Discontinuous Galerkin models for coupled surface-subsurface flow

Thomas De Maet

Thèse présentée en vue de l'obtention
du grade de docteur en sciences agronomiques
et ingénierie biologique

Thesis committee:

Dr. Philippe Ackerer	LHyGeS, Strasbourg, France
Prof. Claude Bragard	Université catholique de Louvain
Prof. Emmanuel Hanert (Supervisor)	Université catholique de Louvain
Dr. Jonathan Lambrechts	Université catholique de Louvain
Prof. Vincent Legat	Université catholique de Louvain
Prof. Marnick Vanclooster	Université catholique de Louvain

Louvain-la-Neuve
November 2018

Certains diront qu'il était grand temps, d'autres pensaient que ça n'arriverait plus, mais nous y voici: le document est finalisé et la défense inscrite à l'agenda ! Cette thèse qui dura 5 ans dans mon esprit mais 10 pour le reste du monde n'a pas été réalisée sans l'appui de beaucoup de personnes, que je souhaite vivement remercier.

Premièrement je suis fortement reconnaissant envers mon promoteur, Emmanuel Hanert, qui a passé beaucoup de son temps à m'écouter projeter des idées parfois farfelues et à corriger mon anglais fortement francisé. Sa disponibilité était sans faille. Au vu des prolongations, je suppose qu'il est soulagé de voir le premier doctorant à avoir commencé avec lui enfin clôturer !

J'aimerais remercier les membres de mon jury pour leur relecture constructive de ce document très technique, et en particulier Philippe Ackerer qui aura fait le déplacement deux fois depuis Strasbourg. Je dois également beaucoup à Cécile Dagès pour ses conseils directeurs lors d'un séjour à Montpellier. Merci à mes co-auteurs pour le travail fourni dans les deux papiers publiés, ainsi qu'aux relecteurs anonymes, essentiels à une méthode scientifique impartiale.

Le cadre de travail est très important pour l'inter-motivation et le partage des idées. Un grand merci à mes collègues de bureau pour les très bons moments passés ensemble, et en particulier à ceux qui y sont restés plus longtemps que d'autres: Sarah, Jolan et Antoine. Merci au reste du groupe ENGE pour toutes nos discussions autour de bonnes tartes à la pause, et d'avoir pensé aux buveurs de thé. Merci également à l'équipe de MEMA pour son accueil lors de mes brèves incursions en Euler. Un merci spécial pour Jonathan, le grand guru du framework SLIM sur lequel cette thèse est basée, lui qui a toujours su me sortir de mes problèmes techniques en un tournemain.

Je remercie l'UCL de m'avoir octroyé une bourse FSR de 4 ans, ce qui n'est pas commun. Près de la fin du temps imparti par cette bourse, la thèse n'en était pas à sa conclusion. Je remercie Pierre Defourny pour m'avoir engagé en tant qu'assistant de recherche à $4/5^{\text{e}}$ temps dans son équipe, travail que je continue d'ailleurs à ce jour avec motivation. Merci également de m'avoir permis d'aménager le 5^{e} restant pour avancer efficacement dans cette thèse.

Malgré toutes ces aides et collaborations, une thèse est un long travail qui se fait principalement en solitaire. Il est dès lors important de se ménager d'autres activités et distractions pour conserver l'optimisme de départ.

Un merci particulier à ma femme, Elisabeth, et à mes deux filles, Yara et Iris, pour m'avoir rappelé à chaque instant passé à la maison qu'il y avait autre chose dans la vie que le travail. Merci à mes compagnons de grimpe pour les temps de midi passés sur les murs du Blocry. Merci aux membres du Sakura dojo pour leur pratique de l'aïkido, régénérante pour le corps et l'esprit. Merci à tous mes amis pour les sorties nature, les repas autour d'un bon verre et les soirées jeux. Enfin, merci à toute ma famille de m'avoir donné l'opportunité d'arriver jusqu'ici. Une pensée émue pour ma grand-mère qui nous a quitté trop tôt et aurait aimé tenir ses petits enfants sur ses genoux, et peut-être, qui sait, glisser ce manuscrit dans sa bibliothèque...

Contents

List of Abbreviations

CPU Central Processing Unit.

CVFE Control Volume Finite Elements.

DG Discontinuous Galerkin.

DOF Degree Of Freedom.

DZ Dry Zone (UZ far from saturation).

FE Finite Elements.

FOEC First Order Echange Coefficient (coupling method).

HPC High Performance Computer.

NI Non-inertia (approximation of the SWE).

OS Operating System.

PC Pressure Continuity (coupling method).

RAM Random Access Memory.

RE Richards Equation.

SWE Shallow Water Equations.

SZ Saturated Zone.

UZ Unsaturated Zone.

WZ Wet Zone (UZ near saturation).

CHAPTER 1

Introduction

1.1 Thesis overview

Throughout History, mankind has always tried to understand its environment. Its quest of knownledge lead to the discovery of sciences and mathematics, deriving rules from observations to predict the behavior of its surronding. For instance, equations were developped for material deformation, air displacement or water flow. However, a lot of frustration came from that those beautiful rules (or models) were too complex to describe the world without a lot of by-hand calculation. *De-facto*, scientists were limited to very simple cases with strong mathematical simplifications. Eventually, the Digital revolution came as a game-changer for modelers.

The real world shows many very complex shapes and patterns difficult to represent with mathematics. An intuitive way is to break down those patterns. If we cut the problem into a large number of very small parts, what is observed in each of those parts becomes more simple. That idea is mathematically translated by discretizations, often in space and in time, *i.e.* a split of the domain of interest into sub-domains and the split of the time into a succession of static states. It is easier to solve those smaller and isolated problems, only influenced by their direct neighborhood. But then a huge number of problems have to be solved. That fits well with computers, where everything has to be discretized

(all is binary, nothing really continuous) and for which performing many basic calculations is not an issue.

The conceptualization of physical processes are translated into models, mathematical representations constituted by one or several equations. The most general forms of physical equations (*e.g.* Maxwell for electro-magnetism, Schrödinger for atoms, Navier-Stokes for fluids, etc.) are rarely used as such as they quickly lead to problems too complex to solve. They are thus simplified under some hypothesis, considered reasonable for a specific range of applications.

In this thesis, we focus on terrestrial surface and subsurface water behavior. It is possible to model all the water cycle via Navier-Stokes, but doing so include behaviors like ripples of raindrops on water surface and all the small vortices that it produces. This level of detail is not very interesting at the scale of a landscape... If we keep the rain example, what is expected of a model is its consequences (how the soil is wetted, if a lake is refilled, etc.), for which approximated forms of Navier-Stokes are sufficient. In this very specific case the rain will often be radically converted into a simple flux per surface area.

Following that idea of simplification, the water cycle is often broken down into more reasonable problems, considered of sufficient precision for the application. The large rivers, seas and oceans can often be considered to be of small depth with regard to their horizontal extent, leading to the 2D shallow waters equations (SWE). The small rivers and ditches can even be reduced to a 1D form. For surface runoff, an additional simplification is often applied, like the non-inertia (NI) approximation. For global atmospheric flows, the 3D nonhydrostatic equations are often used. For subsurface flows, Darcy's law can approximate fluxes through a porous medium. Its conservation form commonly called the Richards equation. This approximation can be further simplified into a 2D groundwater flow equation for saturated areas. But all those classical examples represent only a sample among many other possibilities.

In this plethora of models, one have to find the right tool for its application: soil pollution studies need to know the soil water fluxes over a long period of time; dredgers have an interest in where alluvium will accumulates in estuaries; city architects want to know the consequences of the landscape waterproofing in case of heavy rain events, civil engineers want to assess the consequences of a potential dam break, etc. With a lot of hypothesis, specialized models can produce exactly what is asked, but are often of little use for other cases and frequently rely on user-defined fitting parameters. General purpose models in contrast often require a heavier workload.

Ideally, the whole water cycle should be modeled by coupling together all its simplified components (rain, river, groundwater, etc.), in a continuous way. That continuity ensures that the fluxes at the components interfaces do not need additional assumptions. Rain is easy to measure, but what about assessing

the fluxes between a lake and its surrounding soil? The coupling of general purpose components (sub-models), with the right balance between precision and number of parameters, with additional precision over the study area seems to be the right approach. However, it is still complex to set up as it requires a lot of input data for parametrization, initial and boundary conditions. In addition, even with modern computers it is still computationally expensive.

This work had initially the goal to link the surface and subsurface components of the terrestrial water system to larger water bodies like estuaries, in order to build a continuous and comprehensive model of the the land-sea continuum, with meaningful parameters. However it has been quickly apparent that despite the large amount of work on the subject, the terrestrial models were not totally ready for such a scale. The main reasons are the huge computational cost of continuous 3D subsurface flow models and the difficulty to scale the soils parametrization from the lab sample to the landscape. Although the latter is a very interesting topic, here we focus on the former and try to reduce the simulation time. Indeed, being able to launch a simulation over a large domain and/or over a long period of time is critical to achieve our goal. To improve the terrestrial component, both surface and subsurface components have to be optimized, along with their coupling.

The Richards Equation (RE) models the water fluxes in both saturated-in-water and unsaturated areas of the soil. The RE might look simple but its constitutive relationships, highly non-linear, lead to strong numerical issues. Indeed, its behavior changes from parabolic in steady unsaturated areas (slow, diffusive), to elliptic in saturated areas (incompressible), passing by an advective behavior when infiltration fronts are present (fast displacements). In consequence, numerical methods able to solve the RE are often basic, resorting to simplifications to increase their robustness. Until recently, their development have not been designed from the start for parallel computing architectures.

In this work, three hydrological models were developed, efficiently using the computational resources with a focus on parallelization: two models for the RE and one coupling the NI SWE to the RE. Those models were designed to be as robust as possible without losing accuracy.

In this chapter we briefly remind the reader of the concepts of hydrology, modeling and computer science useful to understand this document. We then describe the fixed objectives and provide a short summary of this thesis. More specific information is provided in the introduction of each chapter. Chapters 3 and 5 are about RE models, Chapter 4 is about the NI SWE coupled with the RE. Chapter 2 gathers some hints for modelers accumulated during this work. Finally, global conclusion on this work are presented in Chapter 6.

1.2 Hydrological models: from the physics to the servers

1.2.1 Physical processes

Out of scientific curiosity, one could wonder why studying the water cycle ?

As everybody knows, water is the essential ingredient of terrestrial life. It is a simple and light molecule and as a consequence is frequent in the universe. Its chemical properties as a good solvent make it the ideal place to mix and assemble other molecules like DNA or proteins. It can also carry many other materials like alluvions, pollutants, bacteria, hydrogen ions (acid rains), minerals, metals, etc. It fills lakes and oceans but also erodes the mountains, drives inundations or mudslide. It is present in the entire biosphere, running through whole landscapes, sometimes scratching and accumulating what is in its reach, sometimes filtered by soil, evaporation or life. As all organisms – including the human – are included in this cycle, many contaminants eventually pass through them, often not without consequences.

The water stock management is of first importance in many countries (*e.g.* the annual flooding of the Nile, irrigation around the Aral sea, or simply water supply during a dry year). It is also important for town and country planning to identify areas prone to flooding, which are more and more exacerbated by the human waterproofing of landscapes and river channeling.

More recently, with the increasing human pressure and the rise of living standards, many officials are asking for more information about the sources and deposition areas of pollutants, and their impact. The human health is at the center of those preoccupations, but every part of the biosphere is endangered, which nowadays leads to a dramatic human-driven reduction of the biodiversity in the world. Modeling such fluxes is very challenging as we are often dealing with small quantities (drugs, pesticides, metals, etc.) that are difficult to measure and to track.

The natural processes linked to water displacement are also interesting subjects of study. Soil erosion, riverbed shifts, sand bar creation or aquatic fauna migration are some of them. The human has frequent interactions with them: barrages blocking fishes migration, channeling or dredging modifying the previous steady flow of sediment and causing new issues downstream, diverting a watercourse causing desertification, forcing a river to keep its track involving no new backwater areas interesting for wildlife, etc. It is also interesting to know how those processes naturally evolve in time and modify their surrounding. This is interesting to study the history of a landscape on Earth, or even on Mars!

In this document, we focus on three parts of the water cycle: below the water table (which is saturated by water), the vadose zone (unsaturated soil) and the

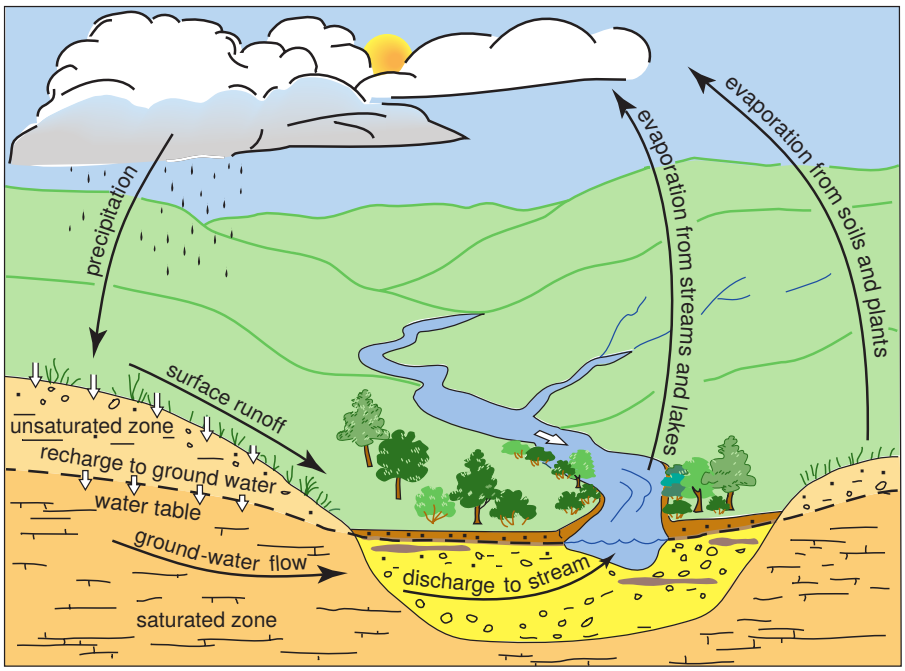


Figure 1.1: Interactions between surface water, groundwaters and atmosphere. Adapted from the Public Information Circular 23 of Kansas Geological Survey.

surface water (runoff, lakes, etc.). The Fig. 1.1 sketch up the main interactions between them.

Groundwater bodies are an important source of drinking water of quality as they are naturally filtered by the soil from contaminants. The intensification of its extraction for growing cities or crop irrigation currently raises a bunch of concerns. The recharge of the aquifers is a slow process which needs a frequent and moderate input of water. Indeed, huge quantities over short periods create mostly runoff and as a consequence only a small fraction infiltrates the soil. The aquifer recharge is thus efficient under temperate or tropical climates. However, freshwater problems can appear anywhere. In the one hand, even in favorable areas a dry year can lead to consumption restrictions. Such years can appear more frequently with climate deregulation. In the other hand, the raise of agricultural activity in arid countries over poorly recharged aquifers or even fossil water is concerning. Another important issue is the evolution of the water quality. It can be slowly tainted by agriculture, industry or garbage

contaminants that eventually bypass or saturate the soil filtering capacity. It can also be victim of saltwater intrusion due to intensive pumping on the coasts.

The unsaturated area of the soil is important for many biogeochemical cycles: the carbon cycle, the nitrogen cycle, the oxygen cycle, the sulfur cycle and obviously the water cycle. It is essential for most of the biotopes as it is a place where micro- and macro-organisms, plants and various chemical components interact with each other thanks to the water present in the soil. In unsaturated areas the drivers for water flows are gravity and the capillary effect. The latter is produced by intermolecular forces between water and soil grains. A soil pore can then create a force inversely proportional to its radius. Small pores can then develop a long-range capillary suction. At rest, a gradient of soil moisture develops from the water table to the soil surface, balancing gravity and capillary forces. This profile is primarily influenced by evapotranspiration (soil evaporation and plant transpiration), water table rise/drop and rain events. It is also impacted by human activities. For instance, plowed fields often end with a compacted soil layer, while intensive irrigation can accumulate the salt naturally present in water at the soil surface.

The surface water includes streams and water bodies at small and medium scales, from the puddle to the small lake, including runoff and rivers. We exclude from that group the large water bodies like large lakes, seas or estuaries as their dynamics are more complex. The rain water stay at surface in two cases: either it goes straight to an area already full of water (as lake or filled aquifer) or its flux is too important for the unsaturated soil conductivity (with impervious human settlements as an extreme case).

Those three parts of the water cycle are intimately linked. The aquifer recharge and the runoff generation are both driven by the unsaturated zone state and properties. Permanent surface water bodies are strongly connected with their surrounding water table. The soil can belong either to the saturated or to the unsaturated zone, depending on the current water table level. Those parts are multi-scale. Indeed, their behavior at micro-scale is very heterogeneous and has a direct impact at macro-scale: small depressions at the surface delay global runoff during a rainfall and thus increase infiltration in the soil; a single continuous soil fracture can dramatically change the behavior of water recharge, bypassing the unsaturated area; a thin lens of clay can isolate the area below it from incoming water, diverting it away.

In this work, the water in saturated and unsaturated areas is described by the Richards equation, the surface water by the NI approximation of the SWE. Chapter 4 is dedicated to the coupling of all those elements into one model.

1.2.2 Mathematical formulation

A mathematical model tries to reproduce the main processes observed in reality. As an example, "the quantity of sugar dissolved in your coffee is what was present before plus what you are adding" is a model. At the first sight it seems right, but if you add too much sugar, your coffee will get saturated and the sugar will precipitate. You can go further by telling that as your coffee is hot, water is evaporating quite quickly, increasing the sugar to water ratio then accelerating the precipitation effect. To cite George Box: "All models are wrong but some are useful". The original example can be used most of the time in a restricted context: we suppose that we want to drink coffee with sugar and not the opposite, and that the evaporation (or the drop in temperature) can be neglected in the time-frame of our coffee break.

The choice of the degree of precision for a model should depend on the application. If what we want to measure is not really impacted by an external process, we can neglect it or at least simplify it. In practice, many processes are neglected, either because they play a minor role, or because they would be too complex to model explicitly or because they cannot be easily measured. Indeed, the parametrization (assigning a value for each tunable parameter of the model) is hazardous when there are many parameters and few reliable measurements available.

In this work we focus on mechanistic models, which are based on physical laws. Such models fit a large range of problems, but can be very complex and difficult to parametrize. As mentioned before, physical laws described by equations can be discretized in order to reduce their complexity into a set of smaller problems. The spatial discretization consists in creating a partition of non-overlapping elements (points, triangles, cubes, etc.) inside which the variables describing the system are evaluated at specific locations (one value at the center, one value per vertex, etc.). That structure is called a mesh. The temporal discretization only considers precise instants in time, jumping from one to another like a succession of images in a film.

A simulation is the model virtually put into practice, in contrast with the experiment. The most used is the numerical simulation (simulation on a computer). The confrontation between simulation results and experimental measurements is the way to validate the physical model, its approximations, its discretization, its implementation (the code), its parametrization and the errors on the measures. Obviously, when there is a mismatch it is difficult to isolate what is the actual source of error.

With existing technologies, the measurement of soil properties, water content, fractures, etc. are mostly done at the surface, or along vertical transects. It is most of the time not sufficient as the soil is by nature very heterogeneous. Even laboratory measurements of soil properties are quite approximate (hysteresis

effect, small samples, degradation by the sample extraction, etc.). Compared to that noisy and sparse input data, the numerical errors of most of the models are often much smaller. And the question of the up-scaling is also still on the table: is what we observe in the lab similar to what is present at watershed scale? If not, how to model this macro-behavior?

Due to the lack of measurements, model validation is difficult and often based on simple water budgets (*e.g.* measures of rain and fluxes at the watershed outlet). Model parametrization is similarly impacted, especially in the estimation of the soil types repartition. To circumvent that aspect, it is often necessary to use a stochastic modeling approach to assess the impact of the variability of model parameters on the simulation results. However, those methods should be used with care as the number of degrees of freedom in model parametrization is often high, leading to over-parametrization (*i.e.* a set of parameters only valid for the input data) or allowing different set of parameters leading to good matches with measurements [1].

Numerical simulations are often distinguished based on their discretizations in space and time. There are many existing methods and room for many more. The most common spatial integration schemes are the finite differences, the finite volumes and the finite elements (FE). They respectively use one value per point, one value per volume or one function per volume. A variant between the last two is the discontinuous Galerkin (DG) FE method, for which each volume function can evolve independently of its neighbors. A 3D model often uses a mesh composed of around 10^5 to 10^9 elements. It can thus be very computationally expensive. For problems with some symmetries (cylinder, cube, etc.) or small aspect ratios, it is possible to reduce the dimensionality to 2D or 1D. Such symmetries are also often assumed on general cases as an approximation to run numerical simulation in reasonable times. Some models are restricted in their choice of mesh, which has to be structured (grid-shaped) or to belong to a restricted set of element types (triangles, quadrilaterals, prisms, etc.). Some elements types can be more appropriate than others, depending on the physics involved. An unstructured mesh can easily fit any geometrical configuration and can simultaneously represent processes at different scales. A multi-scale mesh typically uses small elements to focus on the area of interest and large elements elsewhere to reduce the computational cost.

The temporal integration schemes can be split into explicit and implicit. They differ in the way they handle information about the state of the system. The former extrapolates the state based only on previous states. The result can be easily determined but the time step is restricted for stability reasons. The latter uses information from both the previous and current states. But as the current state is not directly available, it is more difficult to solve, especially if the model is non-linear. Implicit schemes are known to be stable for larger time steps than explicit schemes but each iteration is much more expensive to solve.

As the computational time is the number of iterations (inversely proportional to the time step) times their duration, the optimal method is not obvious and application-dependent. The temporal integration schemes are also often classified by order of precision, reflecting achievable precision when refining the discretization.

We chose to use the RE to model the subsurface water flows as it is valid in both saturated and unsaturated areas with a simple formulation, and it is broadly used. Multi-phase (air, water, oil, etc.) models are based on fewer assumptions, and for instance 2-phases water-air equations can be used instead of RE. However this additional complexity is often not required for environmental flows, except in rare conditions such as a saturated water infiltration front reaching a water table.

Models based on RE equation mostly use first order spatial integration schemes and first order implicit time integration schemes. As explained before, this is mainly due to the difficulty to handle its inherent non-linearities: the variables can change quickly both in space and in time, in a way difficult to predict with any polynomial. An optimal method to solve that equation is still missing [2].

In this document, three numerical models are developed and validated both against other numerical models and measurements. All groundwaters models rely on 3D DG FE method, which is well adapted to advective behaviors. Most of the work is based on explicit time discretizations for its good scaling properties. Some popular approximations and methods of RE are described in Chapter 2.

1.2.3 Computational aspects

The transistors are at the basis of the computer science as we know it. Moore law states that their number on one chip (the main component of a processor) doubles every two years. A decade ago, after reaching quantum limitations in the race to miniaturization, the performances of individual microprocessors stuck, mostly in term of frequency. Instead of changing drastically from silicon chips to another technology, the choice was made to duplicate the processors either on the same chip (multi-core processors), on a common motherboard (several classical processors), with an extension card (Graphical Processing Units or manycore processors) or with other computers (parallel computing via a network). Technologies like vectorization or hyperthreading can further increase the number of tasks running in parallel. In this work we designate the smallest unit of process as a worker, as CPU (central processing unit) has become an ambiguous term with all these new technologies.

That increase in processing power must be handled with care to be efficient. Indeed, the computing workload is often not easily splittable evenly into a

set of tasks, which is required in parallel processing. If one task takes more time and the others wait for it, during the waiting time the whole processing power is reduced to one worker. The memory also has to be handled with care to be either shared to improve performances or isolated to avoid conflicts. Communication is a critical component as well, especially on high performance computing (HPC) centers using many servers connected through a network, which is often much slower than the internal bus speed in individual servers. All those factors can impact the parallel scaling, which is the performance gain when increasing the number of workers. A model with perfect scaling sees its performances double when the number of workers doubles.

In those new programming paradigms, most numerical surface-subsurface water models are still lagging behind. Since they have been developed for sequential workload, it is difficult to enhance their performances on parallel architectures. In this thesis, we focus on this aspect for RE, forcing a model to a perfect scaling in Chapter 3 but with degraded precision in saturated areas. In Chapter 5 a compromise is reached to get back all the precision at the cost of a small part of the scaling.

1.3 Scope of this thesis

In this introduction, we have highlighted the different roles of the water cycle and the human impact on it. As water is the main driver of many biogeochemical processes, modeling its behavior is the first step for more advanced models like those simulating the carbon cycle or pollutants dynamics. Since water dynamics spans many different scales in space and time, models have to focus on specific sub-dynamics like saturated/unsaturated groundwaters, shallow and deep waters, runoff, etc. Such models can be coupled together to form a macro-model of a larger part of the whole water cycle.

