems and applications to flows in porous media, SIAM Journal on Scientific Computing, 31, 1858–1873, https://doi.org/10.1137/08072749X, 2009.

Casulli, V.: A high-resolution wetting and drying algorithm for free-surface hydrodynamics, International Journal for Numerical Methods in Fluids, 60, 391–408, https://doi.org/10.1002/fld.1896, 2009.

Casulli, V.: A coupled surface-subsurface model for hydrostatic flows under saturated and variably saturated conditions, International Journal for Numerical Methods in Fluids, 85, 449–464, https://doi.org/10.1002/fld.4389, 2017.

Casulli, V. and Zanolli, P.: High resolution methods for multidimensional advection–diffusion problems in free-surface hydrodynamics, Ocean Modelling, 10, 137–151, https://doi.org/10.1016/j.ocemod.2004.06.007, 2005.

Casulli, V. and Zanolli, P.: A nested Newton-type algorithm for finite volume methods solving Richards' equation in mixed form, SIAM Journal on Scientific Computing, 32, 2255–2273, https://doi.org/10.1137/100786320, 2010.

Casulli, V. and Zanolli, P.: Iterative solutions of mildly nonlinear systems, Journal of Computational and Applied Mathematics, 236, 3937–3947, https://doi.org/10.1016/j.cam.2012.02.042, 2012.

Celia, M. A., Bouloutas, E. T., and Zarba, R. L.: A general mass-conservative numerical solution for the unsaturated flow equation, Water Resources Research, 26, 1483–1496, https://doi.org/10.1029/WR026i007p01483, 1990.

Chistyakov, V.: On mappings of bounded variation, Journal of Dynamical and Control Systems, 3, 261, https://doi.org/10.1007/BF02465896, 1997.

Clark, M. P., Fan, Y., Lawrence, D. M., Adam, J. C., Bolster, D., Gochis, D. J., Hooper, R. P., Kumar, M., Leung, L. R., Mackay, D. S., et al.: Improving the representation of hydrologic processes in Earth System Models, Water Resources Research, 51, 5929–5956, https://doi.org/10.1002/2015WR017096, 2015a.

Clark, M. P., Nijssen, B., Lundquist, J. D., Kavetski, D., Rupp, D. E., Woods, R. A., Freer, J. E., Gutmann, E. D., Wood, A. W., Brekke, L. D., et al.: A unified approach for process-based hydrologic modeling: 1. Modeling concept, Water Resources Research, 51, 2498–2514, https://doi.org/10.1002/2015WR017198, 2015b.

Clark, M. P., Zolfaghari, R., Green, K. R., Trim, S., Knoben, W. J. M., Bennet, A., and Nijssen, B.: The numerical implementation of land models: problem formulation and laugh tests, Journal of Hydrometeorology - Preprint, Preprint, 2498–2514, https://doi.org/10.1175/JHM-D-20-0175.1, 2021.

Constantz, J. and Murphy, F.: The temperature dependence of ponded infiltration under isothermal conditions, Journal of Hydrology, 122, 119–128, https://doi.org/10.1016/0022-1694(91)90175-H, 1991.

Dai, Y., Wei, N., Yuan, H., Zhang, S., Shangguan, W., Liu, S., Lu, X., and Xin, Y.: Evaluation of soil thermal conductivity schemes for use in land surface modeling, Journal of Advances in Modeling Earth Systems, 11, 3454–3473, https://doi.org/10.1029/2019MS001723, 2019.

Dall'Amico, M., Endrizzi, S., Gruber, S., and Rigon, R.: A robust and energy-conserving model of freezing variably-saturated soil, The Cryosphere, 5, 469–484, https://doi.org/10.5194/tc-5-469-2011, 2011, 2011.

D'Amato, C.: https://github.com/geoframecomponents/BrokerGEO, last accessed: 20 April 2021.

D'Amato, C., Tubini, N., Benettin, P., Rinaldo, A., Noto, V., and Rigon, R.: Investigation of a critical zone column with the GEOframe Plant Atmosphere Continuum Estimator (GEO-SPACE), in preparation for Hydrology and Earth System Sciences, 2022.