1.3.1 Objectives

The aim of this thesis is the development of models for the terrestrial component of the water cycle, to allow their inclusion in a greater multi-model framework[1]. We mostly focus on numerical aspects and computational efficiency. Our goal is to develop numerical methods to:

- model water flow in saturated soils, the primary and fastest underground water storage;
- model water flow in unsaturated soils, the main interface between the land surface and the water table and home of important ecosystems;

[1] The SLIM framework which already models rivers, large lakes, estuaries and seas. Website: www.slim-ocean.be

- model overland flow, including runoff and temporary water bodies appearing during strong rain events;
- connect those three elements into one single model and
- optimize the models numerical properties.

In this work, the water in saturated soils, unsaturated soils and at soil surface is simulated through two sub-models: the RE for the two formers, and the NI approximation of the SWE for the later. Those models are frequent choices.

The RE has been widely modeled for a long time, but rarely at large scale. The main reasons are the model convergence properties [3] and its computational cost. The convergence issue requires to soften the strong non-linearities of the equation, either directly through the constitutive relationships or through numerical approximations. In this work we aim to develop mass conservative models, which do not use non-linear solvers to avoid convergence issues and spare computational time.

The computational cost of large-scale groundwater models is usually reduced by using simpler models. One example is MODFLOW where a set of vertical 1D soil columns are used for unsaturated areas and a 2D surface model for the saturated areas. On the one hand, such a setup makes sense as, at the large scale, the unsaturated water dynamic is essentially vertical [4]. On the other hand, such a design involves a number of special cases when the groundwater dynamic is not standard, for instance when a saturated area appears or disappears. We choose to go with fully 3D saturated-unsaturated groundwater models.

Overall, old state-of-the-art RE models are optimized for sequential execution, limiting their use to small problems. Recently, some models focused their development on parallelization for HPC infrastructures (like Parflow or RichardsFOAM for RE, or PFLOTRAN for multiphase flow), unlocking large scale numerical experiments for RE. We share this focus in this thesis, the importance of this aspect being pointed out in the previous section.

Another design choice for our RE models is the use of a DG FE discretization, which was already implemented in the SLIM code framework. Compared to continuous FE methods, the DG FE method is better-suited for advection-dominated problems (like infiltration fronts), it brings local conservation of the mass with an explicit description of the fluxes, the possibility to use different spatial discretization orders per element (but such p-refinement method has not shown impressive performances when applied to the RE [5]) and the possibility to use physically-discontinuous variables, like the water content. It is also more scalable when used with explicit time integration schemes as the mass matrix is block-diagonal, but this property is already achieved if the mass matrix is

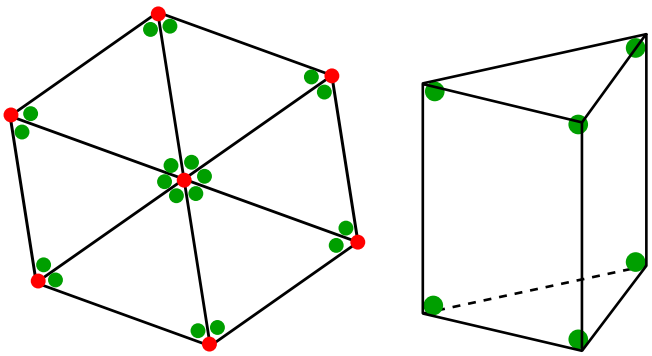


Figure 1.2: Left: illustration of a 2D mesh of continuous (red) and discontinuous (green) DOFs in linear triangular elements. Dots around the same triangle vertex are in practice located at the same place. The DG method then allows several values at each vertex. In 2D for the same messh the DG mesh has 6 times the number of DOFs than the continuous mesh. Right: illustration of a 3D bi-linear triangular prism. The increase in DOFs between the two methods is now close to 12 folds.

lumped. DG FE methods need more degrees of freedom (DOF) than continuous ones, as shown in Fig. 1.2. Finite volume methods possesses roughly the same advantages as DG FE in this context compared to continuous FE. However finite volumes need more complex methods to achieve order of convergence beyond linear. A piecewise linear DG FE method is second-order accurate in space while the standard finite volume method is only first-order accurate.

The NI SWE is a compromise between the full SWE and more basic representations of the surface water like the kinematic wave equation (pure advection in the slope direction). It allows the creation of rivers and reservoirs like ponds or lakes. The DG FE method has been tested on this equation but had issues with negative water heights. It is why the control volume FE (CVFE) method, quite robust and efficient, was used instead. We used this method as such, focusing on the coupling of that surface water model with our subsurface model. That coupling is challenging as the time scales between the models are quite different and various physical processes can occur (evaporation, direct infiltration, excess infiltration, ponding).

To summarize, our objective is to derive numerical methods and algorithms able to model the surface and subsurface water. These tools have to be robust, mass conservative, precise, efficient and scalable.

1.3.2 Thesis summary

Our surface-subsurface model can basically be divided into three parts. The RE for groundwater, the NI SWE for surface water and eventually their coupling. For RE, two numerical approaches have been selected and tested. The overall idea is to obtain models which scale well on large HPC infrastructures, while preserving important physical properties such as mass conservation.

To produce this thesis, many model designs and numerical approximations have been tested for the RE, but those producing unsatisfactory results have been abandoned. The main one is presented in Appendix A.1. One goal is to reduce the number of approximations and their impact on the solution. Eventually, the main points of what emerged from those tests and the literature is compiled into Chapter 2, along with methods to optimize and compute parallel efficiency and a brief overview of other state-of-the-art models.

After this more technical introduction, in Chapter 3 we describe a RE model with a time discretization that scales well on parallel infrastructures: an explicit time integration scheme. The main drawback is the elliptic part of the equation that cannot be solved with such a method. An additional approximation is then applied, slowing the information propagation in saturated areas from instantaneous to a finite speed. The impact of this speed depends both on the number of elements to be crossed by the information and the number of sub-iteration applied during a time step. The model works perfectly for unsaturated simulations, and shows good results for simulations with shallow saturated areas.

Then, as a proof of concept, the first RE model is coupled with a state-of-the-art surface water model: the NI approximation of the SWE, spatially discretized with a CVFE method and with an implicit Euler integration scheme in time. The coupling is done weakly, by naturally extending the DG FE spatial discretization used in the subsurface model, without requiring an user-defined parameter. The time discretizations for the sub-models are harmonized, leading to a more efficient time step. Comprehensive description and analysis are given in Chapter 4.

Eventually, the first RE model was redesigned to avoid the convergence issues in saturated areas. This is achieved by solving the elliptic part of the equation with an implicit method. It comes at the cost of a slight drop of the model scaling obtained beforehand. The model is thoroughly described in Chapter 5.

1.3.3 Supporting publications

T. De Maet, E. Hanert, M. Vanclooster, A fully-explicit discontinuous Galerkin hydrodynamic model for variably-saturated porous media, Journal of Hydrodynamics 26 (4) (2014) 594–607

T. De Maet, F. Cornaton, E. Hanert, A scalable coupled surface–subsurface flow model, Computers & Fluids 116 (2015) 74–87

T. De Maet, E. Hanert, Towards a multi-scale model of the land-sea continuum, in: ENVITAM PhD day, 2010

T. De Maet, E. Hanert, A material discontinuity capturing model based on discontinuous Galerkin method, in: 35èmes Journées du GFHN, 2010

T. De Maet, E. Hanert, Towards a 3D continuous-discontinuous Galerkin subsurface water-flow model, in: Joint ISSH and ICWRER Symposium, 2010

T. De Maet, E. Hanert, An explicit discontinuous-Galerkin surface-subsurface water flow model, in: Fifth International Conference on Advanced COmputational Methods in ENgineering (ACOMEN 2011), 2011

T. De Maet, E. Hanert, Towards an integrated model of the surface-subsurface water cycle, in: PhD Student Day ENVITAM, 2012

T. De Maet, E. Hanert, Towards an integrated and multi-scale model of the land-sea continuum, in: EGU General Assembly Conference Abstracts, Vol. 14, 2012, p. 9634

T. De Maet, E. Hanert, E. Deleersnijder, T. Fichefet, V. Legat, J.-F. Remacle, S. Soares Frazao, M. Vanclooster, J. Lambrechts, C. König Beatty, et al., SLIM: A multi-scale model of the land-sea continuum, in: EGU General Assembly Conference Abstracts, Vol. 14, 2012, p. 9603

T. De Maet, E. Hanert, A 3D explicit discontinuous-Galerkin groundwater model, in: EGU General Assembly Conference Abstracts, Vol. 14, 2012, p. 9665

CHAPTER 2

About the Richards equation: approximations, methods and scaling

Abstract

Some approximations for the Richards equation (RE) are used in many models and commonly accepted as good solutions to different numerical issues. However, their purpose is not alway clearly defined and they often lack of explanatory examples. The large number of different methods is also quite confusing. The pros and cons of these approximations have to be clearly explained and placed in their context. Another concern of modern RE models is the parallel efficiency on HPC infrastructures. It is not simple to achieve and there is no a clear way to measure it for model inter-comparison. In this chapter we discuss some of these approximations and methods, along with potential impact on the model result. A section is dedicated to different parallelization optimizations and breakpoints, along with some measurement considerations. Eventually, we provide a concise summary of state-of-the-art terrestrial models, focusing on parallel scaling.

2.1 Introduction

Despite its simple appearance, the RE can be very difficult to solve. Indeed, ignoring the physically-present air phase produces strong non-linearities in the constitutive relationships. A number of approximations, concepts and numerical methods are routinely used by modelers to handle those non-linearities. For the newcomers to that field, it is often difficult to understand the respective advantages and drawbacks of each approximation. A compilation of the main concepts and their consequences would therefore prove useful.

The RE have a multi-scale and sometimes elliptical behavior, which makes its parallelization on large computing infrastructure complex. Until recently, most optimizations have been done to increase its robustness or sequential performance. The parallel performances of those models are thus quite poor. As this thesis focuses on that point, a summary of the necessary optimizations to achieve a good parallel efficiency along with the current state-of-the-art are necessary.

In this chapter, some of the experience gained during our Ph.D. project is summarized, as well as some tricks, tips and suggestions. All is included in this chapter along with some practical examples. Most of the chapter deals with the non-linearity of RE and the different ways to handle it: mass lumping, interpolation of the constitutive functions, mass imbalance origin, modified Picard iteration, change of variables, variable switching, high order discretizations. In the second part, different parallel scaling optimizations and pitfalls are presented along with more details on how to measure the parallel efficiency. Eventually, a short overview of state-of-the-art models for terrestrial water is given.

2.2 Brief introduction to porous media physics

Physical considerations

At the millimeter scale, intermolecular forces create adhesion between some fluids (like water) and the surrounding materials (like soil particles). In small tubes (pores) those forces combined with the forces of cohesion present inside a fluid can be stronger than gravity and elevate a column of fluid until reaching an equilibrium. Such a mechanism occurs in trees, soils and also in a simple dishcloth. This adhesive force is called the capillary effect.

A porous medium can be seen as a continuum of pores of various radii, within which fluids are driven by capillarity. This description has two limitations: the medium can be discontinuous (fractures) and large pores are not driven by capillarity. The suction pressure required to empty a filled pore depends on its radius and is assumed to be negative.This pressure acts as a threshold: above

its value the pore is empty, below it is filled. If the pores, following a given distribution in radii, are uniformly distributed in space, the capillary capacity C can be derived from this distribution, linking the pressure required to empty a filled pore (denoted as the pressure head ψ [m]) to each pore radius. This is depicted in Fig. 2.1.

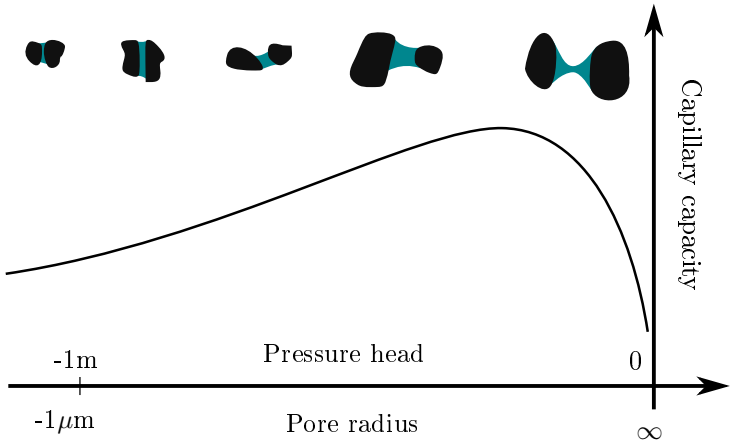


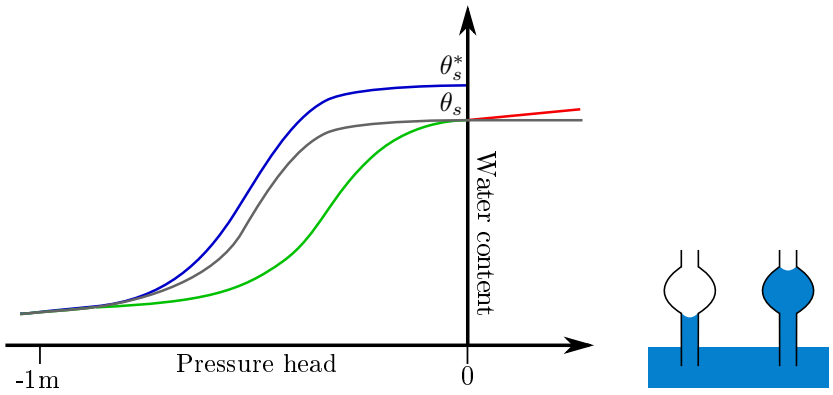
Figure 2.1: The capillary capacity C represents the increase rate of water content with pressure head ψ, at a given ψ. Each ψ is linked to a specific pore radius and $C(\psi)$ is proportional to the volume and density of pores with this corresponding radius. The area under the curve between two values of ψ is the water content that can be hold in the pores of corresponding range of radii.

If we consider that, at a given ψ, all the pores of radius smaller than this threshold are automatically filled, we can link C to the volume of fluid present in the porous media. The limitations of this second hypothesis are first that it is not obvious that the fluid is continuous between two regions, especially in dry condition. In such a case, it would be impossible to exchange fluid between those two regions. Secondly it assumes an instantaneous adaptation of the fluid to the imposed pressure, because transitional effects in pores can often be neglected at the time scale of the involved processes. The integration of C along ψ thus leads to the fluid (water) content θ [-], which is the volumetric fraction of fluid in a volume of porous media:

$$\theta = f_\theta(\psi) = \int_{-\infty}^{\psi} C(\psi) d\psi \quad \text{or} \quad \frac{\partial \theta}{\partial \psi} = C(\psi) \tag{2.1}$$

The relation $f_\theta(\psi)$ linking θ and ψ is called the retention curve. It is strongly linked to the porous media properties and in general is not a function in the

mathematical sense. Indeed, strong hysteresis can be present in the relation due to many factors (Fig. 2.2), the principal one being the ink-bottle effect: when a pore presents a larger section at some point, it is easier to unfill this section than to fill it (Fig. 2.2(b)). For a same ψ, two values of θ are then possible for this pore. In this work and in many model, this hysteresis effect is neglected and the retention curve taken as a true mathematical function. This is mostly done to avoid additional complexity, and the resulting errors can be substantial. Brooks and Corey derived in 1964 a mathematical expression for $f_\theta(\psi)$, followed by Van Genuchten in 1980 (see Eq. (3.7)).



(a) Hysteresis effect, entrapped air and expansive soils.

(b) Ink bottle effect.

Figure 2.2: (a) The classical retention curve considered as a true mathematical function in many models (grey curve) can see several deviations. The hysteresis effect, strongly present in some type of soils, changes this relationship. The equilibrium point between θ and ψ is then located between the drying curve (in grey) and the wetting curve (in green), accordingly to the wetting/drying history. Fast water infiltrations can trap some air, effectively reducing the available volume θ_s^* to θ_s after a full drying/wetting cycle (*i.e.* descending the blue curve and ascending the gray one). Eventually, expansive soils can virtually increase the water content by increasing the size of the representative elementary volume of soil (red line). (b) This schema depicts the main hysteresis effect, where two identical pores including a broader section can show different water content for the same ψ value.

The porous media become saturated when θ reaches a maximum value θ_s. In saturated areas, the physics is no longer governed by capillarity but instead by the hydrostatic pressure. That maximum value can be smaller than the available space due to the presence of another phase, like entrapped air (see Fig. 2.2(a)). It is also possible that the presence of a fluid modifies the porous

media structure, like in expansive soils where the representative volume can be seen as expanded with positive pressure (the red line in Fig. 2.2(a)).

Mathematical derivation

The Darcy law describes the flux in a porous media based on the pressure drop between two points. It can be derived from the Navier-Stokes equation, the main hypothesis being the incompressibility of the fluid and a laminar flow in the pores, hence neglecting inertial effects. The fluid and the porous media matrix are often considered to keep a constant density, neglecting temperature effects (which can be consequent). The Darcy law with gravitational effect reads:

$$\mathbf{q} = a(\nabla p - \rho \mathbf{g}) \tag{2.2}$$

where $\mathbf{q}$ [ms^{-1}] is the Darcy flux, a [$\mathrm{m}^2\mathrm{s}^{-1}\mathrm{Pa}^{-1}$] a parameter linked to the porous media, p [Pa] the fluid pressure, ρ [kgm^{-3}] the fluid density and $\mathbf{g}$ [ms^{-2}] the gravitational vector. Changing the variable p by ψ and considering anisotropic media, we obtain:

$$\mathbf{q} = \mathbf{K} \cdot \nabla(\psi - z) \tag{2.3}$$

with $\mathbf{K}$ [ms^{-1}] the conductivity tensor and z [m] the upwards positive vertical coordinate. The pressure head ψ [m] is defined as the height of a fluid column exerting the pressure, and it is therefore normalized by gravitational acceleration. Richards equation (RE; first derived by Richardson in 1922), designed for water in soils, is obtained by placing this flux inside an equation of conservation:

$$\frac{\partial \theta}{\partial t} = \nabla \cdot \mathbf{q} + s \tag{2.4}$$

where t [s] is the time and s [s^{-1}] a source term. In RE, the Darcy flux originally designed for saturated porous media is extended to unsaturated areas. To reflect the capillary dynamics, a negative ψ is used. It can be seen as a height *above* a water column (*e.g.* a water table). The conductivity $\mathbf{K}$ varies with the cross-sectional area and conductivity of the remaining filled pores. Several relationships were found to link ψ and $\mathbf{K}$, like the functions of Brooks and Corey (1964) and of Mualem (1976; see Eq. (3.8)).

It should be noted that it is frequent to see an additional term in the left-hand side of Eq. (2.4) [16, 17, 18]. In such a formulation, the main variable is the water saturation $S_w = \theta/\theta_s$. The time derivative is then expressed as:

$$\frac{\partial S_w \theta_s}{\partial t} = S_w \frac{\partial \theta_s}{\partial t} + \theta_s \frac{\partial S_w}{\partial t} \tag{2.5}$$

$$\simeq S_w S_s \frac{\partial \psi}{\partial t} + \theta_s \frac{\partial S_w}{\partial t} \tag{2.6}$$

where S_s is the specific storage coefficient and the final θ_s is considered constant. This account for the compressibility of water (negligible in most cases) and for compressibility of the medium (*e.g.* for expansive soils). Its presence also improves the convergence of solvers as it avoids a fully elliptic system. However, it introduces a term to the conservation equation, and with it, it is no longer clear that the time discretization is exactly mass conservative. We therefore do not use it in this work, and show in Appendix A.4 that it is mathematically equivalent to a modification of the retention curve in the classical formulation. The specific storage coefficient can also be directly applied as a slope for f_θ when $\psi > 0$ [19]. This is the approach we selected, although no specific storage coefficient is necessary in the test cases presented in this work.

Eq. (2.4) can be transformed in a ψ-only form:

$$C(\psi)\frac{\partial \psi}{\partial t} = \nabla \cdot [\mathbf{K}(\psi) \cdot \nabla(\psi - z)] + s \tag{2.7}$$

2.3 Richards Equation properties

Now, let us analyze the 1D RE:

$$C(\psi)\frac{\partial \psi}{\partial t} = \frac{\partial}{\partial z}\left(K(\psi)\left(\frac{\partial \psi}{\partial z} - 1\right)\right) + s, \tag{2.8}$$

$$= \frac{\partial K(\psi)}{\partial z}\left(\frac{\partial \psi}{\partial z} - 1\right) + K(\psi)\frac{\partial^2 \psi}{\partial z^2} + s, \tag{2.9}$$

$$= \frac{\partial K(\psi)}{\partial z}\frac{\partial \psi}{\partial z} - \frac{\partial \psi}{\partial z}\frac{\partial K(\psi)}{\partial \psi} + K(\psi)\frac{\partial^2 \psi}{\partial z^2} + s, \tag{2.10}$$

$$= \left(\frac{\partial K(\psi)}{\partial z} - \frac{\partial K(\psi)}{\partial \psi}\right)\frac{\partial \psi}{\partial z} + K(\psi)\frac{\partial^2 \psi}{\partial z^2} + s. \tag{2.11}$$

The last equation looks like as an advection-diffusion equation, with K/C as diffusion factor and a more complex $(\partial K/\partial z - \partial K/\partial \psi)/C$ advection factor. Due to the non-linear functions $C(\psi)$ and $K(\psi)$, the Richards equation can display three classical behaviors:

elliptic in saturated zones, where K is constant and $C \simeq 0$. The advection factor is then zero and the time derivative is negligible, leading to a purely elliptic equation, for which the solution is only driven by the boundary conditions and the source terms. Obviously, the solution of this stationary equation instantly reaches the steady state. The model equation then reduces to:

$$0 = \frac{\partial^2 \psi}{\partial z^2} + \frac{s}{K_{sat}} \tag{2.12}$$

hyperbolic (advective) in unsaturated zones, when $K/\Delta z \ll \partial K/\partial z - \partial K/\partial \psi$, the equation is mainly advective[1]. That occurs at infiltration fronts where ψ shows strong variations in space. The model equation then reads

$$\frac{\partial \psi}{\partial t} = \left[\frac{1}{C(\psi)}\left(\frac{\partial K(\psi)}{\partial z} - \frac{\partial K(\psi)}{\partial \psi}\right)\right]\frac{\partial \psi}{\partial z} + \frac{s}{C(\psi)} \tag{2.13}$$

parabolic (diffusive) in unsaturated zones, when $K/\Delta z \gg \partial K/\partial z - \partial K/\partial \psi$, the equation is mainly diffusive. It is the default behavior of the equation in unsaturated zones outside infiltration fronts. It reads

$$\frac{\partial \psi}{\partial t} = \frac{K(\psi)}{C(\psi)}\frac{\partial^2 \psi}{\partial z^2} + \frac{s}{C(\psi)} \tag{2.14}$$

In practice, those dynamics are often mixed, without a clear delimitation between them. At the interface between two dynamics, the discretized volumes have mixed behavior, and typically such interface moves during a simulation.

The mix of advection and diffusion is well known. The advective term is often treated separately with special treatment like upwinding to avoid numerical issues. The DG FE spatial integration method is specially well designed for this type of equation, and explicit time integration method can be easily applied.

In contrast, the mix of advection-diffusion with an elliptic equation is more difficult, because the time derivative disappear when the problem becomes elliptic. Purely elliptic equations are quite simple to solve with basic implicit time integration solvers, but explicit methods are prohibited. The interface between non-elliptic and elliptic zones is challenging as the pressure variable switches between a progressive variation in time to an instantaneous adaptation to the boundary conditions.

2.4 Dealing with non-linearities

The approximations described in this section are found in most models of RE. They are however often barely mentioned, and even less often described. Here, we try to shed some light on their formulation and their consequences.

2.4.1 Mass lumping

Once discretized in space, a differential equation is transformed into a large system of discrete equations, which can be written in term of large sparse matrices. One of them is the mass matrix M (see for instance Eq. (3.21)),

[1] The Δz is a characteristic length of the system, typically of the order of magnitude of an infiltration front extent.

containing information about element volumes and shapes. In the case of the DG FE method, M is block-diagonal (*i.e.* an empty matrix apart from square blocks on the diagonal). The mass lumping consists in modifying M into a purely diagonal matrix. Several methods are possible but here we will only consider the specific lumping obtained by simple "row sum", which is to our knowledge the only one used for the RE. The entries of the lumped mass matrix M^* then read:

$$M^*_{ii} = \sum_j M_{ij}, \tag{2.15}$$

$$M^*_{ij} = 0 \text{ for } i \neq j \tag{2.16}$$

Overall, the lumping technique is mostly used with explicit time integration schemes to avoid the inversion of the mass matrix. Indeed, the mass then resumes to a diagonal matrix which transforms the linear problem into simple vectors arithmetics. The main drawback is obviously an additional numerical error. It should be noted that if the mesh is fixed and the mass term constant, this method does not improve performances, because the mass matrix inversion can be done once for all before iterating, allowing cheap computations without any approximation.

In the case of RE the mass term is variable, mass lumping can hence spare some computational time for explicit schemes. However that is not why it is used here. Indeed in the RE context, the lumping is known to improve the rate of convergence for implicit schemes [20]. As the whole mass term is M times the capillary capacity C, it is strongly non-linear. Without modification, it is possible to obtain configurations where the product CM has off-diagonal terms dominating the diagonal ones, removing the positive-definite property of the matrix. Although diagonal dominance is mathematically not an absolute criterion to allow a solution to exist, it is often required to ensure a good convergence in iterative solvers, which are the most efficient for large problems. The system is easier to solve if the matrix eigenvalues are close, and being positive-definite ensure eigenvalues of the same sign. More intuitively, as solving a system of equations is mathematically equivalent to the transformation of a matrix into a pure diagonal, it is intuitive that a non-diagonal dominance can lead to bad or ill-conditioned problems.

The second main reason is that mass lumping also removes some artifacts due to non-linearities that are known as spurious oscillations [16, 20, 21]. In a classical approach, the steep gradient of infiltration fronts can produce values below the initial minimum, which in extreme cases can produce negative values of the water content θ. Such low values in the front influence the direct neighbors,

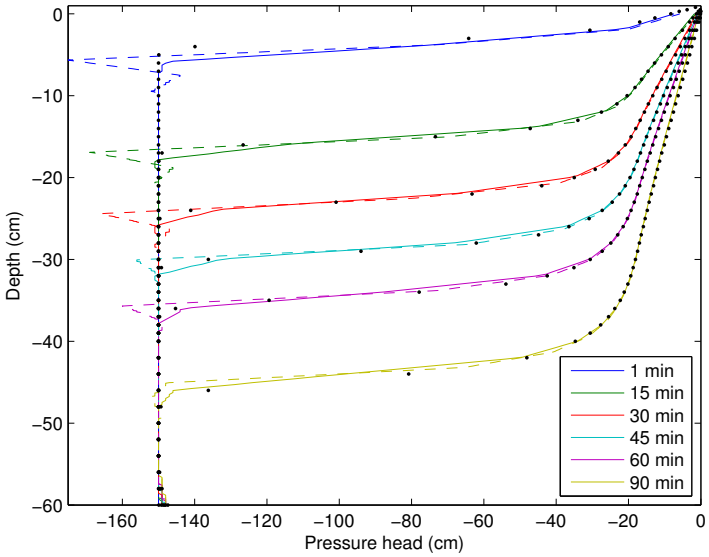


Figure 2.3: A classical effect of the mass-lumping on an infiltration front. The dots represent the solution obtained with Hydrus1D model (which is lumped). The continous/dashed lines represent the solutions obtained with our fully-explicit model with respectively a lumped/normal mass matrix.

which tend to show an oscillating pattern (Fig. 2.3). The origin of that behavior is difficult to isolate but is somewhat linked to the dominance of off-diagonal terms as before. It is possible to get accurate infiltration front speeds despite these oscillations, but besides its unphysical and unattractive look, in extreme cases it can cause the model to crash.

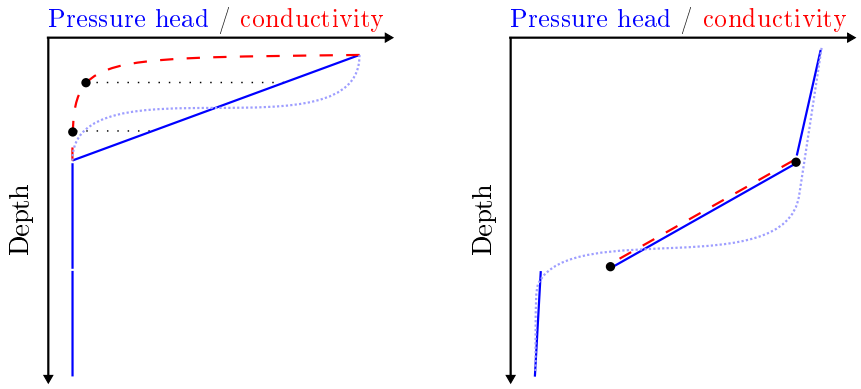
2.4.2 Constitutive relationships interpolations

It is a common (but often untold) practice to evaluate the values of the non-linear functions K and/or C at the nodes of the FE mesh [16, 20, 21]. That implicitly linearizes the function on each element with a simple interpolation (linear for lines/triangles/tetrahedrons, bi-linear for quadrangles, etc.), in comparison with classical Gauss-Legendre quadrature rule (exact for polynomials of degree $2n-1$ with n the number of points), which can bring optimal precision. The Gauss-Lobatto quadrature rule (exact solution for polynomials of order $2n-3$) generalizes such practice for higher-order precision [22]. In 1D for a 2-point stencil the Legendre method can handle 3^{rd} order polynomials while Lobatto only the 1^{st} order. Despite that such a linearisation strongly

decreases the precision in the evaluation of C and K inside the elements, it is nonetheless applied to limit the effect of the strong variations present in some elements.

To illustrate the need of such an interpolation, we consider a classical rain event. To model the infiltration following a rain event it is recommended to use a fine discretization near the surface [23]. The reason is that infiltration fronts with strong gradients naturally start there. To numerically solve such a dynamic, it is better to use small elements to better approximate steep gradients. However, a good fit is difficult to achieve even with elements as small as 1 cm, especially if the soil is initially very dry.

Fig. 2.4(a) shows the 3 uppermost linear elements of a 1D simulation at the beginning of a rain event. The rain is simulated by imposing $\psi = 0$ at the surface, the first element already satisfies this condition. A naive FE model is then blocked. This is due to the high non-linearity of the conductivity function K. A classical Galerkin FE model evaluates K at integration points of a given quadrature method to optimize both precision and computational time. In our 1D case, the K values at those 2 points (in black) are close to zero, in contrast to the uppermost node where the K value is at its maximum. This implies that the flux (which is K times the gradient) will be virtually absent in the volume of the element. As a result, nothing moves.



(a) Stalled infiltration front with low conductivity at quadrature points.

(b) A linearized conductivity is closer to advective physics.

Figure 2.4: Beginning of an infiltration event modeled with a 1D discretization. The conductivity (in red) is evaluated with a quadrature rule at the two black dots in panel (a) and is linearized or evaluated at nodal points in panel (b). The light-blue dotted line is what an analytical solution for an infiltration front can look like.

One can easily see that if the element was twice smaller, the gradient would double and K would keep the same values. Refining the discretization can then reduce this issue, but it does not solve it. In very dry initial conditions, an infiltration front is close to a step-function, and it is moving. It is not possible to fully solve it everywhere when it is progressing, especially with a fixed mesh.

A solution is to linearize K between nodal values, hence strongly increasing the flux in the element volume and allowing the "physical" advective behavior to happen. This strong approximation can be tempered if we consider the sub-element physics. Indeed, inside the element the front is close to a step function moving through it. The linearization can then be seen as a linear fit of such sub-element dynamic (Fig. 2.4(b)). It can also be seen as a kind of upwind method. It is however not well designed for other processes like evaporation (see 5.4), although it yields satisfactory results in most of the cases.

When linearized, it is possible to move the K function outside of the volume integrals, which results in an approach where variables and functions can be evaluated at nodes only (both for interface terms and volume terms). An appropriate implementation can reduce the computational time by pre-computing those constant integrals.

In ψ-forms of the RE, it is important to evaluate C and K with the same values of ψ, as a disconnection of their values – sometimes ranging on many orders of magnitude inside an element – can produces unphysical behavior. Such an issue can be observed for low values of ψ, where the RE is mainly diffusive with a diffusivity proportional to K/C, and with very small values for both K and C. If K is linearized, it is therefore recommended to linearize C with the same method. Models using mass-lumping naturally compute C at the nodes as the mass is already purely nodal.

The two main variables ψ and θ are mathematically linked by a function. If a model uses both variables – as those presented in Chapters 3 and 5 – the numerical transform between them has to be consistent with any other representation of that transform. As such, if a L_2 projection[2] is used for the retention curve, it is recommended to use the same method for its derivative C. Oscillatory behaviors have been observed otherwise.

In the models considered here, a generic FE framework is used[3] and an explicit linearization of the fields K and C has been necessary to bypass the quadrature rules already in place. In an optimized model, these values should be directly computed at the nodes.

[2] A L_2 projection consists in approximating a field of values on a FE mesh, minimizing the sum of the squares of the differences between the field and its FE representation. Numerically, it consists in minimizing this difference at the predefined quadrature points.

[3] It is the basement of the model SLIM

Linearizing the constitutive relationships does not come without consequences. It provides more physical results by degrading the numerical precision. If the physics was correctly modeled inside each element (*e.g.* with higher order discretizations, see Section 2.5.4) those approximations would not be necessary. Sub-models can also lead to a more accurate solution (see Section 5.4).

2.4.3 The hidden mass of the ψ- and mixed-forms

The ψ-form of RE is mathematically equivalent to its mixed-form, as we have $\frac{\partial\theta}{\partial t} = \frac{\partial\theta}{\partial\psi}\frac{\partial\psi}{\partial t} = C(\psi)\frac{\partial\psi}{\partial t}$. However, they behave differently once discretized in time. When the Implicit Euler scheme is used, these two forms read:

$$C(\psi^{n+1})\frac{\psi^{n+1}-\psi^n}{\Delta t} = \nabla\cdot\mathbf{J}(\psi^{n+1}) \tag{2.17}$$

$$\frac{\theta^{n+1}-\theta^n}{\Delta t} = \nabla\cdot\mathbf{J}(\psi^{n+1}) \tag{2.18}$$

with $\mathbf{J}$ a Darcy flux. The composition of this flux is not important for mass conservation, as the divergence operator ensures that, whatever the flux, we will obtain the exact difference between the mass gain and loss if the equation is correctly discretized in space.

It is already clear that both equations diverge as C is arbitrarily evaluated at $n+1$ and not with a value of ψ that satisfies the retention curve. It is easy to show by using the mean value theorem that there exists a value ψ^* between ψ^n and ψ^{n+1} for which the mass balance is achieved (see Fig. 2.5). In case of Implicit Euler this is

$$C(\psi^*) = \frac{\theta^{n+1}-\theta^n}{\psi^{n+1}-\psi^n}. \tag{2.19}$$

which is the first order derivative of $C \triangleq \frac{\partial\theta}{\partial\psi}$. When $C(\psi^*)$ is used in place of $C(\psi^{n+1})$, mass conservation is reached in Eq. (2.17) as it becomes mathematically equivalent to Eq. (2.18). For all other values of ψ (and in particular ψ^{n+1}), mass is gained or lost between iterations. Achieving a good mass balance is especially important when the water fluxes are used inside tracer models.

The chord-slope approximation [24, 25] directly uses Eq. (2.19) in a ψ-form of the RE to improve mass balance. However, it brings additional numerical issues for dry cases and steady nodes, where the ψ variation is close to zero [26, 16].

It is less evident to see that even for the mixed-form of the equation or the chord-slope approximation of the ψ-form, the mass error is still present but often neglected as strongly scaled down. Indeed, in saturated areas, it is not possible to use θ as the driving variable since it is constant. It is then natural

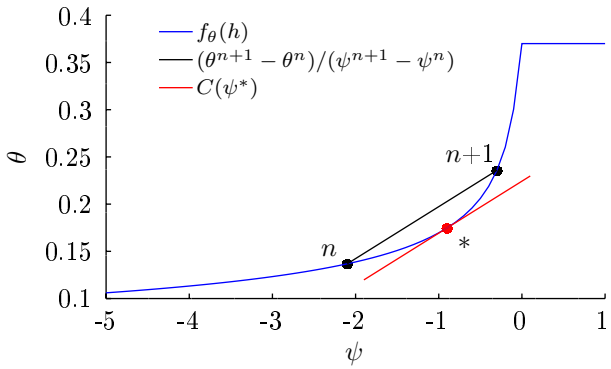


Figure 2.5: As the retention curve is continuously differentiable (even at $\psi = 0$ in the Van Genuchten parametrization), there is a value ψ^* between ψ^n and ψ^{n+1} for which $\frac{\partial\theta(\psi^*)}{\partial\psi} = \frac{\theta^{n+1}-\theta^n}{\psi^{n+1}-\psi^n}$.

to use ψ as the primary variable in non-linear solvers. However, when using an iterative solver, the non-linear functions supposedly computed at time $n+1$ are in fact computed with the last sub-iteration value for ψ, not the final one. Let us explain this with 3 non-linear solvers, in which the sub-iterations are identified with k (the $n+1$ superscript is hidden for readability):

$$\left.\frac{d\left(\theta - \Delta t\nabla\cdot\mathbf{J}\right)}{d\psi}\right|^k (\psi^{k+1} - \psi^k) = \theta^n - \theta^k + \Delta t\nabla\cdot\mathbf{J}^k \tag{2.20}$$

$$\theta^k - \theta^n = \Delta t\nabla\cdot\mathbf{J}^{k,k+1} \tag{2.21}$$

$$\theta^{k+1} - \theta^n = \Delta t\nabla\cdot\mathbf{J}^{k,k+1} \tag{2.22}$$

The Newton method in Eq. (2.20) shows that the mass is conserved if the left-hand term is zero. That term converges (theoretically quadratically) towards zero but any residue changes the mass balance. In Eq. (2.21) the Picard method is used with a "one-step-late" time derivative with ψ as the driven variable. Beside the fact that this scheme probably poorly converges, we see that even with a good solution the mass term is always a sub-step beyond. This scheme is not exactly conservative as θ^k is obtained from $f_\theta(\psi^k)$, which is different of $f_\theta(\psi^{k+1})$. The other Picard method in Eq. (2.22) seems right, but in fact it is close to the previous one. Indeed it is not solvable as such because θ^{k+1} needs to be expressed in term of ψ (*e.g.* via a linearization) to be put into the linear system. This idea is at the base of the next section, although it will not fully solve this issue.

2.5 Methods to cope with the non-linearities

Many strategies have been developed to modify classical methods in order to ensure mass conservation or a better convergence in highly advective cases. Below, we present the most widely-used approaches.

2.5.1 The modified Picard iteration

The modified Picard iteration is often referred as "the" mixed-form of the RE. It is based on the mixed-form solved by a classical Picard non-linear solver, like in Eq. (2.22). Celia et al. [27] have modified the solver iteration by linearizing the θ^{k+1} term with a first order Taylor series expansion:

$$\theta^{k+1} \simeq \theta^k + \left.\frac{d\theta}{d\psi}\right|^k (\psi^{k+1} - \psi^k) \tag{2.23}$$

$$\left.\frac{d\theta}{d\psi}\right|^k (\psi^{k+1} - \psi^k) = \theta^n - \theta^k + \Delta t \nabla \cdot \mathbf{J}^{k,k+1} \tag{2.24}$$

One easily sees that Eq. (2.24) is now closer to the Newton solver described in Eq. (2.20) and shares the same property: when the solution has converged, $\psi^{k+1} - \psi^k$ vanishes and mass is conserved. As the C function $(d\theta/d\psi)$ is isolated in the vanishing term, it is not anymore an issue. As this method is easy to implement, it is a very popular approach. The Newton method (Eq. (2.20)) is also often considered, its theoretically quadratic convergence is balanced by the extra cost of the additional derivatives (that can be computed either analytically or numerically) [28].

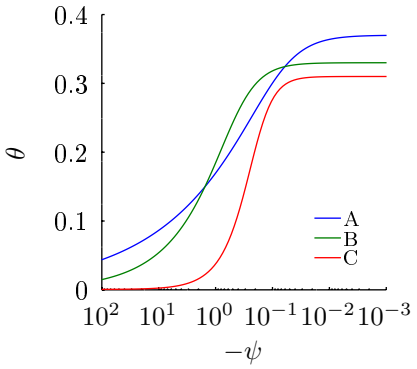
2.5.2 Change of variable

There is a smart way to reduce the non-linearity of the RE, which consists in a simple mathematical change of variable [29]. The main drawback is the soil-dependence of the optimal choice of variable.

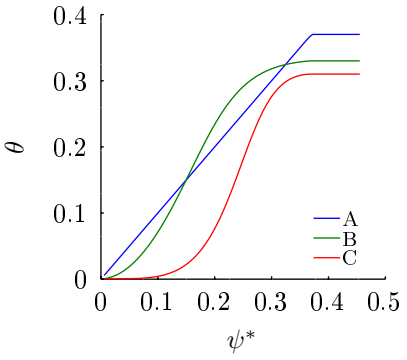
Given a variable transform $\psi \to \psi^*$ such as $\psi = g(\psi^*)$. The ψ^*-form of the RE is then:

$$C^* \frac{\partial \psi^*}{\partial t} = \nabla \cdot \left(\mathbf{K} \cdot \left(\frac{dg}{d\psi^*} \nabla h^* + \mathbf{z} \right) \right) \tag{2.25}$$

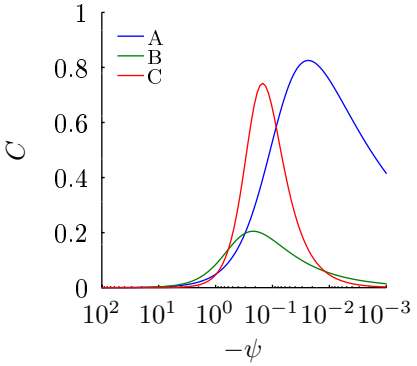
It is desirable for ψ^* to be close to $f_\theta(\psi)$ in the unsaturated area. Indeed, in that case $C^* = \frac{\partial f_\theta}{\partial \psi^*} \simeq 1$, which removes the mass factor. Using the ψ^*-form of the equation with such a transform in the unsaturated zone is roughly the same as using the θ-form. An example is given with the blue curves in Fig. 2.6 (a)-(d). As a positive side-effect, that transform reduces the variations of the K function (see Fig. 2.6(e)). The simplification of both C and K explains why the θ-form of the equation is better-suited for dry soils than the ψ-form.



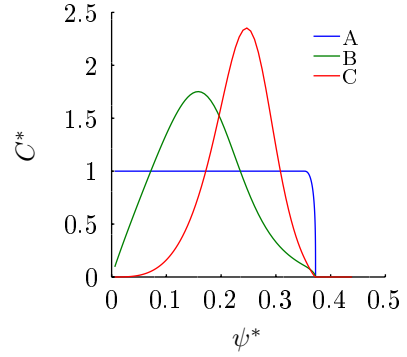
(a) Retention curve



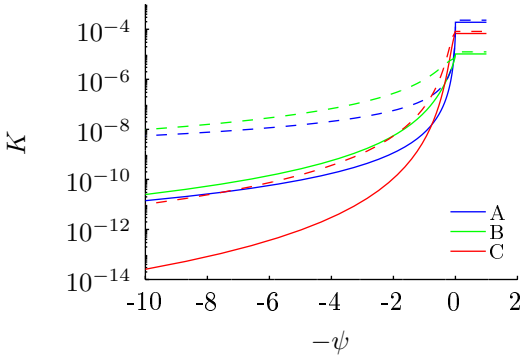
(b) Transformed retention curve



(c) Capillary capacity



(d) Transformed capillary capacity



(e) Original (continuous lines) and transformed (dashed lines) conductivity

Figure 2.6: Constitutive relationships for 3 soils types denoted A, B and C and taken from the ditch experiment in Section 5.5.2. Both the initial and transformed functions are displayed. The transformed functions have a change of variable optimized with the soil A properties. It should be noted that the graphs for the original curves are displayed with logarithmic scales.

The f_θ function is soil-dependent and for most applications several soil types are present. However only one transform is possible. Nevertheless, it is still possible and useful to use such a variable transform. Whatever the choice of soil of reference, it still decreases the stiffness of both C and K, but in a variable way depending on the soil type as shown in Fig. 2.6 (green and red curves). Another challenge is that the ψ^* field is now discontinuous across different soil types, like θ [30]. We already have solved that issue for the θ field with adapted DG FE methods (see the end of the Section 3.3.5).

As an example, a tested and working change of variable function g based on the van Genuchten parametrization is expressed as:

$$\psi^*(\psi) = \begin{cases} f_\theta(\psi) & \text{if } \psi \leq \psi_0 \\ f_\theta(\psi_0) + \left.\dfrac{df_\theta}{d\psi}\right|^{\psi_0} (\psi - \psi_0) & \text{if } \psi > \psi_0 \end{cases} \tag{2.26}$$

$$= g^{-1}(\psi) \tag{2.27}$$

$$\psi_0 = -\frac{1}{\alpha}\left(\beta\left(\frac{2-1/\beta}{\beta-1}\right)-1\right)^{-1/\beta} \tag{2.28}$$

where ψ_0 is the pressure value at the inflexion point of the retention curve (specific to van Genuchten), chosen to ensure a continuous derivative. See Eq. (3.7) for additional informations about the van Genuchten parametrization.

Other simple function transformations [30, 31] greatly decrease the non-linearity, although there is no optimal choice when different soil types are used. More complex transformation methods have been developed, using an integral Kirchoff transform [32] or an hyperbolic function [33].

The approximation of the constitutive functions can also help RE model, both in terms of computational cost and stability [34]. Indeed, the presence of discontinuities in those functions strongly impacts the convergence of non-linear solvers, and sometimes the stability of explicit ones. As an example, the function K is often discontinuous near zero. A smoothing at the saturated/unsaturated interface can thus improve the performances. Another advantage of such an approach is the possibility to use lookup tables to reduce the matrix assembling cost as the power function used with real numbers is quite expensive [34].

A mix of both methods allows the pre-computation of the functions $f_\theta(g(\psi^*))$, $Kdg/d\psi^*$ and C^* in a look-up table (*e.g.* composed of cubic functions). The resulting functions are then just a little smoothed, and the overall model does not suffer from any overhead from complex function evaluations.

2.5.3 Variable switching

Following the same idea as the change of variable, the variable switching method consists in using a different primary variable (*i.e.* the iterated variable ψ or θ) in different areas, and updating the secondary variable accordingly afterwards.

In the original formulation [35], either θ or ψ is used in unsaturated or saturated areas, respectively. This method changes the Jacobian of the Newton non-linear solver, by deriving the residual with the appropriate variable for each degreee of freedom (DOF)[4], then applying the result on that same variable. The method improves mass balance by using the θ variable when possible, but is not fully mass conservative at the saturated-unsaturated interface where ψ has to be used. The method also significantly improves the non-linear solver convergence, especially in dry situations, which is not surprising considering the discussion in the previous section about the use of the θ-form of the equation.

An alternative algorithm was proposed by Krabbenhoft [36]. The classical ψ-form is used to compute the Jacobian and the increment $\Delta\psi$, which is modified into an increment $\Delta\theta$. The switch is applied a posteriori with the a-priori knowledge of θ, allowing to chose the ψ-form when $\theta > \theta_s$ (a significant advantage over the original formulation). The authors prove the mathematical equivalence for fully saturated and fully unsaturated cases. This method is also simpler to implement within existing ψ-form codes.

There are however some drawbacks. As explained in [16], smoothing the constitutive relationships is necessary at the different soil interfaces for continuous FE discretizations. The mass conservation is good but not perfect. Indeed, the ψ-form is used in unsaturated areas between an user-defined value and the saturation.

2.5.4 High order methods and adaptive discretizations

High order methods in time or in space are well designed for diffusive or incompressible equations, where the changes are smooth respectively in time or in space. On the one hand, computing a solution by using nodal values at many previous time steps (*i.e.* higher time discretization order) is fine when the changes for that node follow a predictable polynomial progression. On the other hand, the spatial approximation of the solution within an element with higher order polynomials works well when the continuous solution is effectively close to its polynomial approximation.

[4]Here a DOF consists in a discrete point value for a variable once the model equation is discretized in space. For instance, in a first order DG FE discretization, a 2D triangular element contains tree DOFs, one at each vertex. In such a discretization, a mesh of n triangles then has $3n$ DOFs, while a finite volume discretization on the same mesh has n DOFs. The number of DOF is also proportional to the number of variables used. The linear system matrix size is the number of DOF squared, although most of it is sparse.