David, O.: https://alm.engr.colostate.edu/cb/wiki/16970, last accessed: 1 April 2021.

David, O., Ascough II, J. C., Lloyd, W., Green, T. R., Rojas, K., Leavesley, G. H., and Ahuja, L. R.: A software engineering perspective on environmental modeling framework design: The Object Modeling System, Environmental Modelling & Software, 39, 201–213, https://doi.org/10.1016/j.envsoft.2012.03.006, 2013.

Di Nucci, C.: Theoretical derivation of the conservation equations for single phase flow in porous media: a continuum approach, Meccanica, 49, 2829–2838, https://doi.org/10.1007/s11012-014-0022-y, 2014.

Dong, Y., McCartney, J. S., and Lu, N.: Critical review of thermal conductivity models for unsaturated soils, Geotechnical and Geological Engineering, 33, 207–221, https://doi.org/10.1007/s10706-015-9843-2, 2015.

Dunne, T. and Black, R. D.: An experimental investigation of runoff production in permeable soils, Water Resources Research, 6, 478–490, https://doi.org/10.1029/WR006i002p00478, 1970.

Eckel, B.: Thinking in JAVA, Prentice Hall Professional, 2003.

Eisenberg, D., Kauzmann, W., and Kauzmann, W.: The structure and properties of water, Oxford University Press on Demand, 2005.

Endrizzi, S., Gruber, S., Dall'Amico, M., and Rigon, R.: GEOtop 2.0: simulating the combined energy and water balance at and below the land surface accounting for soil freezing, snow cover and terrain effects, Geoscientific Model Development, 7, 2831–2857, https://doi.org/10.5194/gmd-7-2831-2014, 2014.

Engeler, I., Franssen, H. H., Müller, R., and Stauffer, F.: The importance of coupled modelling of variably saturated groundwater flow-heat transport for assessing river–aquifer interactions, Journal of Hydrology, 397, 295–305, https://doi.org/10.1016/j.jhydrol.2010.12.007, 2011.

Farthing, M. W. and Ogden, F. L.: Numerical solution of Richards' equation: A review of advances and challenges, Soil Science Society of America Journal, 81, 1257–1269, https://doi.org/10.2136/sssaj2017.02.0058, 2017.

Formetta, G., Rigon, R., Chávez, J., and David, O.: Modeling shortwave solar radiation using the JGrass-NewAge system, Geoscientific Model Development, 6, 915–928, https://doi.org/10.5194/gmd-6-915-2013, 2013.

Formetta, G., Antonello, A., Franceschi, S., David, O., and Rigon, R.: Hydrological modelling with components: A GIS-based open-source framework, Environmental Modelling & Software, 55, 190–200, https://doi.org/10.1016/j.envsoft.2014.01.019, 2014a.

Formetta, G., Kampf, S. K., David, O., and Rigon, R.: Snow water equivalent modeling components in NewAge-JGrass, Geoscientific Model Development, 7, 725–736, https://doi.org/10.5194/gmd-7-725-2014, 2014b.

Formetta, G., Bancheri, M., David, O., and Rigon, R.: Performance of site-specific parameterizations of longwave radiation, Hydrology and Earth System Sciences, 20, 4641–4654, https://doi.org/10.5194/hess-20-4641-2016, 2016.

Forums, P. P. D.: https://www.pc-progress.com/forum/viewtopic.php?f=3&t=3632, last accessed: 20 April 2021.

Frampton, A., Painter, S. L., and Destouni, G.: Permafrost degradation and subsurface-flow changes caused by surface warming trends, Hydrogeology Journal, 21, 271–280, https://doi.org/10.1007/s10040-012-0938-z, 2013.

Freeman, E., Robson, E., Bates, B., and Sierra, K.: Head first design patterns, " O'Reilly Media, Inc.", 2008.

Furman, A.: Modeling coupled surface–subsurface flow processes: A review, Vadose Zone Journal, 7, 741–756, https://doi.org/10.2136/vzj2007.0065, 2008.