In contrast, advection-dominated processes – like sharp infiltration fronts produced by rain or other instantaneous water events – lack the smoothness to allow the use of high-order methods [22]. This is why most RE models rely on first-order discretization schemes, both in time and space. Despite that, there are examples of models using high order discretization in time [37, 38, 39, 22], in space [40] or both [41]. In particular, the so-called method of lines (MOL)[5] often uses adaptive high order (1 to 5^{th}) backwards differentiation formulas in time [38, 22].

Instead of increasing the order, intelligent adaptive time-stepping [26, 42, 43, 44, 45] and adaptive meshes [46, 47, 48, 49, 40] have been considered. Li et al. [5] try high-order and adaptation in space and in time. They however recognize that it is a challenging goal and confirm that high order space discretizations are not well designed for RE, but can sometimes bring some improvements.

A promising method is the locally varying time step [50] *e.g.* concentrating computational resources on infiltration fronts. However such a method strongly increases the model complexity to achieve a good scaling.

2.6 About scaling

The model scaling on a parallel computing architecture is at the heart of this work. Indeed, the current trend to improve RE precision or range of application is to parallelize the workload on several workers[6]. In this frame, the scalability is the ability to handle growing computational capacities. The scaling efficiency is then defined as the ratio between the actual use of the computational ressources over the ressources available for the simulation. We summarize in this section how to maximize that scaling efficiency and how to measure it.

2.6.1 Solvers scaling optimizations

All RE models are iterative over time. Often, within each iteration, sub-iterations are used to solve the non-linearities through a Newton or Picard solver. A sub-iteration of a parallel implementation passes through 3 steps: the communication of the values at interfaces between workers, the assembly of the linear matrix, and the solution of the linear system.

[5] The MOL method consists in discretizing all the terms of an equation except one, *e.g.* the time derivative. For RE, this is nearly always the case as the spatial discretization is almost always decoupled from the temporal one (with a notable exception in [41]). In the RE community the MOL most often denotes the use of a state-of-the-art differential algebraic equation (DAE) solver on the time derivative. Such generic solvers are very robust and can use temporal troncation error to adapt both the time step and the order.

[6] In this work, a worker depicts a basic computational unit

To achieve a good weak or strong scaling, it is necessary to minimize the communications between workers, to maximize their work during the waiting time of others, and to balance correctly the work between all workers to finish an iteration at the same time.

The first two steps (matrix assembly and communication) have to be intricate: interfaces values are dispatched and, during the waiting time, the first part of the worker-wise matrix is computed, only using local information. Then, the second part involving interface elements is computed at the end of the communication, locally with the informations gathered from the neighbors.

The third step (solving the linear system) is easy to parallelize in the case of explicit time integration schemes with some assumptions for which it is possible to solve the linear matrix locally. If mass-lumping is applied, the matrix is diagonal and a direct solution is possible for each variable. If a DG FE method is applied, the mass matrix is element-wise block diagonal and the system can be solved element by element.

In the case of implicit time integration schemes, the linear solver will do the work of communicating off-diagonal terms between workers in a series of sub-sub-iterations (often of Krylov type). Scaling optimization on solvers is out of the scope of the present document. Indeed, a perfect scaling for general-purpose linear matrix solvers is very difficult to achieve and is still a very active research domain.

2.6.2 Scaling breakpoints

The scaling of a model implemented as described in the previous part can pass through several drops of parallel efficiency when increasing the number of workers.

The first one appears between one worker and the number of workers present on one node (one physical server). It is due to the sharing of already used resources. Indeed, the RAM, the L1/L2/L3 caches of the CPU and the bus bandwidth are the same whatever the number of workers. A unavoidable drop of parallel efficiency occurs then, especially close to the node capacity. This drop can be mitigated with a good use of the shared memory for a group of workers on a same node (direct access to the local neighborhood information as instance), with tools like openMP in addition with node-independent tools like MPI.

A second drop can appear between 1 and 2 nodes, as some information has to flow through the network. The main reason is the increased latency of each message between workers, but the bandwidth can also be limiting at some point. It is difficult to avoid this issue excepted by limiting the communication (which at this point should be already done) or using better hardware.

A third reason of parallel inefficiencies comes from the output data. Indeed, writing is slower than reading, especially on disks, and concurrent writes on a unique file is an evident bottleneck. To remove this barrier each node should have full capacity to write without waiting for others. Splitting the output to several files by domain decomposition is frequent. There is a similar issue when reading input data, especially at model initialization, *e.g.* when loading the mesh [51] or ancillary data. Firstly, each worker should only read the part of the data which is useful. But even with that precaution, if the input data is contained into one file a congestion can appear when reading the metadata of the file (this metadata indicates where the real data are). On the other hand, if the storage is centralized (which is the usual method in HPC) a large number of rather minor operations (write or read coming from each worker) can overwhelm the whole storage. Alternative methods then use groups with one worker that reads the data then split it to the members of its group. It is also possible to use a local node-wise storage, but the outputs still have to be gathered at the end. Nevertheless it can bring some improvement as other computational workload can begin when this gathering occurs and because sequential copies are known to be much more efficient than random writes over the network. In this work we simply deactivated the outputs to isolate the other factors of scaling deficiency.

Often, the fourth breakpoint is due to the solvers. Indeed, for elliptic models the information in the entire domain has to be known by all the workers. It is then increasingly difficult to achieve convergence for the linear solver. The non-linear solver can also have difficulties. Indeed, it is more likely to end up being stuck in a non-convergent oscillating state, and sometimes the linear solver can return an incorrect solution when it hits its maximum of iterations. The alternative for linear solvers is the use of multi-scale methods that get the information of the global situation then pass it to the finer scale, like multigrids preconditioners. However, they are themselves quite costly in computational time and their parallelization is not trivial [52, 53].

There is always room for additional breakpoints when reaching petascale computing. At this scale, any bit of code that is inefficient in parallel can become dominant. Any imbalance or additional latency (due to network, OS, storage, degrees of freedom per worker, etc.) is highlighted.

The mesh generation and partitioning are also difficult to parallelize but are often neglected in front of the simulation runtime. One notable exception is the case of adaptive meshes, for which the time to continuously modify the mesh is often of the same order of magnitude as the time taken by the solvers themselves.

A frequent observation with scalable models is a super-linear behavior of the scaling efficiency. That means that the model runs more than twice faster when

doubling the capacity. The classical explanation is the increase of fast memory resources like CPU caches. Indeed, if the resource usage is smaller per node (as in strong scaling test case), more computation can be done inside those caches without relying on slower memory.

2.6.3 Fair measurement of a scaling

To understand and compare model scaling efficiencies, it is important to isolate the different causes. Therefore the initialization should not be taken into account and the file writing should be limited by not exporting the variables.

From a set of several runs with the same set of parameters, the lowest runtime is taken. Indeed, it is considered that all other runs experienced troubles during the extra time (*e.g.* from other concurrent programs or from the network), which are independent from the tested model.

For large simulations, the smallest unit of scaling should start with all the resources from a single node. Indeed the available resources inside a node are not easily scaled as pointed out in the previous section. The only resource that does not grow with that rule is the network bandwidth, but if the number of 'neighbor nodes' (nodes possessing information about the neighbor of domains managed by one specific node) stay constant, the required bandwidth should be stable.

A variable of the scaling that is often not enough emphasized is the number of DOF per worker (denoted DOF/w). The scaling efficiency is nicely described by that metric. Indeed the time of computation is linked to the number of DOF (or the sub-domain volume), when the time of communication depends on the interface between workers (the surface of this volume). As the scaling performance is roughly the time of computation versus the time passed to wait for communication, it is clear that a large number of DOF/w (high volume on surface ratio) naturally produces a good scaling. An interesting metric is the number of DOF/w at which the scaling drops. That isolates the cases where a huge number of DOF/w is used to display perfect scalings. It should be noted that boosting the local computational load also enhances the scalability and therefore reduces the number of DOF/w before breakpoint. It can be done for instance by adding some sub-scale physics.

The activity rate in a simulation has an impact on scaling efficiency results. A situation close to a steady state is easy to solve and parallelize, as virtually no information has to be spread among the workers [54, 51]. In contrast, a test case with sharp local changes involving only a small part of the workers will not be well balanced from the start. Long-range effects, typical of elliptic equations, also need more communication to pass information throughout the domain. The linear matrix complexity (rows ordering, sparse structure) can have its role

as its decomposition in sub-domains can show greater interfaces with the other sub-domains. Typically a structured grid from finite differences can generate a smaller and simpler interface in its linear matrix than unstructured FE meshes.

There are two common tests to measure scaling efficiency. The first one is the strong scaling test, which compare simulations with a fixed mesh and thus no variation in the number of DOF. The number of DOF/w then decreases when increasing the number of workers, eventually reaching a breakpoint in the scaling efficiency. This test evaluates the ratio of communication over local computation (surface/volume) and shows when the former becomes dominant. The second one is the weak scaling test, which compares simulations with a constant number of DOF/w. Usually a base mesh is used then duplicated up to the number of workers. This test is often used with a very basic dynamics to avoid imbalance. Such 'dummy' weak scaling tests are nonetheless useful to isolate scaling issues as virtually nothing but implementation or algorithmic concept issues can interfere. However, we suggest that a third test should be considered to describe the real use of the model, maybe under another name as the results should be much less impressive, like a weak scaling on a heterogeneous workload, or a weak scaling on a fixed domain size like in [55].

2.7 State-of-the-art terrestrial water models

Terrestrial water models have been developed for two decades, with a broad range of different mathematical approximations and numerical methods. A representative set of surface-subsurface and subsurface-only models is given in Table 2.7. Some are built on large framework like MODFLOW [56, 17], DUNE [57] or openFOAM [25], others are developed in search for better suited numerical methods and some are built from scratch with specific optimizations. The last three rows of the table describe our models.

It is not possible to make a comprehensive unbiased ranking between all those models. Indeed, even if there was a unified bunch of test cases, the results would not be really meaningful as each model has been developed for a certain type of application. Nonetheless, such a bunch of test cases should prove useful for the selection of the appropriate tool. Those test cases should be put into practice and simulated by model developers themselves to allow for specific unbiased optimizations. The metrics to use for performance comparison are not obvious. The computational time seems essential but depends on too many parameters (spatial and time discretizations, convergence criterion, etc.). Here are a few examples of interesting results:

- Computational time for a precise problem with fixed mesh (with specified rules to match the same number of DOFs across the methods), fixed time step and a maximum error threshold. The error is then maximized up to

Name	Subsurface	Surface	Coupling	Source(s)
InHM	R,CVFE(i)	NI, CVFE(i)	α, full	[58]
WASH123D	R,FE(i)	any, FE(i)	=, full	[59]
'Morita2000'	R,FD(e/i)	NI, FD(e/i)	α, iter	[60]
CATHY	R,FE(i)	NI, FD(i)	=, nest	[61]
'Panday2004'*	R,FD(i)	NI, FD(i)	α, iter/full	[17]
Parflow	R,FD(i)	KW, FD(i)	=, full	[62]
'Dawson2008'	R,LDG(i)	SWE, DG(e)	DG, nest	[63]
Cast3M	R,MHFE(i)	KW(i)	**, full	[19]
GW	R,CVFE(i)	NI, CVFE(i)	$=/\alpha$, full	[64]
HydroGeoSphere	R,CVFE(i)	NI, CVFE(i)	$=/\alpha$, full	[18, 65]
MODFLOW	K/R,FD(i)	NI,FD(i)	α, iter	[56, 66, 67]
ParSWMS*	R,FE(i)	–	–	[68]
Hydrus3D*	R,FE(i)	–	–	[20]
RichardsFOAM*	R,FV(i)	–	–	[25]
OpenGeoSys	M,FE(i)	–	–	[69]
DuMu$^{\mathrm{x}}$*	M,FE(i)	–	–	[57]
PFLOTRAN	M/R, FV(i)	–	–	[70]
Explicit	R, DG(e)	–	–	Ch. 3
Coupled	R, DG(e)	NI,CVFE(i)	α, expl	Ch. 4
Predictor-corr.	R, DG(i/e)	–	–	Ch. 5

Table 2.1: Some state-of-the-art terrestrial water models, along with their main characteristics. **Discretizations:** finite difference/volume/element (FD/FV/FE), control volume FE (CVFE), mixed hybrid FE (MHFE), discontinuous Galerkin FE (DG), local DG FE (LDG); explicit (e) or implicit (i) in time; **Equations:** Richards (R), 1D kinematik wave on R + 2D saturated zones (K), multi-phase flow (M); kinematic wave (KW), non-inertia (NI), full shallow water equation (SWE), KW or NI or SWE (any); **Coupling:** continuity coupling (=), coefficient coupling (α); coupling with nested iterations (nest), coupling in the non-linear solver (nl), coupling in one matrix (full), iterative coupling (iter), explicit coupling (expl).
* Both ParSWMS and Hydrus3D are derived from SWMS, Panday2004 is based on MODFLOW, RichardsFOAM on openFOAM, DuMu$^{\mathrm{x}}$ on DUNE.
** In this model, the SWE is transformed into something very similar to the RE, then it is included as an additional 'soil' layer.

that limit to decrease as much as possible the computational time. This is a useful metric for a fair comparison of the algorithmic and implementation efficiencies.

- Minimizing the error for a fixed maximum computational time and a precise problem. This allows for efficient discretization methods like adaptive time stepping or adaptive meshes to display their performances.
- Minimizing the error compared to a precise lab or field experiment (up to the measurement error), with a set of measured properties. This allows for parameterizing methods to show their efficiency.

A set of such problems describing the main dynamics in the different applications should be chosen. However, the computation of the error is not trivial as many problems have no analytical solution and thus the solution should be derived from finely-discretized simulations. A rough approximation of human time spent to set-up the simulation should also be very interesting, to differentiate optimized and out-of-the-box simulations. Setting up such a bunch of tests and spread it to every modeler is a huge work beyond the scope of this thesis [71, 72][7]. However, as we focus on the parallel scaling, we are interested in that specific property amongst the other terrestrial water models.

The scaling efficiency is detached from absolute computational time value, as it estimates the computational time ratio between two simulations of the same model. A model with poor sequential performances can still compete with others if it displays a better scaling. That metric allows simpler inter-model comparisons, giving a feeling of independence in regard to the test case used. However, it is not entirely independent as described in Section 2.6.3. The simulation complexity or the physical infrastructure (which are both difficult to quantify) and the number of DOF/w have an influence on the results. In Fig. 2.7 the scaling efficiencies of 8 models are gathered from a large number of publications, selecting the most appropriate and favorable cases. Under 100 workers, it can be seen that models not designed from the root for parallel efficiency are often lagging behind. Indeed, they are often relying on shared memory methods like openMP, which are great for easy parallelization on one machine but cannot be ported to a computer cluster. They often display poor results, as shared memory methods increase the use of already shared resources. In the middle range, there are models that efficiently use parallel resources until they reach a breakpoint (see Section 2.6.2). It should be noted that the weak scaling test cases of PFLOTRAN are done on a fixed computational domain size, by reducing the elements size. In the extreme range, Parflow is the only model that can run on more than 100,000 workers.

[7] See also for the resources gathered by the International Soil Modeling Consortium https://soil-modeling.org

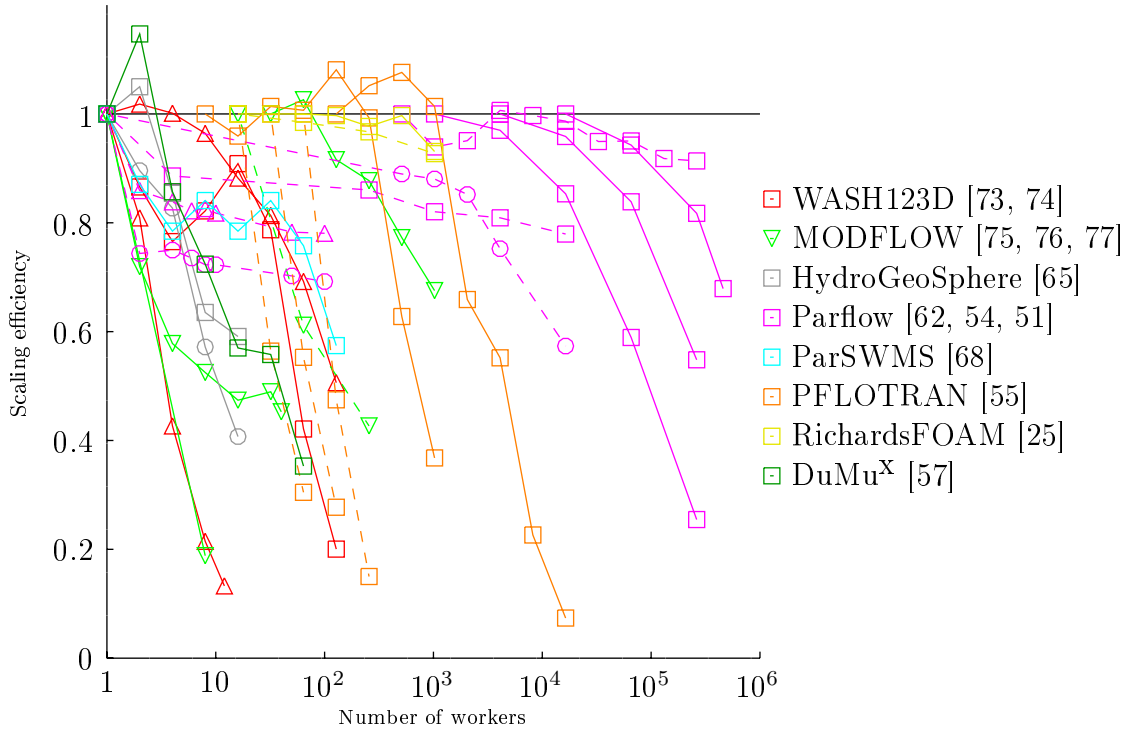


Figure 2.7: Scaling efficiency for terrestrial models. The strong scalings (continuous lines) are displayed along with the weak scalings (dashed lines), for models using the Richards equation (squares), a variant of the shallow waters (upward triangles), saturated zones only (downward triangles) or models coupling surface and subsurface (circles). The efficiency is computed based on the first data available in the series and the results are very sensible to it.

Although the graph in Fig. 2.7 is expected as it is often used to demonstrate scalability, it is not really meaningful. Indeed, even if we are assuming that all those tests are conducted on perfect machinery, and that the simulation complexities are similar, the number of DOF/w is often very high to boost the performances. Indeed, this increase the volume on surface ratio, which is directly related to the local computation over communication ratio (see Section 2.6.1). The point of reference of the scaling efficiency is also different between the different model simulations. This efficiency is thus misleading,

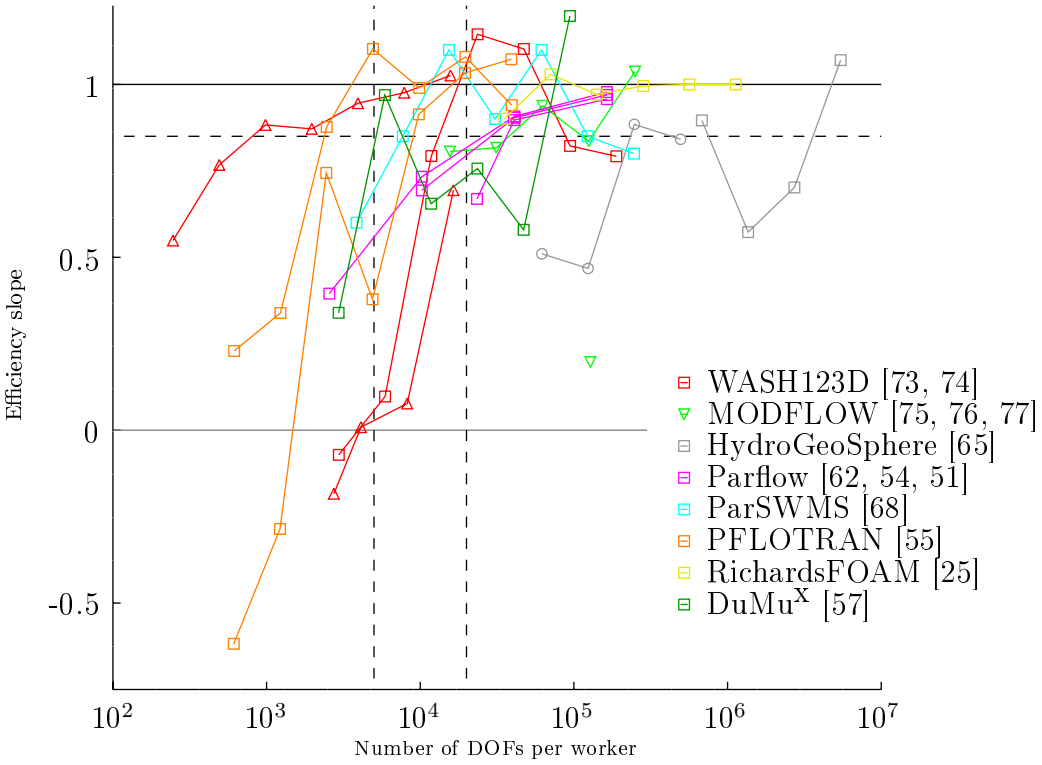


Figure 2.8: Strong scaling efficiency slopes for models of the Richards equation (squares), a variant of the shallow waters (upward triangles), saturated zones only (downward triangles) or models coupling surface and subsurface (circles). The value of one means a perfect scaling, values below zero an increase of computational time when increasing the number of workers. The dashed lines represent our limit at 0.85 to assess the scaling efficiency breakpoint and the range between 5000 to 2000 DOF/w where most models display their breakpoints.

even if we agree that it is not possible to produce a correct scaling under a whole node. What is interesting is the point at which the curve is breaking. It is why we introduce a new metric, the efficiency slope S, which is the slope in a $\log_2$ scale of the speed-up over the number of nodes:

$$S_j = \frac{\log_2(t_i/t_j)}{\log_2(j/i)} \tag{2.29}$$

with j the number of workers in use, t_j the simulation time and the equivalent for a reference point i. The best reference point is the first available with $i < j$. This metric removes the dependence to the base value for scaling efficiency. It is however a little more noisy as the simulation time derivatives are computed over sparse measurements. The same data as in the previous figure are displayed in Fig. 2.8 with this new metric over the number of DOF/w. A S of 1 is a perfect scaling while 0 means that there are no improvements when adding computational resources. Negative value of S indicate an increase of the simulation time when more nodes are added. Only strong scaling are displayed as weak scalings have a constant number of DOF/w and are not meaningful here. A efficiently-build model will reach high values of S even for low values of DOF/w. Therefore, this graph allows us to determine the scaling communication breaking point of a model. Here we consider that such breakpoint is reached when S drops below 85%. Being at this limit means that doubling the workers sees a 10% drop of scaling efficiency, and that 100 workers perform at 50% of a single worker efficiency. The model that performs the best is a surface models which benefits of its 2D spatial discretization (decrease of surface/volume ratio in comparison to 3D meshes). The petascale simulations of Parflow are using a consequent number of DOF/w and are staked on each other, showing the purpose of the choice for the x axis (this behavior is already pointed out in [51]). RichardsFOAM results are less impressive with the change of axis, but still very good. PFLOTRAN shows the best results here for subsurface models. Overall, the drop of efficiency for subsurface models is situated between 5000 and 20000 DOF/w.

2.8 Conclusion

The RE looks like a standard diffusion equation but it can exhibit an elliptical, parabolic or hyperbolic dynamics depending on the variables state. The highly non-linear constitutive relationships can lead to many numerical issues. Moreover, the simple ψ-form is not mass conservative. In addition to the error on the mass balance, mass conservation is especially important for subsequent tracers models using the water fluxes as an input.

A number of approximations of RE models have been introduced to cope with those issues. Mass lumping avoids variable oscillations near sharp gradients. Computing the constitutive relationships at nodes simplifies the linear system and avoids a numerical pitfall where the dynamic is stopped. The modified Picard iteration changes the non-linear solver of a mixed-form of RE to greatly improve mass conservation. A change of variable can soften the sharp relations between pressure and water content, yielding a better convergence. The variable switching method improves both convergence and mass conservation by combining the mixed and a ψ forms.

The infiltration front dynamics of RE is often not well suited to high order models. To improve the performances, it is more frequent to maximize the time step while keeping sufficient precision and stability, with different adaptive time step strategies. Recently, adaptive meshes have proved to be quite useful to represent dynamically-active local phenomena [47, 49, 40].

Identifying the purpose and consequences of each of those approximations and model designs allowed us to propose new model formulations, which are described in Chapters 3 and 5.

To enhance a model, it is important to know which approximation can be abandoned and which has proved to work. To find a good parametrization, it is necessary to achieve a simulation in an acceptable time, as it will be run multiple times. A good parallel efficiency can achieve that goal, in addition to allow larger computational domains and longer timeframes for real-scale simulations.

The parallelization of the RE is a rather new research topic that deserves attention. Indeed, many applications should be simulated at larger scale and finer resolution, which is not possible with a sequential program in reasonable computing time. The recent advances in that direction already lead to regular-grid simulations on petascale HPC clusters.

In this thesis, both models presented for RE use first-order discretizations in space and time, mass lumping to remove spurious oscillations, and element-wise linearization of the K and C functions to smooth out strong non-linear variations. The time integration schemes were designed to avoid any mass balance error, by iterating on the θ variable directly. The spatial discretization is done with unstructured meshes of various types of elements. Compared to the state-of-the-art methods presented in this chapter, our models do not rely on non-linear solvers (increasing the efficiency and robustness), are machine-precision mass conservative and efficient on parallel infrastructures.

There is still some room to enhance the efficiency and precision of RE models, especially in the domain of spatial adaptation to the problem state. As groundwater simulations concern both large scales (the watershed) and very small scales (the soil pores distribution), sub-modeling can bring the precision of the latter to the former. Eventually, better parallelization can be reached by selecting efficient linear solvers.

CHAPTER 3

A fully-explicit discontinuous Galerkin hydrodynamic model for variably-saturated porous media

This Chapter is based on the paper by De Maet, T., Hanert, E., Vanclooster, M., 2014. Journal of Hydrodynamics 26(4), 594-607 [6]. The computational resources at the time of writing this paper were more limited than now. In our local HPC a maximum of 64 cores were easily accessible for our test cases. We however expect our scaling results to remain valid on much more cores. The original manuscript has seen several modifications, the main ones being an homogenization of the variables thorough the document and a disambiguation of the basic computational unit (worker). We also made the following minor modifications: addition of a definition for the l_e parameter, additional explanation for the treatment of $[\theta^h]$, additional details on the values of p and on the mass balance achieved with Hydrus, removal of the term *adaptive time step* as it is often used for more complex techniques.

Abstract

Groundwater flows play a key role in the recharge of aquifers, the transport of solutes through subsurface systems or the control of surface runoff. Predicting these processes requires the use of groundwater models whose applicability is directly linked to their accuracy and computational efficiency. In this chapter, we present a new method to model water dynamics in variably saturated porous media. Our model is based on a fully-explicit discontinuous-Galerkin formulation of the 3D Richards equation, that shows a perfect scaling on parallel architectures. We make use of an adapted jump penalty term for the discontinuous-Galerkin scheme and of a slope limiter algorithm to produce oscillation-free exactly conservative solutions. We show that such an approach is particularly well suited to infiltration fronts. The model results are in good agreement with the reference model Hydrus-1D and look promising for large scale applications involving a coarse representation of saturated soil.

3.1 Introduction

A good understanding of subsurface water dynamics is essential in many hydrological, environmental and engineering applications. However, predicting such a dynamics is still a difficult task. The difficulty mainly comes from the heterogeneity of the soil properties, the non-linearity of the flow process and the absence of fast techniques to measure the hydraulic properties everywhere, at the appropriate scale. Given these issues, modeling the water flow in heterogeneous and often partially saturated porous media quickly and robustly is still a challenge.

By using a continuum approach, the water flow in variably saturated heterogeneous porous media can be modeled with the Richards equation (RE). This equation concerns both the water saturated zone (SZ) and the unsaturated zone (UZ). In the former, the equation models an incompressible fluid, leading to a constant water content θ. The pressure head ψ then reacts instantaneously to the boundary conditions. In the latter, the equation represents capillary physics and is complemented with the so-called retention-curve equation, which links the two variables. It is usual to refer to the θ-form and the ψ-form of the RE for the equations obtained by incorporating the retention curve in the conservation law to keep either θ or ψ as the sole model variable, respectively. On the one hand, the θ-form is not valid in SZ, but is known to be more efficient in the UZ. On the other hand, the ψ-form is valid everywhere, although it is not mass conservative once discretized in time. The issues associated with the θ- and ψ-forms are usually overcome by combining both variables into a mixed formulation. When considering only the SZ, the RE reduces to the groundwa-

ter equation that has its own numerical issues such as those associated with the treatment of the free surface [78, 79].

Despite the abundance of subsurface-flow models, the development of efficient and accurate discretizations of the RE is still an active field of research. The presence of two very different dynamics (in the SZ and UZ) leads to strong non-linearities in the constitutive relations and then in the coupling between parabolic and elliptic partial differential equations. Implicit schemes have difficulties to handle such non-linearities resulting in a lack of convergence [80]. Moreover, as the system dynamics can be highly transient, adaptive time steps are often mandatory [44]. In infiltration cases, this results in very small time steps, well below what is needed to reach a sufficient level of accuracy. As a result, the efficiency of implicit models for RE is generally suboptimal when infiltration fronts occur.

The efficiency of implicit models is further impaired by their often poor scaling on parallel architectures. The current trend in computer designs is indeed to increase the performances relying on parallel architectures instead of enhancing the computational power of each processor individually. Current subsurface flow models therefore have to take advantage of parallelism, and some steps in that direction have been made with ParFlow [62, 54], ParSWMS [68, 81] and DuMu$^{\text{X}}$ [57]. These models however only achieve sub-optimal scaling for RE, excepted for some fully unsaturated test cases [81]. This is due to the use of implicit solvers as they require a large amount of communication between computational nodes to solve linear systems. Unlike implicit solvers, explicit solvers require only one exchange of information between nodes per time step. As implementing an explicit solver is simple, one can easily achieve a super-linear scaling, *i.e.* a scaling better than 100%, thanks to the additional computer caches coming from additional resources. Moreover, explicit solvers do not require any global matrix linear solver or the computation of a Jacobian, which are complex and costly.

In this chapter, we present a three-dimensional model of RE based on the discontinuous Galerkin (DG) finite element (FE) method and an explicit time integration scheme. The model relies on slope limiters to locally ensure the monotonicity of the solution. It scales optimally at least up to 64 workers. A special treatment on the interface term between elements allows the existence of physical discontinuities in the water content between different types of soils. The model is mass-conservative at the machine precision. In sections 3.2 and 3.3 we present the mathematical formulation of the model and the explicit DG FE discretization. In section 3.4, we present 1D and 3D numerical results, which focus on different physical and numerical aspects of groundwater flows.

3.2 Mathematical formulation

The RE is obtained by embedding Darcy's law into a mass conservation equation:

$$\frac{\partial \theta}{\partial t} = \nabla \cdot (\mathbf{K} \cdot \nabla(\psi - z)) + s, \tag{3.1}$$

$$\theta = f_\theta(\psi), \tag{3.2}$$

where θ [–] is the volumetric soil water content, ψ [m] is the pressure head, z [m] is the upward positive vertical coordinate, s [s^{-1}] a sink-source term, $\mathbf{K}$ [ms^{-1}] the water conductivity tensor and f_θ [–] the retention curve. Eqs. (3.1) and (3.2) are complemented with appropriate initial and boundary conditions:

$$\psi = \psi_0, \qquad \text{on } \Omega, \quad t = 0 \tag{3.3}$$

$$\psi = \psi_D, \qquad \text{on } \Gamma_D, \quad t \in [0, T] \tag{3.4}$$

$$(\mathbf{K} \cdot \nabla(\psi + z)) \cdot \mathbf{n} = J_N, \qquad \text{on } \Gamma_N, \quad t \in [0, T] \tag{3.5}$$

with $\mathbf{n}$ the outward normal vector, T [s] the simulation duration, Ω the computational domain, Γ_D the Dirichlet part of the boundary (where the value ψ_D is imposed) and Γ_N the Neumann part (where the flux J_N is imposed). The constitutive relations defining $\theta = f_\theta(\psi)$ and $\mathbf{K}$ have been introduced by Van Genuchten and Mualem:

$$S_e = \frac{\theta - \theta_r}{\theta_s - \theta_r}, \tag{3.6}$$

$$S_e = \begin{cases} (1 + |\alpha\psi|^\beta)^{-\nu} & \psi < 0 \\ 1 & \psi \geq 0 \end{cases}, \tag{3.7}$$

$$\mathbf{K} = \mathbf{K}_s \; S_e^{l_p} \left(1 - (1 - S_e^{1/\nu})^\nu\right)^2, \tag{3.8}$$

where S_e [–] is the effective saturation, θ_r [–] is the residual volumetric water content, θ_s [–] is the saturated volumetric water content, $\mathbf{K}_s$ [m/s] is the anisotropic saturated water conductivity tensor, α [m^{-1}] is a parameter related to the air-entry pressure value, β [–] is a parameter related to the pore-size distribution, $\nu = 1 - 1/\beta$ and l_p is the pore-connectivity parameter, set to 0.5 in this work.

By splitting the domain between the UZ ($\psi < 0$) and the SZ ($\psi \geq 0$), the following adapted θ and ψ-forms can be derived:

$$\frac{\partial \theta}{\partial t} = \nabla \cdot \left(\frac{\mathbf{K}}{C} \cdot \nabla\theta - \mathbf{K} \cdot \nabla z\right) + s \qquad \text{in UZ}, \tag{3.9}$$

$$S_s \frac{\partial \psi}{\partial t} = \nabla \cdot (\mathbf{K}_s \cdot \nabla(\psi - z)) + s \qquad \text{in SZ}, \tag{3.10}$$

where $C = \partial f_\theta / \partial \psi$ [m^{-1}] is the capillary capacity and $S_s = C(\psi > 0)$ is the specific storage. Eq. (3.9) is parabolic and corresponds to a diffusion-reaction equation for θ. Eq. (3.10) is elliptic when S_s is negligible and then corresponds to a Poisson equation for $\psi - z$. Both equation types are well known and efficient discretization methods exist for both of them. However, since RE couples them in one single formulation and since in the UZ the RE can become advection-dominated, the numerical solution of RE remains challenging.

Here, we modify Eq. (3.10) by using the false transient method to obtain a parabolic equation that can be discretized with an explicit scheme. It is done by modifying the time derivative factor as follows:

$$\frac{K}{\tau}\frac{\partial \psi}{\partial t} = \nabla \cdot (\mathbf{K} \cdot \nabla(\psi - z)) + s \qquad \text{in SZ,} \tag{3.11}$$

where K is the largest eigenvalue of $\mathbf{K}$ and τ [$\mathrm{m}^2\mathrm{s}^{-1}$] is a free parameter that controls the relaxation towards the steady state defined by Eq. (3.10). When the general diffusivity tensor $\mathbf{K}$ is simply a scaled identity matrix (isotropic soil), it obviously reduces to K. In Eq. (3.11), the relaxation parameter τ can then be interpreted as the diffusivity. As such, it will constrain the stability of any explicit time discretization of that equation, *i.e.* when τ increases, the time step should decrease. Reciprocally, for a given time step, it is possible to determine the maximum value of τ that ensures stability. It should be noted that the approximation of Eq. (3.10) by Eq. (3.11) is only made for numerical purposes in order to deal with a system that includes a parabolic and an elliptic component. In the SZ, it physically amounts to increase the value of specific storage S_s.

If the ψ-form of Eq. (3.1) is used in zones where an explicit time integration scheme is stable and Eq. (3.11) is used otherwise, we can easily combine them as follows:

$$C_m \frac{\partial \psi}{\partial t} = \nabla \cdot (\mathbf{K} \cdot \nabla(\psi - z)) + s, \tag{3.12}$$

$$C_m = \max\left(C, K/\tau\right). \tag{3.13}$$

Eq. (3.12) is exact when $C_m = C$, which is verified in most of the UZ but not in the SZ (except if $K_s/\tau \leq S_s$, which is unlikely). In this case, Eq. (3.12) is mathematically identical to the original ψ-form. For practical purpose, the subset of the UZ where $C_m = C$ will be called the dry zone (DZ) and the subset where $C_m = K/\tau$ will be called the wet zone (WZ). Eq. (3.12) works in both the UZ and SZ with an upper threshold for the diffusion coefficient equal to τ, as shown in Fig. 3.1. Although the ψ-form of the equation is not mass-conservative once discretized in time, the whole algorithm is mass-conservative at the machine error precision (see section 3.3.2).

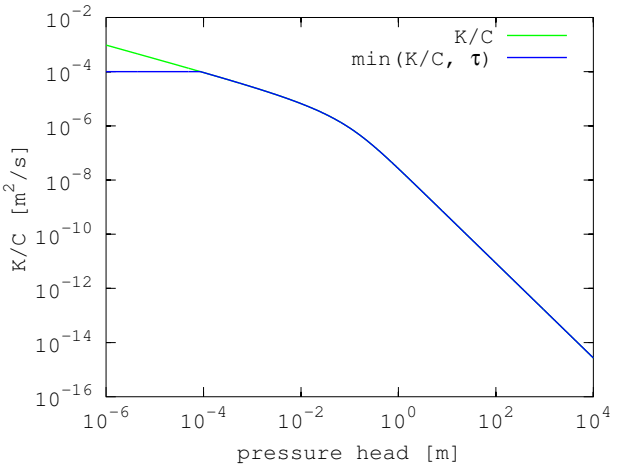


Figure 3.1: The effective diffusion K/C has to be limited by a threshold τ for the stability of the explicit solver.

In the time-stepping algorithm described in the next section, the model equations are complemented with the following equation:

$$\frac{\partial \theta}{\partial t} = \lambda \nabla^2 (\theta - f_\theta(\tilde{\psi})), \tag{3.14}$$

where λ is a free parameter and $\tilde{\psi}$ is the value of ψ obtained after the time integration of Eq. (3.12), refreshed at each time step before the resolution of Eq. (3.14). Eq. (3.14) allows us to spread out, in a mass-conservative way, the possible overshoots that can be caused by the approximation made in Eq. (3.13).

A weak form of the model equations is obtained by multiplying Eqs. (3.1), (3.12) and (3.14) by test functions u, v and $w \in H^1(\Omega)$. Taking the volume integral over the domain Ω and using the divergence theorem, we obtain the following weak formulations of RE:

$$\begin{aligned} \left\langle \frac{\partial \theta}{\partial t} u \right\rangle_\Omega &= \langle (\mathbf{K} \cdot \nabla(\psi - z)) \cdot \nabla u \rangle_\Omega \\ &\quad - \langle\!\langle \mathbf{n} \cdot (\mathbf{K} \cdot \nabla(\psi - z))\ u \rangle\!\rangle_{\partial\Omega} + \langle su \rangle_\Omega , \end{aligned} \tag{3.15}$$

$$\begin{aligned} \left\langle C_m \frac{\partial \psi}{\partial t} v \right\rangle_\Omega &= \langle (\mathbf{K} \cdot \nabla(\psi - z)) \cdot \nabla v \rangle_\Omega \\ &\quad - \langle\!\langle \mathbf{n} \cdot (\mathbf{K} \cdot \nabla(\psi - z))\ v \rangle\!\rangle_{\partial\Omega} + \langle sv \rangle_\Omega , \end{aligned} \tag{3.16}$$

which are coupled with the weak form of Eq. (3.14)

$$\begin{aligned}\left\langle \frac{\partial \theta}{\partial t} w \right\rangle_{\Omega} = & \left\langle \lambda \nabla (\theta - f_{\theta}(\tilde{\psi})) \cdot \nabla w \right\rangle_{\Omega} \\ & - \left\langle\!\!\left\langle \mathbf{n} \cdot \lambda \nabla (\theta - f_{\theta}(\tilde{\psi}))\ w \right\rangle\!\!\right\rangle_{\partial\Omega},\end{aligned} \tag{3.17}$$

where $\mathbf{n}$ is the outward normal vector, $\langle \cdot \rangle_{\Omega}$ is the volume integral over Ω and $\langle\!\langle \cdot \rangle\!\rangle_{\partial\Omega}$ is the integral over $\partial\Omega$ the boundary of Ω.

3.3 Space and time discretizations

The model equations are now discretized in space, by means of a discontinuous Galerkin FE scheme, and then in time with the Euler explicit time integration scheme.

3.3.1 Discontinuous Galerkin space discretization

Partitioning the domain Ω into N non-overlapping elements Ω_e with interface Γ_e, the model variables θ and ψ can be approximated as

$$\theta \simeq \theta^h = \sum_{j=1}^{N_d} \theta_j \phi_j, \qquad \psi \simeq \psi^h = \sum_{j=1}^{N_d} \psi_j \phi_j, \tag{3.18}$$

where N_d is the total number of degrees of freedom (DOF) and ϕ_j are piecewise polynomials defined on each element Ω_e such that

$$\phi_j(\mathbf{x}_i) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \text{ and } \sum_{j=1}^{n_e} \phi_j(\mathbf{x}) = 1 \quad \forall \mathbf{x} \in \Omega_e, \tag{3.19}$$

where $\mathbf{x}_i$ is the vector of coordinates for the node i and $n_e = N_d/N$ is the number of DOF per element. Here we use piecewise linear (P_1) basis functions for both variables. Since the discrete solution can exhibit discontinuities between mesh elements, the following jump $[\cdot]$ and averaging $\{\cdot\}$ operators on the interface Γ_e have to be introduced:

$$[x] = x^+ - x^-, \qquad \{x\} = \frac{x^+ + x^-}{2}, \tag{3.20}$$

where the '+' superscript indicates the trace value taken on one side of Γ_e and the '–' superscript indicates the trace value on the other side, from the second element. At the boundaries, both operators are defined in terms of an external value derived from the Dirichlet boundary condition. The resulting

weak boundary conditions are known to be more stable than strong ones [82, 83].

The discrete equations are obtained by replacing the test functions u, v and w in Eqs. (3.15) and (3.16) by basis functions ϕ_i ($1 \leq i \leq N_d$), and the solution θ and ψ by θ^h and ψ^h, respectively. Since θ^h and ψ^h are discontinuous between mesh elements, integration by part is performed over the partition of Ω into elements e of extent Ω_e and the interior penalty DG FE method is applied. The discrete equations thus read:

$$M_{ij}\frac{d\theta_j}{dt} = K_{ij}\psi_j + Z_i + P_i, \qquad 1 \leq i \leq N_d, \tag{3.21}$$

$$C_{m,j}M_{ij}\frac{d\psi_j}{dt} = K_{ij}\psi_j + Z_i + P_i, \qquad 1 \leq i \leq N_d, \tag{3.22}$$

$$M_{ij}\frac{d\theta_j}{dt} = \lambda_{ij}(\theta_j - f_\theta(\tilde{\psi}_j)) + P_i^\theta, \qquad 1 \leq i \leq N_d, \tag{3.23}$$

where $C_{m,j}$ is the C_m function evaluated on the j node, P_i and P_i^θ are penalty terms described further,

$$M_{ij} = \langle \phi_i \phi_j \rangle, \tag{3.24}$$

$$K_{ij} = \langle (\mathbf{K}\cdot\nabla\phi_j)\cdot\nabla\phi_i \rangle - \langle\!\langle \{(\mathbf{K}\cdot\nabla\phi_j)\cdot\mathbf{n}\}\,[\phi_i] \rangle\!\rangle, \tag{3.25}$$

$$Z_i = \langle -(\mathbf{K}\cdot\mathbf{z})\cdot\nabla\phi_i + s\phi_i \rangle + \langle\!\langle \{(\mathbf{K}\cdot\mathbf{z})\cdot\mathbf{n}\}\,[\phi_i] \rangle\!\rangle, \tag{3.26}$$

$$\lambda_{ij} = \langle \lambda\nabla\phi_i\cdot\nabla\phi_j \rangle - \langle\!\langle \{\lambda\ \nabla\phi_j\cdot\mathbf{n}\}\,[\phi_i] \rangle\!\rangle, \tag{3.27}$$

$$\langle\cdot\rangle = \sum_{e=1}^{N}\int_{\Omega_e}\cdot\, d\Omega, \qquad \langle\!\langle\cdot\rangle\!\rangle = \sum_{e=1}^{N}\int_{\partial\Omega_e}\cdot\, d\Gamma, \tag{3.28}$$

where $\partial\Omega_e$ is the contour of the element domain Ω_e. The material properties are assumed constant by element.

The DG FE method uses a piecewise linear approximation that allows discontinuities between mesh elements. The approximation however becomes continuous when the solution is smooth. Discontinuities between mesh elements usually appear when the spatial resolution is insufficient to represent sharp gradients such as the one appearing in infiltration fronts. In this case, the size of the jumps between the elements gives a direct information of the local spatial error. To stabilize the diffusion term and enforce a weak continuity constraint between elements, penalty terms P_i and P_i^θ have to be added to Eqs. (3.21), (3.22) and (3.23) [84]. The continuity constraint on ψ can be expressed as:

$$P_i = \sigma \left\langle\!\!\left\langle [\phi_i] \left[\psi^h\right] \right\rangle\!\!\right\rangle, \tag{3.29}$$

where

$$\sigma = \frac{4}{l_e}\ (n_0+1)(n_0+2)\ K_{sc}, \tag{3.30}$$

$$K_{sc} = (\mathbf{K}\cdot\mathbf{n})\cdot\mathbf{J}/\|\mathbf{J}\|, \tag{3.31}$$

$$\mathbf{J} = \mathbf{K}\cdot\nabla(\psi - z), \tag{3.32}$$

with n_0 the order of the FE approximation (in our case $n_0 = 1$), K_{sc} the normal flux-oriented scalar conductivity, $\mathbf{J}$ the water flux, and l_e a characteristic length of the two adjacent elements (defined as the fraction of the minimum volume between the two elements over their interface area). It should be noted that the exact solution for the pressure head ψ is continuous, while the water content θ can be discontinuous between different soil horizons. One of the advantages of DG FE method is to have the possibility to represent these discontinuities. However, imposing a continuity constraint on θ in a classical way would tend to smooth them out. The continuity constraint is thus only imposed on ψ with Eq. (3.29) to indirectly stabilize the Eq. (3.21). For Eq. (3.23), the continuity constraint P_i^θ is expressed as follows:

$$P_i^\theta = \sigma_d \left\langle\left\langle [\phi_i]\left[\theta^h - f_\theta(\tilde{\psi})\right]\right\rangle\right\rangle, \tag{3.33}$$

where σ_d is similar to σ but with K_{sc} replaced by λ. The presence of $f_\theta(\tilde{\psi})$ weakly ensures similar jumps between the couples $f_\theta(\tilde{\psi}^+)$ – $f_\theta(\tilde{\psi}^-)$ and $\theta^{\psi+}$ – $\theta^{\psi-}$. A special treatment is needed for $[\theta^h]$ at interfaces between media with different hydrological properties. To be able to handle those physical discontinuities of θ, the treatment consists in modifying θ by first transforming it into an intermediate pressure head variable (physically continuous), then derive a new θ which is compatible which its neighbor value. The resulting relation is described at the end of section 3.3.5.

To increase the stability, the mass matrices M_{ij} in Eqs. (3.21) and (3.22) have been lumped. Hence $C_{m,j}$ is only present on the diagonal, which allows us to avoid inverting the product $C_{m,j}M_{ij}$ at each time step. Following other model designs, we use nodal element values for $\mathbf{K}$.

An element is considered affiliated to the SZ, the WZ or the DZ in the following priority order:

1) if one node of the element is saturated (e.g. $\psi >= 0$), the element belongs to the SZ,

2) if one node of the element has the value $C_m < C$, the element belongs to the WZ,

3) otherwise the element belongs to the DZ.

3.3.2 Mass conservation

As mentioned before, the pure ψ-form of the RE is the easiest to solve but it is not mass-conservative. This is due to the time-discretization of the C function which is highly non-linear. Here we propose a simple solution to this issue based on the application of Eq. (3.1) as a post-processing step to achieve mass-conservation.

Using the same discretization as for Eq. (3.1) and an a-priori non-conservative ψ-form as Eq. (3.12), we can obtain two equations with the same right-hand side:

$$M\frac{d\theta}{dt} = T(\psi), \tag{3.34}$$

$$C(\psi)\frac{d\psi}{dt} = T(\psi), \tag{3.35}$$

where M, C, T are the matrices resulting from the spatial discretization. As M is constant in time, one could see that Eq. (3.34) will produce exactly-conservative results for any value of ψ, as the Darcy flux is included in a divergence. On this basis, we can compute an approximate solution ψ^* of Eq. (3.35) and then Eq. (3.34) could be solved with $T(\psi) \simeq T(\psi^*)$. With such a method, mass is conserved at machine precision. The opposed approach has to be compared to models based on the classical chord-slope method [27, 24], for which the mass-error decreases with the solution-error along the successive non-linear iterations. Using an explicit iteration scheme to ensure mass conservation could theoretically be applied to any non-conservative implicit method.