Gamma, E., Helm, R., Johnson, R., Vlissides, J., and Patterns, D.: Elements of Reusable Object-Oriented Software, Design Patterns. massachusetts: Addison-Wesley Publishing Company, 1995.

Germann, P. and Beven, K.: Water flow in soil macropores I. An experimental approach, Journal of Soil Science, 32, 1–13, https://doi.org/10.1111/j.1365-2389.1981.tb01681.x, 1981.

Greenspan, D. and Casulli, V.: Numerical Analysis for Applied Mathematics, Science, and Engineering, Addison Wesley, https://doi.org/https://doi.org/10.1201/9780429493393, 1988.

Gugole, F., Dumbser, M., and Stelling, G.: An efficient three-dimensional semi-implicit finite volume scheme for the solution of coupled free-surface and variably saturated sub-surface flow, in: EGU General Assembly Conference Abstracts, p. 600, 2018.

Hall, C. A., Saia, S. M., Popp, A. L., Dogulu, N., Schymanski, S. J., Drost, N., van Emmerik, T., and Hut, R.: A Hydrologist's Guide to Open Science, Hydrology and Earth System Sciences Discussions, pp. 1–23, https://doi.org/10.5194/hess-2021-392, 2021.

Hay, L. E., Leavesley, G. H., Clark, M. P., Markstrom, S. L., Viger, R. J., and Umemoto, M.: Step wise, multiple objective calibration of a hydrologic model for a snowmelt dominated basin 1, JAWRA Journal of the American Water Resources Association, 42, 877–890, https://doi.org/10.1111/j.1752-1688.2006.tb04501.x, 2006.

Horton, R. E.: The role of infiltration in the hydrologic cycle, Eos, Transactions American Geophysical Union, 14, 446–460, https://doi.org/10.1029/TR014i001p00446, 1933.

Hrachowitz, M., Benettin, P., Van Breukelen, B. M., Fovet, O., Howden, N. J., Ruiz, L., Van Der Velde, Y., and Wade, A. J.: Transit times—The link between hydrology and water quality at the catchment scale, Wiley Interdisciplinary Reviews: Water, 3, 629–657, https://doi.org/https://doi.org/10.1002/wat2.1155, 2016.

IDEAS: Methodologies and "How To", https://ideas-productivity.org/ideas-classic/how-to/, last accessed: 20 October 2021.

Jones, J. E. and Woodward, C. S.: Newton–Krylov-multigrid solvers for large-scale, highly heterogeneous, variably saturated flow problems, Advances in Water Resources, 24, 763–774, https://doi.org/10.1016/S0309-1708(00)00075-0, 2001.

Kelley, C. T.: Solving nonlinear equations with Newton's method, SIAM, 2003.

Kennedy, J. and Eberhart, R.: Particle swarm optimization, in: Proceedings of ICNN'95-international conference on neural networks, vol. 4, pp. 1942–1948, IEEE, 1995.

Kollet, S. J. and Maxwell, R. M.: Integrated surface–groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model, Advances in Water Resources, 29, 945–958, https://doi.org/10.1016/j.advwatres.2005.08.006, 2006.

Kosugi, K.: General model for unsaturated hydraulic conductivity for soils with lognormal pore-size distribution, Soil Science Society of America Journal, 63, 270–277, https://doi.org/10.2136/sssaj1999.03615995006300020003x, 1999.

Kurylyk, B. L. and Watanabe, K.: The mathematical representation of freezing and thawing processes in variably-saturated, non-deformable soils, Advances in Water Resources, 60, 160–177, https://doi.org/10.1016/j.advwatres.2013.07.016, 2013.

Liu, S., Lu, L., Mao, D., and Jia, L.: Evaluating parameterizations of aerodynamic resistance to heat transfer using field measurements, Hydrology and earth system sciences, 11, 769–783, https://doi.org/10.5194/hess-11-769-2007, 2007.