3.3.3 Explicit time discretization

Eqs. (3.21) and (3.22) are discretized in time with an explicit Euler scheme. Using a matrix notation, the overall solution procedure is the following:

1. Solve the equation for the intermediate steps $\Psi^{n+1,k}$, given $\Psi^{n+1,0} = \Psi^n$:

$$\text{for } k = 1, \ldots, m: \tag{3.36}$$

$$C_m M\frac{\Psi^* - \Psi^{n+1,k-1}}{\Delta t/m} = K\Psi^{n+1,k-1} + Z + P, \tag{3.37}$$

$$\Psi^{n+1,k} = \text{limit}(\Psi^*), \tag{3.38}$$

   where the limiting procedure is described in section 3.3.5.

2. Solve the equation for $\Theta^{n+1,0}$, given Θ^n and $\Psi^{n+1,m}$:

$$M\frac{\Theta^* - \Theta^n}{\Delta t} = K\Psi^{n+1,m} + Z + P, \tag{3.39}$$

$$\Theta^{n+1,0} = \text{limit}(\Theta^*). \tag{3.40}$$

3. Compute Ψ^{n+1} by combining $\Psi^{n+1,m}$ and the retention curve relationship:

$$\Psi^{n+1} = \begin{cases} f_\theta^{-1}(\Theta^{n+1,0}) & \text{in DZ,} \\ \Psi^{n+1,m} & \text{in WZ and SZ.} \end{cases} \tag{3.41}$$

4. Weakly correct $\Theta^{n+1,0}$ in WZ and SZ:

$$\text{for } k = 1, \ldots, q: \tag{3.42}$$

$$M \frac{\Theta^{n+1,k} - \Theta^{n+1,k-1}}{\Delta t} = \Lambda(\Theta^{n+1,k-1} - f_\theta(\Psi^{n+1})) + P^\theta, \tag{3.43}$$

$$\Theta^{n+1} = \text{limit}(\Theta^{n+1,q}) \tag{3.44}$$

where we have replaced the reference value $\tilde{\Psi}$ used in Eq. (3.23) by Ψ^{n+1}.

In the DZ, this algorithm reduces to the θ-form of RE. This form is known to be numerically better suited than the ψ-form, especially in very dry cases. In the SZ, it reduces to an approximation of the ψ-form which exponentially converges to Eq. (3.10) as m increases. The resulting approximation error leads to approximate fluxes in the SZ but the total mass is exactly conserved. Step 4 allows us to bring θ towards its correct value in both SZ and WZ.

It should be noted than when Eq. (3.37) has converged in the SZ, the result is equal to the one obtained with an implicit time integration scheme. Indeed, if $\Psi^{n+1,m} = \Psi^{n+1,m-1}$, we have reached the incompressible state which corresponds to the solution of the implicit equation for ψ, and also for θ once inserted in Eq. (3.39). To satisfy this property, the right-hand sides of Eqs. (3.21) and (3.22) have to be identical. Another point is that increasing m directly decreases the strength of the approximation. Indeed, the division of Δt by m in Eq. (3.37) allows us to magnify τ by a factor m.

3.3.4 Selecting the values of τ and λ

Any explicit time integration scheme of Eq. (3.21) is fully mass-conservative and valid both in UZ and SZ but requires the field ψ_j to be known. Inverting the retention curve, *i.e.* $\psi = f_\theta^{-1}(\theta)$, is the easiest way to obtain it. However there is an underlying Courant-Friedrich-Lewy (CFL) stability condition. For instance, the explicit Euler time discretization reads:

$$\psi^n = f_\theta^{-1}(\theta^n), \tag{3.45}$$

$$\theta^{n+1} = \theta^n + \Delta t F(\psi^n), \tag{3.46}$$

where the superscripts indicate the time step, Δt represents the time step and $F(\psi)$ the right-hand side of Eq. (3.11). One could see that this collapses exactly

into the θ-form, and simple manipulations show that this is unstable when K/C exceed the stability limit for diffusion, which is:

$$\frac{K}{C} \leq p\frac{\Delta l^2}{\Delta t}, \tag{3.47}$$

where Δl is the smallest element length and $0 < p < 1$ depends on the type of explicit solver used. Although our algorithm is slightly more complex than Eqs. (3.45) and (3.46), we hypothesize that its stability condition is similar and, with Eq. (3.13), it leads to the following condition on τ:

$$\tau = \left.\frac{K(\psi)}{C(\psi)}\right|_{\psi=\psi_{max}} \leq p\frac{\Delta l^2}{\Delta t}, \tag{3.48}$$

where ψ_{max} is soil-dependent.

The parameter τ controls the position of the separation between DZ and WZ. As the solution in the WZ is approximated, it is better to limit its extent as much as possible by increasing τ. However, to respect Eq. (3.48) for a constant mesh size Δl, increasing τ leads to decreasing the time step. In the following, we use the experimentally-found optimal values $p = 1/5$ in 1D and $p = 1/15$ in 3D.

The quasi-elliptic approximation produces fluxes in the SZ resulting in a local mass excess or deficit that has to be corrected with Eq. (3.43). This equation has the following stability condition:

$$\lambda \leq p_\theta\frac{\Delta l^2}{\Delta t}. \tag{3.49}$$

To optimize the correction, we take the explicit diffusive limit, *i.e.* we take the larger stable value of λ given the sizes of the adjacent elements, in a similar way as for τ. It has been observed than a smaller value $p_\theta = 1/20$ has to be taken for stability, due to complex interactions with the UZ/SZ interface.

The present model could develop unphysical fluxes in WZ and SZ because of the approximation in Eq. (3.12). A-priori, the parameters p and p_θ are fixed once for all, and τ and λ could then be fixed by the choice of Δt. The free parameters which act on the approximation are then Δt, m and q. One could obtain an approximation of the local error from the false transiant method:

$$e = (C_m^n - C^n)\frac{(\psi^{n+1} - \psi^n)}{\Delta t}, \tag{3.50}$$

where e is close to the mass difference error applied to θ at the time step n. Limiting this error below a certain threshold could then be achieved by adapting the values of m and q.

3.3.5 Slope limiters

As mentioned previously, FE schemes of the RE can produce spurious oscillations. To avoid these, we have used a technique designed for advection-dominated problems that consists in locally modifying the slope of the solution when it does not respect the monotonicity of the solution. By doing so, we aim to achieve a total variation diminishing in the mean property on each element [85]. The slope limiting algorithm used is the same as in [86]. It simply checks that the solution at the nodes of an element e is bounded by the means of the solution in the neighboring elements $e_1 \dots e_{n_g}$. For each mesh node j, we collect the extrema max_j and min_j of the means of all the elements including the node j. The limited value U^* of a variable U over one element e could then be expressed as:

$$U^*_{ej} = \overline{U}_e + L_e \, (U_{ej} - \overline{U}_e), \tag{3.51}$$

$$L_e = \min(1, L^*_e), \tag{3.52}$$

$$L^*_e = \min_{j \in e} \left[\frac{extr_{e,j} - \overline{U}_e}{U_{ej} - \overline{U}_e} \right], \tag{3.53}$$

$$extr_{e,j} = \begin{cases} max_j, & \text{if } \; U_{ej} - \overline{U}_e > 0 \\ min_j, & \text{if } \; U_{ej} - \overline{U}_e < 0 \end{cases} \tag{3.54}$$

where overlines represent the mean over one element, the indices e or j the affiliation of the variable to the element e or to the node j. This expression is correct and mass conservative only if the element has the property that the sum of its nodes weighted by their number is the mean of the element, *i.e.* $\sum_j U_{ej} = n_e \overline{U}_e$. All elements present in the following test cases respect this rule.

The limited values U^*_{ej} are unchanged (*i.e.* $L_e = 1$) if the values inside the element (*i.e.* $U_{ej}, \forall j \in e$) are between the maximum and the minimum of the means of their neighbors. Otherwise, the value of L_e is less than one, and the reduction on the gradients applies in all directions. It would be possible to increase the accuracy of this limiter by considering separately each spatial direction and space derivative. However, the simplicity of this algorithm has been preferred to reduce the computational cost. A 1D example is shown in Fig. 3.2.

A different behavior is expected on the boundaries of the domain. Only tangential components to the boundary have to be limited while the normal one should not. Indeed, we want to stabilize the scheme, without limiting the extrema on the boundaries as such these appear frequently in physical cases. This is achieved by using a mirror image of the solution outside of the domain.

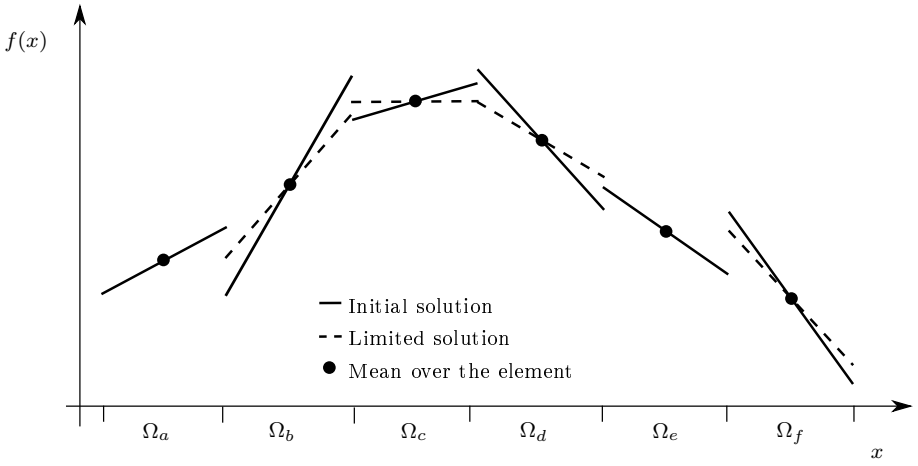


Figure 3.2: One-dimensional exampleof the slope limiter behavior. The slope of the solution on elements located at local extrema (as Ω_c) is set to zero, otherwise the solution is bounded by the means of solutions in neighboring elements, without modifying the local mass (*i.e.* the mean remains constant). In this example, the solution slope is limited in $\Omega_b, \Omega_d, \Omega_f$ but not in Ω_a and Ω_e.

Eventually, special care must be taken for the limiter and continuity constraint applied to θ. Indeed, both have to preserve the physical discontinuities of θ between different media A and B. The way to proceed is to translate θ into ψ by means of the inverse of the retention curve $f_\theta^{-1}(\theta)$, as ψ has the property to be physically continuous. Once θ has been translated in term of ψ with the properties of the medium it belongs to (*i.e.* A or B), it is translated back into θ using the properties of the medium where the computation is necessary (resp. B or A). Then, if subscripts are used to represent the medium, one could define

$$2[\theta^h] = \left(\theta_A^+ - f_{\theta_A}(f_{\theta_B}^{-1}(\theta_B^-))\right) + \left(f_{\theta_B}(f_{\theta_A}^{-1}(\theta_A^+)) - \theta_B^-\right), \tag{3.55}$$

$$min_j^A = f_{\theta_A}\left(\min_{e \ni j}\left(f_{\theta_e}^{-1}(\overline{U}_e)\right)\right), \tag{3.56}$$

$$max_j^A = f_{\theta_A}\left(\max_{e \ni j}\left(f_{\theta_e}^{-1}(\overline{U}_e)\right)\right), \tag{3.57}$$

as the continuity constraint and extrema per node. The differences for the continuity constraint and the means for the limiter are then coherent with physical discontinuities of θ.

3.4 Numerical examples

In this section, we present three numerical examples demonstrating the ability of the model to produce results similar to the widely-used model Hydrus-1D [87], confirming its convergence as the number of iterations increase, and showing its scalability on a 3D application.

3.4.1 Unsaturated infiltration

This simulation is based on data numerically reproduced as a benchmark test in the Hydrus-1D code [87]. The experimental setup consists in a homogeneous column of soil of 60 cm that is assumed to have an initial constant pressure head of -1.5 m. The characteristics of the soil are the same as in the Hydrus code and are represented by the modified Van Genuchten–Mualem relations from Vogel and Cislerova [88] summarized in Appendix A.2, with the parameter values given in Table 3.1. The material properties are homogeneous and isotropic. At the beginning of the simulation, the pressure of a thin layer of water whose height is assumed to be approximately equal to zero is imposed at the top of the column. Mathematically, this is done by imposing the Dirichlet boundary condition $\psi(z = 0) = 0$. A zero-flux boundary condition is imposed on all the other boundaries. The spatial discretization is made with 30 equidistant layers in the vertical direction. Here, a time step of 1 s was used, with only one sub-iteration for ψ and θ (*i.e.* $m = q = 1$).

Table 3.1: Modified Van Genuchten parametrization for the unsaturated infiltration test case (see Appendix A.2).

Sand	α [m^{-1}]	β	θ_r	θ_s
	4.1	1.964	0.02	0.35
θ_m	θ_k	θ_a	K_s [ms^{-1}]	K_r [ms^{-1}]
0.35	0.2875	0.02	7.22×10^{-6}	6.95×10^{-6}

Pressure head profiles at different instants of the simulation are displayed in Fig. 3.3. The maximum over one element remains below the means of the neighboring elements thanks to the slope-limiting algorithm. It can be seen that our results are very close to the ones obtained with Hydrus-1D [87]. In this test case, the false transient approximation does not apply. Indeed, all the elements except the top one lie in the DZ. For most natural infiltrations, dynamics happen in the DZ and are therefore not approximated.

The mass balance error of Hydrus at the end of the simulation is 1.95×10^{-6} m versus 2×10^{-16} m for our code (with a maximum at 6×10^{-16} m). To produce this result the default convergence parameters of Hydrus are used. The error is

expected to drop with more non-linear iterations. However the actual difference with our model is that our mass balance error is always minimal and does not depend on a convergence criterion.

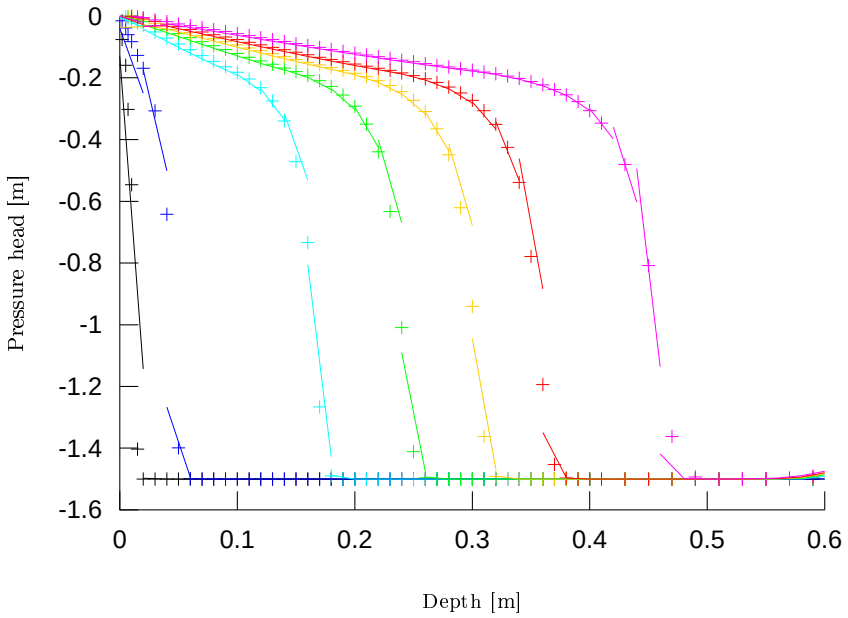


Figure 3.3: Pressure head profiles for an infiltration in an initially dry sandy soil at simulation times t= 1/12, 1, 15, 30, 45, 60, 90 minutes, from left to right. Our model results are represented by solid lines, and Hydrus-1D results by the '+' signs.

Spatial or temporal convergence studies are difficult to fulfill with the proposed model. Indeed classical convergences orders are disrupted by the false transient approximation in WZ and SZ, as shown in the next section. Those zones should therefore be limited to isolate proper convergences when keeping m and q constant as they are not the studied variables. Since the size of the WZ increases with the time step and decreases with the mesh size, we have to impose a maximum time step of 1 sec and minimum mesh size of 1 cm. For the mesh convergence study, the top fifth of the domain is kept fixed with a discretization of 2 cm to make sure the WZ is always discretized with the same resolution and hence prevent the false transient approximation from interfering with the convergence analysis. Fig. 3.4 shows a temporal convergence rate of 1 and a spatial convergence rate of 1.5. The first-order explicit time integration scheme behaves as expected but it is not the case of the spatial integration

scheme which is theoretically second-order accurate. Such a discrepancy is likely due to the slope limiters.

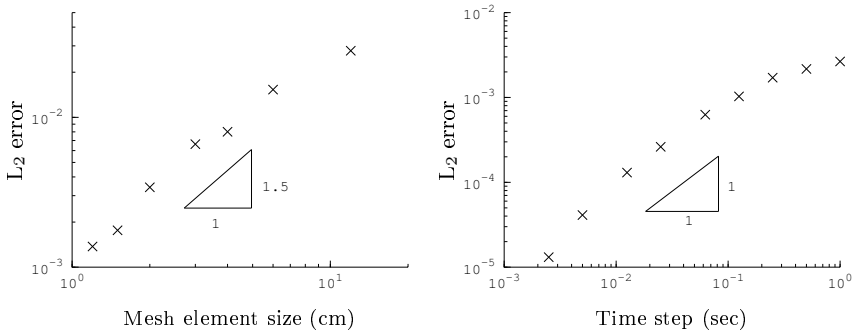


Figure 3.4: Mesh element size and time step convergence studies for the Unsaturated infiltration test-case. The L_2 norm error is computed from the finest runs (1cm for the mesh element size and 1.2510^{-3}sec for the time step).

3.4.2 Filling of aquifer

This second test case highlights the effects of the "quasi-elliptic" approximation in the SZ given by Eq. (3.11). We consider a soil of one-meter depth described by four equally-thick layers of sand, loam, clay and loam whose properties are given in Table 3.2. Each layer is assumed homogeneous and isotropic. Initially, the water table is located at half of the soil depth and the system is at physical equilibrium. This is achieved by taking a linear initial pressure field going from 0.5 m at the bottom to -0.5 m at the top. A Neumann boundary condition is imposed at the top to represent a strong infiltration of 10^{-5} ms^{-1} that stops after 2 h. All the other boundaries are impervious. The soil column is discretized into 40 equidistant layers, the time step was equal to 1 s.

Table 3.2: Van Genuchten parameter values for the aquifer filling test case.

	α [m^{-1}]	β	θ_r	θ_s	K_s [ms^{-1}]
Sand	14.5	2.68	0.045	0.43	$8.25\ 10^{-5}$
Loam	3.6	1.56	0.078	0.43	$2.89\ 10^{-6}$
Clay	0.8	1.09	0.068	0.38	$5.56\ 10^{-6}$
Loam	3.6	1.56	0.078	0.43	$2.89\ 10^{-6}$

Fig. 3.5 shows the evolution of the ψ and θ profiles during the simulation. The discontinuity between different types of soils in the θ-profile is in good

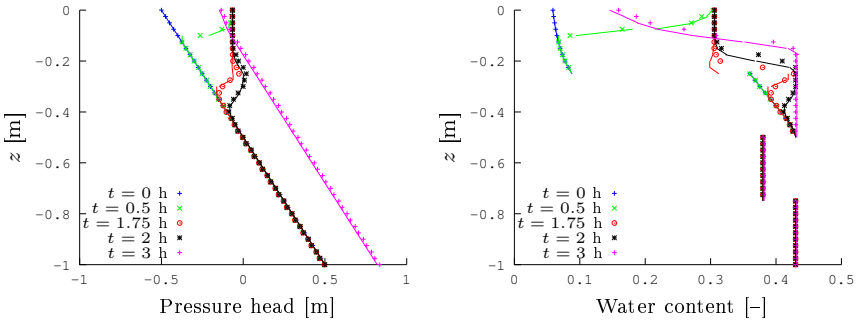


Figure 3.5: Infiltration profiles in a soil column initially at equilibrium. Our model results are represented by solid lines, while Hydrus-1D results are represented by the '+', '×', 'o' and '∗' markers. Our model results are very close to those obtained with Hydrus until $t \simeq 2.15$h. Afterwards, a slight discrepancy appears due to the approximation (3.11).

agreement with the physics, while a classic continuous representation would have to rely on the mean of the soil properties at the interfaces. In this model each DOF belongs to a particular material and all properties are well defined, even at the interfaces between different soil layers.

The results are similar for both models until the infiltration front reaches the water table depth, at around $t = 2.15$ h. Before this point, we set $m = q = 1$.

The succession of the large conductivity of the sand and the conductivity of the loam inferior to the incoming flux causes an accumulation of water at this interface. However, some water passes through this capillary barrier and eventually fills the loam layer. Once the loam layer has been filled, the depth of the incompressible water column instantaneously increases from half to nearly 3/4 of the computational domain.

After $t = 2.15$, the ψ-profile is not instantaneously adapted as it should, and a curvature appears, as shown in Fig. 3.6. This generates fluxes in the SZ leading to an increase in the water content, which is visible in the last panel of Fig. 3.5. However, this excess is spread over time and the model converges towards the steady solution. Increasing m and q improves the convergence of the solution, as shown in Fig. 3.6 and Fig. 3.7. It can be seen that a value of $m \simeq 25$ is sufficient on this quick event to transport the information through 30 saturated elements.

The mass balance error of Hydrus-1D at the end of the simulation is 4.78×10^{-6} m versus 5×10^{-16} m for our code (with a maximum at 10^{-15} m). For this

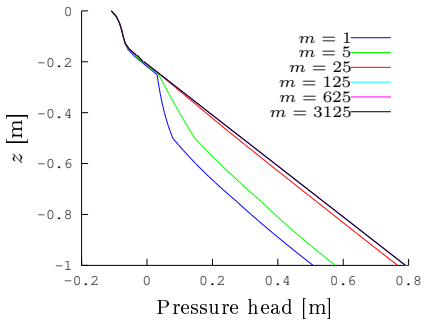


Figure 3.6: Pressure profile at $t = 2.2$ h. The shift results from the approximation (3.11), which replaces an elliptic equation by a parabolic one. This approximation converges to correct the solution as m increases. There are no visible differences for $m = 125, 625$ and 3125.

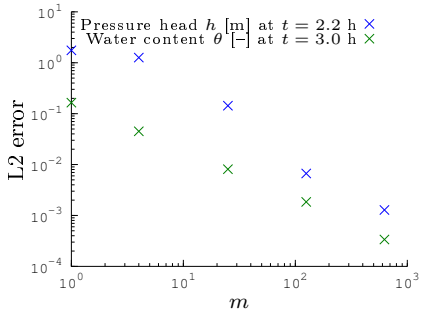


Figure 3.7: Error convergence computed as the L_2 norm of the difference with a solution where $m = 3125$. The chosen times are such as the error is maximal.

result, the limiter on θ has been removed. It adds robustness to the algorithm but is not mandatory. The limiter applies some additional multiplications and additions to the mass variable θ, which increase the mass balance error. Despite that, with the limiter the mass balance error is not larger than 1.2×10^{-14} m.

3.4.3 Capillary barrier in a simple landfill design

For this theoretical test case, a simulation of the water dynamics within the simple landfill represented in Fig. 3.8 is considered. The waste is stocked over a capillary barrier made of sand and gravel. An impervious geotextile placed under the gravel diverts the water to the bottom where it is drained. The

efficiency of the capillary barrier is assessed for an important flux of water into an initially dry system. We take advantage of the symmetries of the computational domain to model the problem only over one eighth of the domain. The waste layer is defined by soil-like properties that are given in Table 3.3 with the properties of the sand and gravel. The material properties are considered homogeneous and isotropic within each layer. The mesh is constituted of nine layers of 2,892 prisms for a total of 156,168 DOF, refined over corners as shown in Fig. 3.8. Water input is taken as a homogeneous flux of 5 mm/h on the top of the waste that is stopped after 12h. The geotextile is assumed to be perfectly impervious while the drain continuously covers the bottom of the landfill. The drain boundary condition is defined as follows:

$$\begin{aligned} &\text{if } \psi < \psi_c \text{ or flux} < 0: && \text{zero flux (Neumann)},\\ &\text{if } \psi \geq \psi_c \text{ and flux} \geq 0: && \psi = \psi_c \text{ (Dirichlet)}, \end{aligned}$$

where ψ_c is the air-entry pressure of the gravel layer. It should be mentioned that such a boundary condition is quite difficult to apply in an implicit model. Indeed, conditional statements frequently produce oscillatory states into the convergence process, as a continuous and monotone system is theoretically required to ensure convergence. The mesh used is made of prisms obtained by extruding a 2D triangular mesh. Here, we set $m = q = 1$ and use a time step which varies between 50 s at the beginning of the simulation to 0.075 s when the gravel, which is very conductive, reaches saturation. The time step is set as the minimum between 50 s and the CFL condition.

Table 3.3: Van Genuchten parameters for the capillary barrier test case.