Lloyd, W., David, O., Ascough II, J. C., Rojas, K. W., Carlson, J. R., Leavesley, G. H., Krause, P., Green, T. R., and Ahuja, L. R.: Environmental modeling framework invasiveness: Analysis and implications, Environmental modelling & software, 26, 1240–1250, https://doi.org/10.1016/j.envsoft.2011.03.011, 2011.

Lu, N.: Generalized soil water retention equation for adsorption and capillarity, Journal of Geotechnical and Geoenvironmental Engineering, 142, 04016 051, https://doi.org/10.1061/(ASCE)GT.1943-5606.0001524, 2016.

Martin, R. C.: Clean code: a handbook of agile software craftsmanship, Pearson Education, 2009.

Mcdonough, J. M.: Introductory lectures on turbulence physics, mathematics and modeling, Tech. rep., University of Kentucky, https://uknowledge.uky.edu/me_textbooks/2, 2004.

Menard, C. B., Essery, R., Krinner, G., Arduini, G., Bartlett, P., Boone, A., Brutel-Vuilmet, C., Burke, E., Cuntz, M., Dai, Y., et al.: Scientific and human errors in a snow model intercomparison, Bulletin of the American Meteorological Society, 102, E61–E79, https://doi.org/10.1175/BAMS-D-19-0329.1, 2020.

Mualem, Y.: A new model for predicting the hydraulic conductivity of unsaturated porous media, Water resources research, 12, 513–522, https://doi.org/10.1029/WR012i003p00513, 1976.

Muskat, M. and Meres, M. W.: The flow of heterogeneous fluids through porous media, Physics, 7, 346–363, https://doi.org/10.1063/1.1745403, 1936.

National Research Council: Basic research opportunities in earth science, https://doi.org/10.17226/9981., 2001.

Newman, S.: Building Microservices: Designing Fine-Grained Systems, 2015.

Nicolsky, D. and Romanovsky, V. E.: Modeling long-term permafrost degradation, Journal of Geophysical Research: Earth Surface, 123, 1756–1771, https://doi.org/10.1029/2018JF004655, 2018.

Nobel, P. S. et al.: Physicochemical & environmental plant physiology, Academic Press, 1999.

Ochsner, T. E., Horton, R., and Ren, T.: A new perspective on soil thermal properties, Soil science society of America Journal, 65, 1641–1647, https://doi.org/10.2136/sssaj2001.1641, 2001.

Paniconi, C. and Putti, M.: A comparison of Picard and Newton iteration in the numerical solution of multidimensional variably saturated flow problems, Water Resources Research, 30, 3357–3374, https://doi.org/10.1029/94WR02046, 1994.

Paniconi, C. and Putti, M.: Physically based modeling in catchment hydrology at 50: Survey and outlook, Water Resources Research, 51, 7090–7129, https://doi.org/10.1002/2015WR017780, 2015.

Paniconi, C. and Wood, E. F.: A detailed model for simulation of catchment scale subsurface hydrologic processes, Water Resources Research, 29, 1601–1620, https://doi.org/10.1029/92WR02333, 1993.

Peckham, S. D., Hutton, E. W., and Norris, B.: A component-based approach to integrated modeling in the geosciences: The design of CSDMS, Computers & Geosciences, 53, 3–12, https://doi.org/10.1016/j.cageo.2012.04.002, 2013.

Penna, D., Hopp, L., Scandellari, F., Allen, S. T., Benettin, P., Beyer, M., Geris, J., Klaus, J., Marshall, J. D., Schwendenmann, L., et al.: Ideas and perspectives: Tracing terrestrial ecosystem water fluxes using hydrogen and oxygen stable isotopes–challenges and opportunities from an interdisciplinary perspective, Biogeosciences, 15, 6399–6415, https://doi.org/10.5194/bg-15-6399-2018, 2018.

Priestley, C. H. B. and Taylor, R. J.: On the assessment of surface heat flux and evaporation using large-scale parameters, Monthly weather review, 100, 81–92, https://doi.org/10.1175/1520-0493(1972)100<0081:OTAOSH>2.3.CO;2, 1972.