	α [m^{-1}]	β	θ_r	θ_s	K_s [ms^{-1}]
Waste	1.43	1.51	0.032	0.345	$2.78\ 10^{-6}$
Sand	6.34	1.53	0.046	0.345	$6.57\ 10^{-5}$
Gravel	469.	2.57	0.074	0.419	$3.50\ 10^{-3}$

Water infiltration at the beginning of the simulation is visible on Fig. 3.9. As expected, water crosses the capillary barrier first at the external lower corner of the domain and at the bottom of the slope, where it accumulates before being drained out, as shown on Fig. 3.10. In the middle of the slope plane, the capillary barrier plays its role and diverts most of the water which slowly flows to the bottom of the sand layer, towards the drainage zone. The effect of the capillary barrier is also visible at the top of the corner, where a small area stays dry. On such a configuration, the most sensitive parts are the corners where the geotextile has to divert the strongest fluxes.

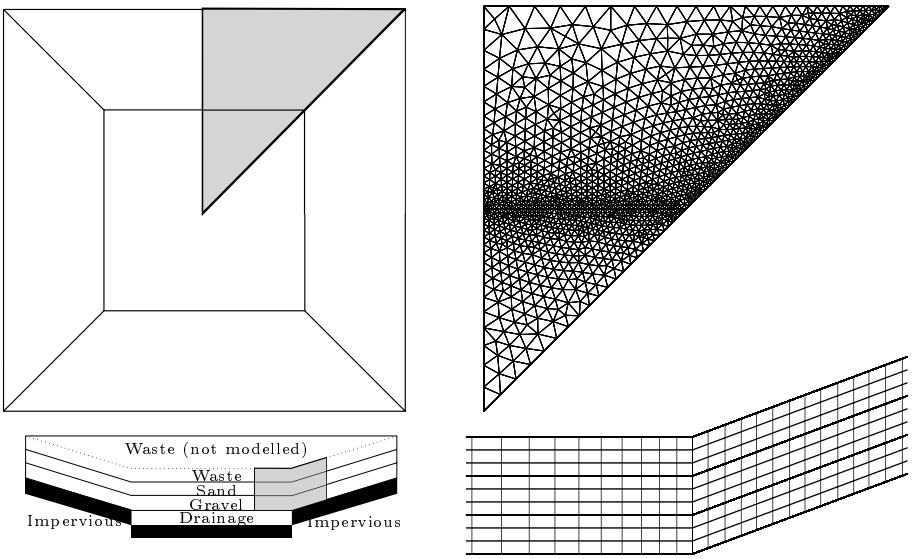


Figure 3.8: Top (top) and side views (bottom) of the landfill design (left) and its discretization (right). The dimensions of the pit are 5×100×100 m, each soil layer being 50 cm deep. The borders slope is set to 20%. The computational domain, represented in grey color in the schematic top view, is discretized into 9 layers of prisms for a total of 26,028 elements.

3.4.4 Parallel efficiency

The parallel scaling of a model is the ratio between speed-up and the number of workers N, *i.e.* the fraction of available computational resources fully-used. When assessing the parallel efficiency, it is usual to distinguish between weak and strong scalings. The former emphasizes the ability of a model to handle a greater computational domain with additional resources while the latter shows the scalability limit of a model for a fixed problem size. The weak scaling is the most frequently used for large-scale simulations. The ParSWMS model, on 64 workers with 7690 DOF per worker, achieves a weak scaling of 75% in the best cases but could decrease to 29% in the worst case shown in [68]. The ParFlow model has been tested for coupled surface/subsurface flows. With 100 workers, for 2000 DOF per worker, ParFlow reaches 40% of weak scaling and for 50,000 DOF per worker it reaches 72% [62]. A terrain-following grid formulation has been developed by [54] to avoid staircase boundaries. They presented a test-case using 500,000 DOF per worker with 16,384 wokers and achieved a weak scaling of 78% with a non-symmetric preconditioner and 51% with a symmetric preconditioner. The cost per iteration is greater for the non-symmetric than

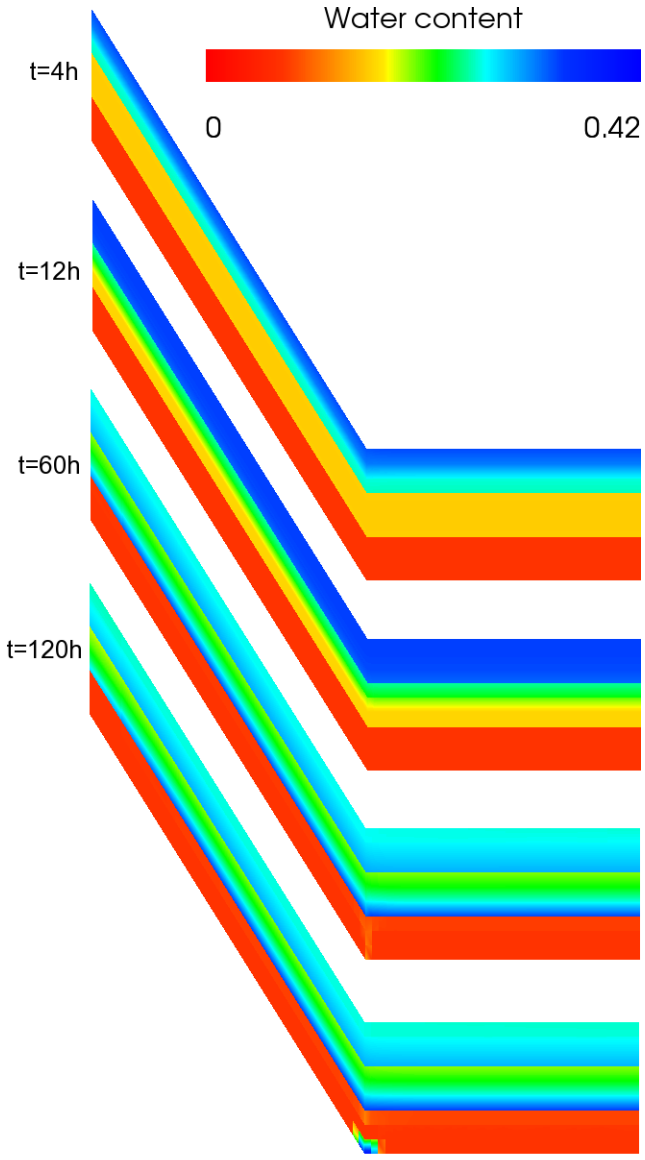


Figure 3.9: Clipped view of the water content in a vertical plane at the middle of the landfill domain. The infiltration in the waste (t=4h) and in the sand (t=12h) is homogeneous as expected. After one day, the water concentrates in the bottom of the sand layer, leading after 60h to an infiltration visible at the bottom of the slope. These graphs are scaled in the vertical direction.

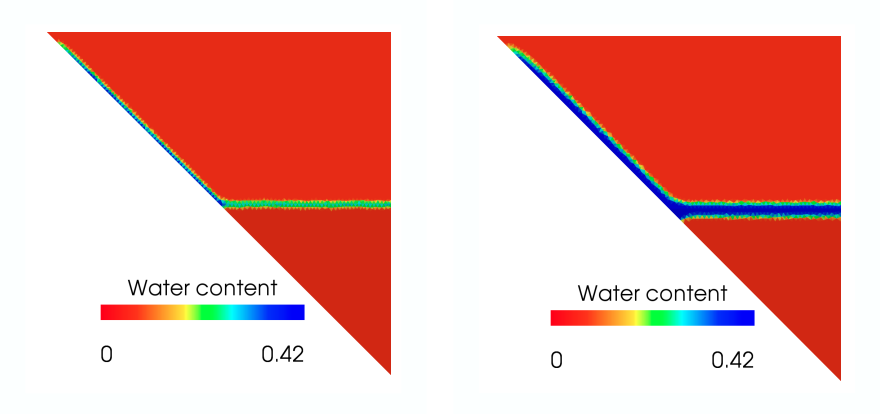


Figure 3.10: Bottom view of the water content in the computational domain. After 3 days (left figure), the infiltration reaches the bottom of the gravel. Then it quickly flows towards the bottom of the landfill, where it is drained. After 5 days (right figure) the water reaching the bottom remains confined.

for the symmetric preconditioner. In the framework of DUNE, DuMu$^{\mathrm{X}}$ has achieved a strong scaling of 36% with 64 workers with about 7341 DOF per worker. They propose the use of an algebraic multigrid solver to overcome the bad scaling of the linear solver [57].

The first reason for the suboptimal scaling observed with all these models is the reduction of the computational domain (CD) related to each worker, compared to the extent of the interfaces (I) between different CD which require communication between workers. A second reason is the elliptic behavior of the equation in the SZ, which implies that any change impacts the entire SZ. The information has to pass through several CD's to cover the whole SZ. It therefore requires a lot of communication (often via additional linear solver iterations). The first case could be magnified by the so-called "strong" scaling test cases, which keep the same mesh when increasing N. Then the ratio I/CD increases and the scaling is expected to decrease. The second case is magnified by "weak" scaling test cases, which keep the sizes of the CD and the I constant when increasing N and thus the domain size as well. This results in an increase in the number of CD, which in turn increases the number of linear solver iterations and reduces the scalability. But strong scaling test cases are also affected by this situation. That is why we choose to show a strong scaling test-case.

When studying strong scaling, it is important to look at the number of DOF per worker, as it is closely related to the ratio I/CD. Any model will see its performances degraded below a certain I/CD ratio for which the communication time begins to dominate the overall computation time. It is then more difficult to achieve a good scaling with a small number of DOF per worker. Strong scaling test cases could benefit from additional fast-memory caches when N increases, as the memory load per worker decreases accordingly. This can lead to super-linear scaling.

It is conceptually difficult to develop a model of an elliptic-type equation with a perfect scaling. Mathematically, any small local change in the model solution will have a global influence. This leads to a strong exchange of information between the sub-domains linked to each worker, and thus a poor scaling. The multigrid method is specially designed to overcome this issue. The false transient method that is used here simply reduces this exchange and thus achieves a perfect scaling by transferring information at a finite speed. The consequence that the information in an explicit method is transferred only to neighboring elements is that the number of iterations m could increase and hence reduce the efficiency as compared to implicit methods. The model efficiency would therefore be reduced in cases where the SZ occupies a large portion of the domain or is discretized with a large number of DOF's. The minimum value of m in this case is of the order of $\exp(N_c)$ where N_c is the number of saturated elements in a line.

The scaling of the model has been evaluated on a cluster of 64 cores by running the model on a test-case similar to the last one, except for the 3D mesh which is now composed of 101,508 prisms, or 609,048 DOFs. Speed-up results are shown in Fig. 3.11 and exhibit an optimal scaling.

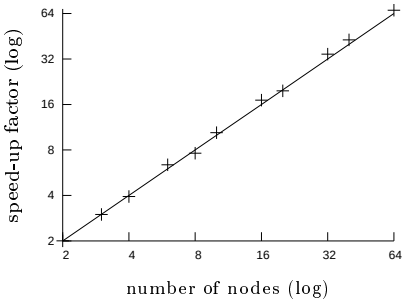


Figure 3.11: Scaling of the model. The line represents the theorically-expected speed-up and the '+' signs are the obtained results. It can be seen that the model scaling is optimal.

3.5 Conclusion

We have developed a 3D discontinuous-Galerkin model of the Richards equation using an explicit solver. As the mass variable θ is updated with the mixed-form of the equation, the mass is conserved up to machine-error precision. The DGFE method is well-suited to the advection-dominated physics that occurs at sharp gradient fronts. The DGFE method also allows the use of stabilizing techniques such as slope limiters. These limiters remove the oscillations that appear with the classical FE methods. A discontinuous approximation can also nicely capture the physical discontinuities of the water content. In our model, the UZ is modeled with the θ-form of RE and the SZ by an approximation of the ψ-form of this equation. Such an approximation, which results in a finite pressure propagation speed, allows us to use the same explicit time integration scheme for both the saturated and unsaturated zones.

Explicit methods are constrained by a limit on the time step value, depending on the size of the elements and on the water conductivity. However, the increase in the number of time steps is balanced by a smaller computational cost per time step and an optimal scaling on parallel architectures. The latter being often hard to achieve with an implicit scheme. Explicit methods also allow a direct use of conditional statements in the resolution of the equation and in the boundary conditions, which would hamper the convergence with an implicit scheme. Eventually, the present model avoids linear or non-linear solvers issues, and in this meaning is more robust. Several codes use an explicit time integration scheme in the UZ, either because they are dedicated to this zone, or to improve the execution speed [58]. In [89], a term of the equation is set explicit to obtain a strictly convex minimization problem. To our knowledge, explicit solvers have never been used for the RE in SZ.

Accuracy and efficiency could be further improved by using a better method to solve the purely elliptic part of the equation, such as the multigrid method. Indeed, if the SZ occupies a large part of the domain, the present model could require an important number of iterations to converge. That would dramatically reduce the overall efficiency of the method. As the elliptic part is linear, a linear solver alone is sufficient, without the burden of a non-linear solver. Speed improvements could be achieved by using an interpolation for the constitutive relationships, as the power function is very time-consuming. This could be also accomplished by using a multi-rate method, which consists in reducing the time steps only on the part of the domain where the physical processes have the smallest time scale [90]. The latter seems very promising for further developments.

CHAPTER 4

A scalable coupled surface-subsurface flow model

This Chapter is based on the following paper: De Maet, T., Cornaton, F., Hanert, E., 2015. A scalable coupled surface-subsurface flow model. Computers and Fluids, 116, 74-87 [7]. Some parts have been modified, principally to homogenize variables names with the other chapters. It should be noted that our original objective was to present a fully discontinuous Galerkin (DG) method for the surface model, but we faced serious issues with negative water depths as many before us. The CVFE method was then chosen to overcome those issues. The main modifications are additional details on Eq. (4.38), the correction of a parameter name for the FOEC method and possible reasons for its difference in the different models for the Borden test case, and the addition of an appendix for the CVFE method.

Abstract

The coupling of physically-based models for surface and subsurface water flows is a recent concern. The study of their interactions is important both for water resource management and environmental studies. However, despite constant innovation, physically-based simulations of water flows are still time consuming. That is especially problematic for large and/or long-term studies, or to test a large range of parametrizations with an adjoint model. As the current trend in computing sciences is to increase the computational power with additional computational units, new model developments are expected to scale efficiently on parallel infrastructures. This chapter describes a coupled surface-subsurface flow model that combines implicit and explicit time discretizations for the surface and subsurface dynamics, respectively. Despite that the surface flow has a faster dynamics than the subsurface flow, we are able to use a unique nearly-optimal time step for each submodel, hence improving the resources use. The surface model is discretized with an implicit control volume finite element method while the subsurface model is solved by means of an explicit discontinuous Galerkin finite element method. The surface and subsurface models are coupled by weakly imposing the continuity of water pressure. By imposing a threshold on the influence coefficients of the control volume finite element method, we can prevent the occurrence of unphysical fluxes in anisotropic elements. The proposed coupling is shown to produce results similar to state-of-the-art models for four different test cases while achieving better strong and weak scalings on up to 192 processors.

4.1 Introduction

The anthropogenic impact on the environment intensifies continuously with the expansion of the population and the development of its standard of living. To study how human activities influence its surrounding environment, it is important to fully understand the biogeochemical exchanges across the biosphere. Such exchanges are mainly driven by surface and subsurface water flows, which are difficult to predict without appropriate tools. Numerical models are increasingly used for this purpose.

Physically-based models are competing with statistically-based models. The formers are based on complex mathematical equations that can be difficult to parametrize [91] but with the added value of an understanding of the underlying processes. The laters, based on simple generic formulas, can provide an easy and precise fit with observations data but lack of flexibility when a change occurs in the system [92]. It is possible to mix both approaches with an uncertainty analysis to assess the variability of the results [?]. The main sources of uncertainty of physically-based models are the physical model hypotheses, the

mathematical approximations, the numerical discretization, the heterogeneity and variability of the parameters, and the calibration of non-linear models with uncertain measures.

In a physically-based model of the terrestrial water cycle, the processes are usually modeled by means of the shallow water equations for the surface flows and the Richards' equation for the subsurface flows. The shallow water equations are a convenient 2D approximation of the full 3D Navier-Stokes equations when the water height is small, which is the case for surface flows. It can be complemented by additional 1D equations for rivers and channels to handle the jump in the physical process scales. The Richards' equation approximates the soil as a porous medium with highly non-linear parameters. It assumes an isothermal and laminar flow with no chemical gradients or inertial forces and water as the unique fluid phase, hence neglecting the air component [91]. It can be complexified by adding hysteresis, fractures, multiple phases or macropores, although those extensions are difficult to spatialize and parametrize. Since the physical complexity of the shallow water equations goes beyond what is required for surface-subsurface flow interactions, simpler models are generally used. The most popular ones are the non-inertia or diffusive wave model and the kinetic wave model. Some simplified approaches based on the kinetic wave equation are going further in the approximations, simulating the surface water via a tree-structured network of water reservoirs, following the topographical slopes [93]. While being fast, this method is based on strong underlying hypothesis and is hence inappropriate for natural reservoirs. Among the many existing approaches [61, 94, 19], a state-of-the art method to discretize the non-inertia equation is the control volume finite element (CVFE) method, also called the influence coefficient method [95, 58, 96, 64, 18]. This method applies upwind fluxes between the nodes of a mesh element. Its main advantage over the classical continuous Galerkin formulation is to avoid the issues related to zero or negative water depths. As the non-inertia equation is nearly elliptic when the water height becomes significant, an implicit time integration scheme is recommended.

The numerical discretization of the Richards' equation has been extensively studied, as it presents various numerical difficulties such as unphysical oscillations, mass conservation errors or a lack of robustness. These issues can be partly circumvented by carefully selecting the non-linear solver [27, 35] as well as the space discretization [58, 97, 22]. Most Richards' equation models rely on implicit time integration schemes and hence present convergence issues [98] or sub-optimal scaling on parallel infrastructures [62, 54, 68, 81, 57, 63]. The time step of implicit time integration schemes is unrestricted for simple diffusion equations, but the non-linearities of the Richards' equation put an upper limit to it [98]. Recently, De Maet et al. [6] have proposed a model using an explicit time integration scheme and a discontinuous Galerkin (DG) finite element (FE)

spatial discretization. Such an approach achieves an optimal strong scaling as it does not require linear or non-linear solvers and hence avoids the associated convergence issues. It relies on the use of slope limiters to increase the scheme robustness and a special DG interface term that allows physical discontinuities in the water content at the elements interface. The interface between two different soils is precisely represented by the DG FE approximation, therefore no mixing between the different properties is necessary as it is the case in continuous Galerkin FE models. A detailed review of Richards' equation models can be found in [22].

In the last decade, the coupling of the shallow water equations and the Richards' equation has been an increasingly active domain of research (see for instance [99] or [71] for an overview). The complexity of studies in this field are mostly due to the fact that surface and subsurface interactions are difficult to measure. Another issue is the difficulty to model water fluxes that often exhibit a large spatial and temporal variability. Indeed, processes occurring at small spatial scales, like river flows, coexist with processes occurring at large spatial scales, like groundwater flows. Similarly, slow processes like the dynamics of the vadose zone coexist with rapid processes like surface runoff.

In a continuous world, when surface water is present, the most physically consistent coupling is to match the hydrostatic pressure of the surface flow with the pressure head of the subsurface flow at the top of the soil layer [62, 18]. However, a pressure continuity (PC) coupling strategy would require the soil to be discretized up to the scale of the smallest water fluxes between surface and subsurface, which is rarely feasible in practice because of the associated computational cost. Additionally, the small features of the surface linked to those specific fluxes, such as the microtopography, the surface soil compression and vegetation cover, are often very difficult to estimate. Eventually, such a coupling strategy requires the surface and subsurface models to be connected in one non-linear solver step. The solution is then provided either by iterative coupling methods, which require multiple iterations per time step, or by an implicit time integration scheme, which produces a non-linear system that is often difficult to solve and scales poorly on parallel architectures. Another coupling is the first-order exchange coefficient (FOEC) coupling for which the pressure continuity is weakly imposed [58, 96, 18]. The FOEC coupling allows the surface and subsurface to be solved separately and it can assume additional sub-scale physics at interfaces. It converges towards the PC coupling when the coefficient tends towards infinity. With an appropriate choice of coefficient it can produce results very close to the PC coupling with enhanced model performances [100, 101].

Although the research on coupled surface-subsurface models is well developed, few models achieve a good scaling on parallel architectures. For Richards' equation, the parallel efficiency (defined as the fraction of available computational

resources fully-used) of a model like PARSWMS is of 75% but it can decrease to 29% in some cases [68]. For the coupled model PARFLOW, the efficiency varies between 40% and 72% [62]. As a general rule of the thumb, performances decrease with the number of computational units and increase with the number of degrees of freedom allocated to each computational unit. This is mainly due to the complexity of the global system solution, which requires many communications to exchange information between subdomains. That amount of communications limits the parallel efficiency, especially when a large number of nodes is involved. Those performances are likely to keep decreasing in the future with the use of newer technologies. Indeed, today new computers increase their power by adding more computational units. That implies a change of paradigm for computational code development as individual computing units are no longer increasing in power. Instead, they are duplicated. To use all capabilities of future devices, adapted algorithms have therefore to be developed to achieve efficient parallel codes.

In this chapter, we present a coupled surface-subsurface flow model that combines an implicit model for the non-inertia shallow water equation and an explicit model for the Richards' equation [6]. Such an approach allows us to use the same time step for both models, as the slow dynamics of the groundwater requires an explicit time step close to the implicit time step required for convergence of the surface flow non-linear solver. Despite using an implicit scheme for the surface model, the overall scaling is still nearly optimal as the subsurface model generally needs the largest part of the computational resources. The FE method has been selected mostly for its ability to solve the model equations on unstructured meshes, which are well suited to complex geometries such as real catchments. Its CVFE declination for surface flow is close to a finite volume method, increasing first the robustness and then the scheme convergence. Its DG FE declination for subsurface flow allows for physical discontinuities of the water content and for the use of limiters to also increase the scheme robustness. As both the non-inertia and the Richards equations are strongly non-linear, robustness is often favored over precision, which would be achieved for instance by a higher order spatial discretization. The use of similar spatial discretizations for the surface and the subsurface models allows an easier coupling, as each surface element has a unique corresponding subsurface element face. We introduce a flexible coupling approach that lies between an exact surface-subsurface pressure coupling, and the FOEC formulation. This hybrid coupling comes together with the DG FE method when using its Dirichlet boundary condition. It has the advantages to be easier to solve than a direct coupling, as it is less stringent, to converge towards the pressure continuity coupling after a transitory phase, and to be usable with an explicit time integration scheme. This is not the case of the exact coupling that requires a conjoint implicit solution of both the surface and the subsurface models. A number of numerical examples

are provided to highlight the model properties and show how the scaling is affected by the surface and subsurface models.

4.2 Mathematical formulation

In this section, we give an overview of the subsurface and surface models and describe different approaches to couple these two models.

4.2.1 Subsurface model

The 3D subsurface model relies on the three-dimensional Richards' equation, which can be expressed as follows:

$$\frac{\partial \theta}{\partial t} = \nabla \cdot (\mathbf{K} \cdot \nabla(\psi - z)) + s, \tag{4.1}$$

$$\theta = f_\theta(\psi), \tag{4.2}$$

where θ [$\mathrm{L^3L^{-3}}$] is the volumetric soil water content, ψ [L] is the pressure head, z [L] is the upward positive vertical coordinate, s [$\mathrm{T^{-1}}$] a sink-source term, $\mathbf{K}$ [$\mathrm{LT^{-1}}$] the water conductivity tensor and f_θ [–] the retention curve. Eqs. (4.1) and (4.2) are complemented with appropriate initial and boundary conditions:

$$\psi = \psi_0, \qquad \text{on } \Omega, \quad t = 0 \tag{4.3}$$

$$\psi = \psi_D, \qquad \text{on } \Gamma_D, \ t \in [0, t_{end}] \tag{4.4}$$

$$-(\mathbf{K} \cdot \nabla(\psi - z)) \cdot \mathbf{n} = J_N, \qquad \text{on } \Gamma_N, \ t \in [0, t_{end}] \tag{4.5}$$

with $\mathbf{n}$ [–] the outward normal vector, t_{end} [T] the simulation duration, Ω the computational domain, Γ_D the Dirichlet part of the boundary (where the value ψ_D is imposed) and Γ_N the Neumann part (where the flux J_N is imposed). The constitutive relations defining $\theta = f_\theta(\psi)$ and $\mathbf{K}$ are derived from van Genuchten [102] and Mualem [103]:

$$S_e = \frac{\theta - \theta_r}{\theta_s - \theta_r}, \tag{4.6}$$

$$S_e = \begin{cases} (1 + |\alpha\psi|^\beta)^{-\nu} & \text{if } \psi \leq 0, \\ 1 & \text{if } \psi > 0, \end{cases} \tag{4.7}$$

$$\mathbf{K} = \mathbf{K}_s \ S_e^{l_p} \left(1 - (1 - S_e^{1/\nu})^\nu\right)^2, \tag{4.8}$$

where S_e [–] is the effective saturation, θ_r [$\mathrm{L^3L^{-3}}$] is the residual volumetric water content, θ_s [$\mathrm{L^3L^{-3}}$] is the saturated volumetric water content, $\mathbf{K}_s$ [$\mathrm{LT^{-1}}$] is the anisotropic saturated water conductivity tensor, α [$\mathrm{L^{-1}}$] is a parameter related to the air-entry pressure value, β [–] is a parameter related to the pore-size distribution, and $\nu = 1 - 1/\beta$ [–].