Quarteroni, A., Sacco, R., and Saleri, F.: Numerical mathematics, vol. 37, Springer Science & Business Media, 2010.

Raupach, M. and Thom, A. S.: Turbulence in and above plant canopies, Annual Review of Fluid Mechanics, 13, 97–129, https://doi.org/10.1146/annurev.fl.13.010181.000525, 1981.

Regenass, D., Schlemmer, L., Jahr, E., and Schär, C.: It rains and then? Numerical challenges with the 1D Richards equation in kilometer-resolution land surface modelling, Hydrology and Earth System Sciences Discussions, pp. 1–33, https://doi.org/10.5194/hess-2021-426, 2021.

Richards, L. A.: Capillary conduction of liquids through porous mediums, Physics, 1, 318–333, https://doi.org/10.1063/1.1745010, 1931.

Richardson, L. F.: Weather prediction by numerical process, Cambridge University Press, 1922.

Rigon, R., Bertoldi, G., and Over, T. M.: GEOtop: A distributed hydrological model with coupled water and energy budgets, Journal of Hydrometeorology, 7, 371–388, https://doi.org/doi.org/10.1175/JHM497.1, 2006a.

Rigon, R., Ghesla, E., Tiso, C., and Cozzini, A.: The HORTON machine: a system for DEM analysis The reference manual., Tech. rep., University of Trento, http://hdl.handle.net/11572/48576, 2006b.

Rigon, R., Bancheri, M., Formetta, G., and de Lavenne, A.: The geomorphological unit hydrograph from a historical-critical perspective, Earth Surface Processes and Landforms, 41, 27–37, https://doi.org/10.1002/esp.3855, 2016a.

Rigon, R., Bancheri, M., and Green, T. R.: Age-ranked hydrological budgets and a travel time description of catchment hydrology, Hydrology and Earth System Sciences, 20, 4929–4947, https://doi.org/10.5194/hess-20-4929-2016, 2016b.

Rizzoli, A., Svensson, M., Rowe, E., Donatelli, M., Muetzelfeldt, R., van der Wal, T., van Evert, F., and Villa, F.: Modelling framework (SeamFrame) requirements, Tech. rep., SEAMLESS, 2006.

Roe, P. L.: Approximate Riemann solvers, parameter vectors, and difference schemes, Journal of Computational Physics, 43, 357–372, https://doi.org/10.1016/0021-9991(81)90128-5, 1981.

Romano, N., Brunone, B., and Santini, A.: Numerical analysis of one-dimensional unsaturated flow in layered soils, Advances in Water Resources, 21, 315–324, https://doi.org/10.1016/S0309-1708(96)00059-0, 1998.

Ronan, A. D., Prudic, D. E., Thodal, C. E., and Constantz, J.: Field study and simulation of diurnal temperature effects on infiltration and variably saturated flow beneath an ephemeral stream, Water resources research, 34, 2137–2153, https://doi.org/10.1029/98WR01572, 1998.

Saito, H., Šimůnek, J., and Mohanty, B. P.: Numerical analysis of coupled water, vapor, and heat transport in the vadose zone, Vadose Zone Journal, 5, 784–800, https://doi.org/10.2136/vzj2006.0007, 2006.

Serafin, F.: Enabling modeling framework with surrogate modeling capabilities and complex networks, Ph.D. thesis, University of Trento, 2019.

Shewchuk, J. R. et al.: An introduction to the conjugate gradient method without the agonizing pain, 1994.

Sophocleous, M.: Analysis of water and heat flow in unsaturated-saturated porous media, Water Resources Research, 15, 1195–1206, https://doi.org/10.1029/WR015I005P01195, 1979.

Srivastava, R. and Yeh, T.-C. J.: Analytical solutions for one-dimensional, transient infiltration toward the water table in homogeneous and layered soils, Water Resources Research, 27, 753–762, https://doi.org/10.1029/90WR02772, 1991.

Tomin, P. and Lunati, I.: Investigating Darcy-scale assumptions by means of a multiphysics algorithm, Advances in water resources,

95, 80–91, https://doi.org/10.1016/j.advwatres.2015.121013, 2016.

Tubini, N. and Rigon, R.: https://pypi.org/project/geoframepy/, last accessed: 22 October 2021, 2021.

Tubini, N., Gruber, S., and Rigon, R.: A method for solving heat transfer with phase change in ice or soil that allows for large time steps while guaranteeing energy conservation, The Cryosphere Discussions, 15, 1–41, https://doi.org/10.5194/tc-15-2541-2021, 2020.

Unidata: Chunking Data with NetCDF-4, https://www.unidata.ucar.edu/software/netcdf/workshops/2008/nc4chunking/index.html, last accessed: 1 April 2021, a.

Unidata: Formats and Performance, https://www.unidata.ucar.edu/software/netcdf/workshops/2008/performance/index.html, last accessed: 1 April 2021, b.

Unidata: NetCDF version 4.3.22, Boulder, CO: UCAR/Unidata Program Center, https://doi.org/https://doi.org/10.5065/D6H70CW6, 2021.

Van Genuchten, M. T.: A closed-form equation for predicting the hydraulic conductivity of unsaturated soils, Soil science society of America journal, 44, 892–898, https://doi.org/10.2136/sssaj1980.03615995004400050002x, 1980.

Vanderborght, J., Kasteel, R., Herbst, M., Javaux, M., Thiéry, D., Vanclooster, M., Mouvet, C., and Vereecken, H.: A set of analytical benchmarks to test numerical models of flow and transport in soils, Vadose Zone Journal, 4, 206–221, https://doi.org/10.2113/4.1.206, 2005.

Šimůnek, J., Van Genuchten, M. T., and Sejna, M.: The HYDRUS-1D software package for simulating the one-dimensional movement of water, heat, and multiple solutes in variably-saturated media, University of California-Riverside Research Reports, 3, 1–240, 2005.

Šimůnek, J., Van Genuchten, M. T., and Sejna, M.: The HYDRUS software package for simulating the two-and three-dimensional movement of water, heat, and multiple solutes in variably-saturated porous media, Technical manual, 2012.

Walvoord, M. A. and Kurylyk, B. L.: Hydrologic impacts of thawing permafrost—A review, Vadose Zone Journal, 15, https://doi.org/10.2136/vzj2016.01.0010, 2016.

Walvoord, M. A., Voss, C. I., and Wellman, T. P.: Influence of permafrost distribution on groundwater flow in the context of climate-driven permafrost thaw: Example from Yukon Flats Basin, Alaska, United States, Water Resources Research, 48, https://doi.org/10.1029/2011WR011595, 2012.

Whitaker, S.: Flow in porous media I: A theoretical derivation of Darcy's law, Transport in porous media, 1, 3–25, https://doi.org/10.1007/BF01036523, 1986.

Wierenga, P., Hagan, R. M., and Nielsen, D.: Soil temperature profiles during infiltration and redistribution of cool and warm irrigation water, Water Resources Research, 6, 230–238, https://doi.org/10.1029/WR006i001p00230, 1970.

Yang, Y., Wu, J., Zhao, S., Han, Q., Pan, X., He, F., and Chen, C.: Assessment of the responses of soil pore properties to combined soil structure amendments using X-ray computed tomography, Scientific reports, 8, 695–705, https://doi.org/10.1038/s41598-017-18997-1, 2018.

Zha, Y., Yang, J., Zeng, J., Tso, C.-H. M., Zeng, W., and Shi, L.: Review of numerical solution of Richardson–Richards equation for variably saturated flow in soils, Wiley Interdisciplinary Reviews: Water, 6, e1364, https://doi.org/10.1002/wat2.1364, 2019.

Zhang, S., Meurey, C., and Calvet, J.-C.: Identification of soil-cooling rains in southern France from soil temperature and soil moisture observations, Atmospheric Chemistry and Physics, 19, 5005–5020, https://doi.org/10.5194/acp-19-5005-2019, 2019.