4.2.2 Surface model

The surface water is modeled with the non-inertia approximation of the shallow water equations. These equations rely on the assumption that the flow aspect ratio is very small. The non-inertia approximation further assumes that inertial terms can be neglected. Gottardi and Venutelli [104] have shown that this approximation is acceptable to simulate runoff flows as it yields results very close to the analytical and numerical solution of the full shallow water equations. The non-inertia approximation reads:

$$\frac{\partial h}{\partial t} = \nabla \cdot \left(h^{3/2} \sqrt{\frac{g}{G}} \frac{\nabla(h+b)}{\sqrt{|\nabla(h+b)|}} \right) + q_s \tag{4.9}$$

for which the z-axis is defined positive upward, where h [L] is the thickness of the water layer, b [L] is the surface elevation, g [LT^{-2}] is the gravitational acceleration, G [–] a friction coefficient and q_s [LT^{-1}] a source/sink term. G can be expressed as follows:

$$G = \frac{g n_x^2}{h^{1/3}} \text{ or } \frac{g}{C_x^2} \text{ or } \frac{f_x}{8}. \tag{4.10}$$

where n_x [$\mathrm{TL}^{-1/3}$] is the Manning coefficient, C_x [$\mathrm{L}^{1/2}\mathrm{T}^{-1}$] is the Chezy coefficient and f_x [–] is the Darcy coefficient.

4.2.3 Coupling between the surface and subsurface models

The shallow water equations and the Richards' equation are coupled through a boundary condition of the subsurface model, by equating the surface hydrostatic pressure (assumed to be equal to h) and the subsurface pressure head ψ when $h > 0$. To ensure mass-conservation, the interface flux F_{BC} [T^{-1}] is applied to the surface model as a source term through q_s in Eq. (4.9). The expression of F_{BC} depends on the coupling method. In this study, we will consider two common coupling strategies and then present a new one that combines the two previous ones. Each coupling has its own advantages and drawbacks both in terms of numerical discretization and of physical accuracy.

When no ponding occurs, the interface flux is driven by the evapotranspiration. However, over the time scale of a rainfall event the evapotranspiration flux can be neglected as compared to the magnitude of the rain flux. All the test cases presented in this chapter are driven by rainfall events and hence neglect evapotranspiration. Of course, evapotranspiration will have to be taken into account to achieve realistic, long-term simulations.

When ponding occurs, two main approaches are commonly used, the PC coupling and the FOEC coupling:

$$F_{BC,PC} = \mathbf{n} \cdot (\mathbf{K} \cdot \nabla(\psi - z)), \tag{4.11}$$

$$F_{BC,FOEC} = \alpha_{FOEC}(\psi - h), \tag{4.12}$$

where α_{FOEC} [T^{-1}] is an exchange coefficient. If this coefficient is large, ψ and h quickly converge. If it is small, the two models are almost decoupled. The PC coupling could be viewed as a special case of the FOEC coupling when α_{FOEC} is infinite. This coefficient is able to model surface features different from the bulk of the soil. The flux of Eq. (4.12) is then viewed as the simplest expression of a Darcy flux. Following the approach of VanderKwaak [58] α_{FOEC} is defined as:

$$\alpha_{FOEC} = K_z \frac{\zeta}{a_s} f^a \chi, \tag{4.13}$$

where K_z [LT^{-1}] is the soil conductivity in the z direction, ζ [L^{-1}] is the surface exchange interface area to volume ratio, a_s [L] is the surface coupling length scale, f^a [–] is the isotropic porous media area fraction (used only when macropores are considered) and χ [–] is an exchange scaling coefficient. As these parameters are often difficult to estimate and measure in practice, it is usual to use the empirical formula $\alpha_{FOEC*} = K_z/l_c$ instead of Eq. (4.13), where l_c [L] is a characteristic length which acts as a fitting parameter [100].

A third approach is to combine the pressure-continuity and FOEC fluxes to obtain a hybrid coupling. The coupling flux is then mathematically expressed as

$$F_{BC,DG} = \sigma_B(\psi - h) + \mathbf{n} \cdot (\mathbf{K} \cdot \nabla(\psi - z)), \tag{4.14}$$

where σ_B [T^{-1}] is a penalty parameter quite similar to α_{FOEC*}. On the one hand, the above definition of α_{FOEC*} involves only parameters that have a physical meaning, although they are often difficult to measure. On the other hand, σ_B is set according to the penalty parameters of the DG space discretization (see Eq. (4.31)). σ_B is defined by assuming that the soil-surface is like any other inter-element interface within the soil computational domain and is proportional to K_z/l_e, where l_e is a characteristic length of the two adjacent elements. The difference with the α_{FOEC*} formula is that l_c is generally user-defined while l_e is defined automatically from the model. The interface flux described by Eq. (4.14) can be viewed as a compromise between the continuous coupling and the FOEC coupling, close to a Robin boundary condition. It is a weak coupling, enhancing the solver convergence, and it converges towards the physical continuity coupling when ψ has reached h. It is consistent with the soil domain discretization described thereafter. Indeed, no specific code was necessary to introduce Eq. (4.14) as it is already part of the DG FE method when a Dirichet boundary condition is specified.

As a side note we shall point out that the main advantage of the PC coupling is its physical meaning, as water pressure does not exhibit discontinuities. However, as mentioned before, it brings additional numerical issues. Both the FOEC and the hybrid couplings avoid those issues by relaxing the equality constraint with a weak coupling. The coefficient σ_B of the hybrid coupling is coherent with the DG discretization that will be introduced in the next section and is automatically fixed. The constraint at the coupling interface has the same strength than the constraint of the subsurface model at elements interfaces. The result is therefore close to the one obtained with the PC coupling. The FOEC coupling can reach the same goal or can describe near-surface physics, depending on the user-defined parameter l_c. In the present model we do not use the PC coupling. The hybrid coupling is always used, except when additional near-surface physical processes are required (like in section 4.4.2). In that case, the FOEC coupling is used and the parameter l_c is calibrated to achieve the best fit with observations.

4.3 Space and time discretizations

The model equations are now discretized in space and time, before being coupled. We first present the discretization of Richards' equation with a DG FE scheme in space and an explicit scheme in time. We then consider the discretization of the non-inertia approximation of the shallow water equations with a CVFE scheme in space and an implicit scheme in time. Finally, we summarize the entire coupling algorithm.

4.3.1 Discretization of the subsurface model

Before formally discretizing the subsurface model, the ideas of the resolution procedure are presented for additional clarity. Eq. (4.1) is usually expressed in two forms, namely the ψ-form and the θ-form:

$$C\frac{\partial \psi}{\partial t} = \nabla \cdot (\mathbf{K} \cdot \nabla(\psi - z)) + s, \tag{4.15}$$

$$\frac{\partial \theta}{\partial t} = \nabla \cdot \left(\frac{\mathbf{K}}{C} \cdot \nabla\theta - \mathbf{K} \cdot \nabla z\right) + s, \tag{4.16}$$

where $C(\psi) = df_\theta(\psi)/d\psi$ [L^{-1}]. Eq. (4.15) is valid everywhere but generally leads to mass conservation issues. Eq. (4.16) is best suited to dry soils and is not valid in saturated soils. We have adopted the following approach for approximating the subsurface model: (1) solving a modified version of Eq. (4.15) by initially neglecting the mass-conservation issue, (2) solving Eq. (4.1) by using the value of ψ from the first step as a predictor, (3) updating the ψ field with $f_\theta^{-1}(\theta)$ [–] in unsaturated areas and (4), if necessary, using a smoother on θ in saturated areas.

The approximation to the step (1) consists in modifying the function C to allow the use of an explicit time integration scheme. Indeed, C (almost) reached 0 in saturated areas and an explicit time integration scheme would require extremely small time steps to remain stable. We therefore make the following approximation:

$$C \simeq \tilde{C} = \max\left(C, K/\tau\right), \tag{4.17}$$

where K $[\mathrm{LT}^{-1}]$ is the largest eigenvalue of conductivity tensor $\mathbf{K}$ and τ $[\mathrm{L}^2\mathrm{T}^{-1}]$ is a free parameter. In transient situations, we iterate over step (1) to converge towards the exact solution. The iterations on θ in steps (2) and (4) are perfectly mass-conservative. The approximation occurs in the saturated zone but also in a small part of the unsaturated zone where $\tilde{C} > C$. When an abrupt transition occurs in saturated areas, the algorithm can produce spurious mass fluxes. To correct these in an explicit and mass-conservative way, the following smoother equation is used:

$$\frac{\partial \theta}{\partial t} = \lambda \nabla^2 (\theta - f_\theta(\tilde{\psi})), \tag{4.18}$$

where λ $[\mathrm{L}^2\mathrm{T}^{-1}]$ is a free parameter and $\tilde{\psi}$ is the value of ψ at the current time step (constant within this equation). The effect of this equation is simply to filter out unwanted variations of θ. Several iterations of this equation can be applied to increase its effect.

Now Eqs. (4.1), (4.15) and (4.18) are discretized in space with the DG FE method. This method is well-suited to represent advection-dominated flows like infiltration fronts. The DG FE method also allows physical discontinuities of θ between soils of different properties, or the use of slope limiters to prevent spurious oscillations. By partitioning the domain Ω into N non-overlapping elements Ω_e with interfaces Γ_e, the spatially and temporally continuous model variables θ and ψ can be approximated by the discrete variables θ^h and ψ^h as

$$\theta(x,y,z,t) \simeq \theta^h(x,y,z,t) = \sum_{j=1}^{N_d} \theta_j(t)\phi_j(x,y,z), \tag{4.19}$$

$$\psi(x,y,z,t) \simeq \psi^h(x,y,z,t) = \sum_{j=1}^{N_d} \psi_j(t)\phi_j(x,y,z), \tag{4.20}$$

where N_d $[\dashv]$ is the total number of degrees of freedom (DOF's) and ϕ_j $[\dashv]$ are piecewise first-order Lagrange polynomials defined on each element Ω_e. The jump $[\cdot]$ and averaging $\{\cdot\}$ operators on the interface Γ_e are defined as:

$$[x] \triangleq x^+ - x^-, \qquad \{x\} \triangleq \frac{x^+ + x^-}{2}, \tag{4.21}$$

where the '+' and '–' superscripts indicate the trace value taken either on one or opposite side of Γ_e. At the boundaries, both operators are defined in terms of an external value derived from the Dirichlet boundary condition. The resulting weak boundary conditions are known to be more stable than strong ones [82, 83] and will be used further for the coupling as well. The discrete equations are obtained by deriving the Galerkin formulation of Eqs. (4.15), (4.1) and (4.18) with the basis functions ϕ_i $(1 \leq i \leq N_d)$. The interior penalty DG method is then applied. By using a matrix notation and capitalizing vectors variables, we obtain

$$\tilde{C}M\frac{d\Psi}{dt} = K\Psi - Z + P, \tag{4.22}$$

$$M\frac{d\Theta}{dt} = K\Psi - Z + P, \tag{4.23}$$

$$M\frac{d\Theta}{dt} = \Lambda(\Theta - f_\theta(\tilde{\Psi})) + P_\theta, \tag{4.24}$$

where the matrices M, K, Λ are obtained by assembling the following element-wise matrices, with n_e the number of nodes per element and $1 \leq i, j \leq n_e$ over each element e and its neighbors:

$$M_{ij}^e = \langle \phi_i \phi_j \rangle, \tag{4.25}$$

$$K_{ij}^e = \langle \mathbf{K} \cdot \nabla\phi_i \cdot \nabla\phi_j \rangle - \langle\!\langle \{(\mathbf{K} \cdot \nabla\phi_j) \cdot \mathbf{n}\} [\phi_i] \rangle\!\rangle, \tag{4.26}$$

$$\Lambda_{ij}^e = \langle \lambda \nabla\phi_i \cdot \nabla\phi_j \rangle - \langle\!\langle \{\lambda \nabla\phi_j \cdot \mathbf{n}\} [\phi_i] \rangle\!\rangle, \tag{4.27}$$

where $\langle \cdot \rangle$ is defined as $\sum_{e=1}^{N} \int_{\Omega_e} \cdot \, d\Omega$ and $\langle\!\langle \cdot \rangle\!\rangle$ as $\sum_{e=1}^{N} \int_{\partial\Omega_e} \cdot \, d\Gamma$.. The vectors Z, P, P_θ are assembled in the same way, with $1 \leq i \leq n_e$:

$$Z_i^e = \langle \mathbf{K} \cdot \mathbf{z} \cdot \nabla\phi_i + s\phi_i \rangle - \langle\!\langle \{(\mathbf{K} \cdot \mathbf{z}) \cdot \mathbf{n}\} [\phi_i] \rangle\!\rangle, \tag{4.28}$$

$$P_i^e = \sigma \left\langle\!\!\left\langle [\phi_i] \left[\psi^h\right] \right\rangle\!\!\right\rangle, \tag{4.29}$$

$$P_{\theta i}^e = \sigma_\theta \left\langle\!\!\left\langle [\phi_i] \left[\theta^h - \theta_{ref}\right] \right\rangle\!\!\right\rangle, \tag{4.30}$$

where the penalty parameters σ $[\mathrm{T}^{-1}]$ and σ_θ $[\mathrm{LT}^{-1}]$ are defined as:

$$\sigma = \frac{4}{l_e} \, (n_0 + 1)(n_0 + 2) \, K_{sc}, \tag{4.31}$$

$$K_{sc} = (\mathbf{K} \cdot \mathbf{n}) \cdot \mathbf{J} / \|\mathbf{J}\|, \tag{4.32}$$

$$\mathbf{J} = \mathbf{K} \cdot \nabla(\psi - z), \tag{4.33}$$

$$\sigma_\theta = \frac{4}{l_e} \, (n_0 + 1)(n_0 + 2) \, \lambda, \tag{4.34}$$

with n_0 [–] the order of the FE approximation (in our case $n_0 = 1$), K_{sc} $[\mathrm{LT}^{-1}]$ the normal flux-oriented scalar conductivity, $\mathbf{J}$ $[\mathrm{LT}^{-1}]$ the water flux, and l_e

[L] a characteristic length of the two adjacent elements. On the boundaries, σ reduces to σ_B introduced in Eq. (4.14), variable which is identical excepted for the l_e parameter which relies on the unique adjacent element.

To increase the stability and efficiency of the model, the mass matrices in Eqs. (4.22) and (4.23) have been lumped. Following other model designs, the diffusivity tensor $\mathbf{K}$ has been linearized over each element in terms of its nodal values [20, 18]. Eqs. (4.22), (4.23) and (4.24) are discretized in time with an explicit Euler scheme. When correctly implemented, explicit schemes can yield a perfect scaling, as shown in [6]. Indeed, their inherent simplicity and the absence of linear system solver limit the communication overhead between processors. The detailed equations are given in section 4.3.3.

4.3.2 Discretization of the surface model

The surface model has been discretized in space with the CVFE method [95]. This method appears to be more stable and efficient than other methods like the continuous or the discontinuous Galerkin discretizations. It is mainly due to the fact that the CVFE method naturally avoids zero or negative depth issues. Indeed, the flux between two nodes of an element is set to zero if no water is present in the upwind node. Linear and non-linear solvers achieve a better convergence with the CVFE method because the equation coefficients are then constrained to physical values. This allows the use of larger time steps and thus improves of the overall computational speed.

The CVFE method is based on a continuous Galerkin formulation. The expansion of the model unknowns is similar to Eq. (4.19) with the exception that the basis functions $\phi_i (1 \leq i \leq n)$ are now continuous. By applying the classical continuous Galerkin method to Eq. (4.9), we obtain the following discrete equations:

$$\mathbb{M}\frac{dH}{dt} = NH + B, \tag{4.35}$$

where $\mathbb{M}^e_{ij} = \langle \phi_i \phi_j \rangle$, $N^e_{ij} = \langle n \nabla \phi_i \cdot \nabla \phi_j \rangle$, $B^e_i = \langle n \nabla b \cdot \nabla \phi_i + q_s \phi_i \rangle$ and $n = h^{3/2}\sqrt{g/G}/\sqrt{|\nabla(h+b)|}$.

To obtain a CVFE discretization, the mass matrix $\mathbb{M}$ is lumped and an inter element upwinding is applied. This amounts to modify the diffusive terms in $NH + B$ as follows (for more details, see Appendix A.3):

$$\langle n \nabla(u) \cdot \nabla \phi_i \rangle \rightarrow n_{ij}(u_j - u_i) \langle \nabla \phi_j \cdot \nabla \phi_i \rangle, \tag{4.36}$$

where $u = b + h$ and

$$n_{ij} = \begin{cases} n(x_i) & \text{if } (u_j - u_i) \langle \nabla \phi_j \cdot \nabla \phi_i \rangle < 0, \\ n(x_j) & \text{if } (u_j - u_i) \langle \nabla \phi_j \cdot \nabla \phi_i \rangle > 0, \end{cases} \tag{4.37}$$

When a mesh element is anisotropic, this method can result in negative influence coefficients, *i.e.* $\langle \nabla\phi \cdot \nabla\phi_i \rangle < 0$ [58]. The resulting flux between two nodes is then non-physical, as it follows the pressure gradient instead of being opposite to it. As n is a non-linear function of h, such a flux between two nodes can dominate the other internode fluxes inside the element. The resulting flow inside the element is then unphysical. To avoid those issues the simple solution used here (called approximate CVFE method thereafter) consists in canceling all the negative coefficients:

$$n_{ij}^{*} \simeq \max(n_{ij}, 0) \tag{4.38}$$

This approximate CVFE method has no impact on the model solution as compared to the original CVFE, except for anisotropic elements for which the solution is improved (see section 4.4.3). Since triangular elements are less likely to yield negative influence coefficients than quadrangles, the approximate CVFE method is mainly used for the latter. Despite the good results obtained with this method, emphasis should be put on producing meshes without anisotropic elements to avoid the approximation. In our case, we were constrained by a specific regular mesh refinement for the V-catchment test case, with 100×5 m rectangles in the channel. It should be noted that Eq. (4.38) is mass conservative as it only cancels a flux between two nodes on both sides.

Eq. (4.35) is discretized in time with an implicit Euler scheme. The non-linear solver is based on the Newton-Raphson method and the linear solver is based on the Generalized Minimal RESidual (GMRES) method [105]. The fully discretized equation is given in section 4.3.3.

4.3.3 Summary of the numerical solution procedure

The overall solution procedure for the coupled model is summarized below. For a given parametrization, an initial state (Ψ^0, Θ^0, H^0) and a time step Δt, we iterate over the following steps:

1. Solve the equation for the intermediate steps $H^{n+1,k}$, given $H^{n+1,0} = H^n$:

$$\text{for } k = 1, \ldots, m: \tag{4.39}$$

$$F_{BC}^{k} = \text{couple}(H^n, \Psi^{n+1,k-1}) \tag{4.40}$$

$$\tilde{C}M\frac{\Psi^{*} - \Psi^{n+1,k-1}}{\Delta t/m} = K\Psi^{n+1,k-1} - Z + P, \tag{4.41}$$

$$\Psi^{n+1,k} = \text{limit}(\Psi^{*}), \tag{4.42}$$

where m [–] is the number of iterations for the ψ-form, the coupling function is either Eq. (4.12) or Eq. (4.14) and the limiter is described in [6].

2. Solve the equation for $\Theta^{n+1,0}$, given Θ^n and $\Psi^{n+1,m}$:

$$F_{BC} = \text{couple}(H^n, \Psi^{n+1,m}) \tag{4.43}$$

$$M\frac{\Theta^* - \Theta^n}{\Delta t} = K\Psi^{n+1,m} - Z + P, \tag{4.44}$$

$$\Theta^{n+1,0} = \text{limit}(\Theta^*). \tag{4.45}$$

3. Compute Ψ^{n+1} by combining $\Psi^{n+1,m}$ and the retention curve relationship:

$$\Psi^{n+1} = \begin{cases} f_\theta^{-1}(\Theta^{n+1,0}) & \text{in the dry zone,} \\ \Psi^{n+1,m} & \text{elsewhere.} \end{cases} \tag{4.46}$$

4. Weakly correct $\Theta^{n+1,0}$ in the wet and the saturated zones:

$$\text{for } k = 1, \ldots, q: \tag{4.47}$$

$$M\frac{\Theta^{n+1,k} - \Theta^{n+1,k-1}}{\Delta t} = \Lambda(\Theta^{n+1,k-1} - f_\theta(\Psi^{n+1})) + P^\theta, \tag{4.48}$$

where q [–] is the number of iterations for the θ filter and where $\tilde{\Psi}$ is defined as Ψ^{n+1}.

5. Implicitly solve the equation for H^{n+1} using F_{BC}:

$$\mathbb{M}\frac{H^{n+1} - H^n}{\Delta t} = NH^{n+1} + B^{n+1} + P_h^{n+1}, \tag{4.49}$$

In the current form of the model, the parameters m and q are set manually based on indicators of convergence like an excess of water content. Indeed, a bad convergence in saturated areas creates non-elliptical fluxes that yields a unphysical water content above the saturation limit. A method to estimate those two parameters along with the time step in order to optimize the computation time is still needed to obtain a model suitable for long-term simulations.

4.4 Results

This section is divided into four subsections presenting different test-cases that highlight the model properties. First, we verify the ability of the surface model and of the coupled model to reproduce results of similar models in three test-cases. Then a well-known integrated modeling example from a field-scale experiment is given. A third section analyses the sensitivity of two aspects of the presented model: the anisotropy of surface elements and the FOEC coupling. Finally, results for both strong and weak scalings are presented.

As the whole model is mass conservative at machine precision and as the 'double' C++ floating number precision has been used, the relative mass balance error for all the simulations stays below 10^{-15}.

4.4.1 Model verification and validation

The model verification is done through the validation of the two sub-models and the validation of their coupling. The subsurface model has been validated in a previous publication [6]. The surface model implementation with the CVFE method is compared with the tilted V-catchment 2D example from [95] that has been extensively used to validate surface models (as for instance [19, 106]). As this method has already been proved accurate [95, 58, 18, 64], we only assess the reproducibility of the results with our implementation.

The hybrid coupling method is used in the two coupled test cases of this section. The coupled model is firstly validated with a test case of Kollet and Maxwell [62] to check the model's ability to handle an Hortonian runoff process. The second validation case is the sandbox example of Abdul and Gillham [107] which is based on experimental outflow results and has been extensively simulated [108, 62, 19, 106]. It presents both infiltration and exfiltration and is especially challenging for the subsurface model as the almost totally saturated media loads to for a strong elliptical behavior. Similar results with some PC coupling models can assess for a fast-enough convergence of the coupling methods towards the PC coupling.

Surface model – V-catchment

The numerical setup of [95] consists in a impermeable 1000×800 m plane, with a slope of 2% along its length and 5% along its width. A 1-m deep and 10-m wide channel is located at the bottom and carries the water towards the unique outlet of the domain. The Manning roughness coefficient is set to 0.015 $\mathrm{sm}^{-1/3}$ for the plane and 0.15 $\mathrm{sm}^{-1/3}$ for the channel. Zero-flux boundary conditions are imposed everywhere except at the outlet of the channel where a critical depth boundary condition is used. The initial state is a dry surface. A uniform and constant rain is then applied for 90 min with a rate of 3×10^{-6} m/s. The resulting outflow, computed on a regular mesh of 100-m resolution squares and a time step of 100 s, is shown in Fig. 4.1. The other model results were obtained with a 100-m resolution and time steps ranging between 5 and 600 s. It can be seen that our model with the 100-m resolution produces an hydrograph very close to the ones obtained with HGS which also uses the CVFE method, SHE [95] or MODFLOW-SURFACT. The results of di Giammarco [95] (CVFE and finite differences) converge more quickly to the steady state. This delay can be reduced by refining both the temporal and the spatial discretization as can be seen for our model result with finer resolutions of 10-m and 1 s in Fig. 4.1.

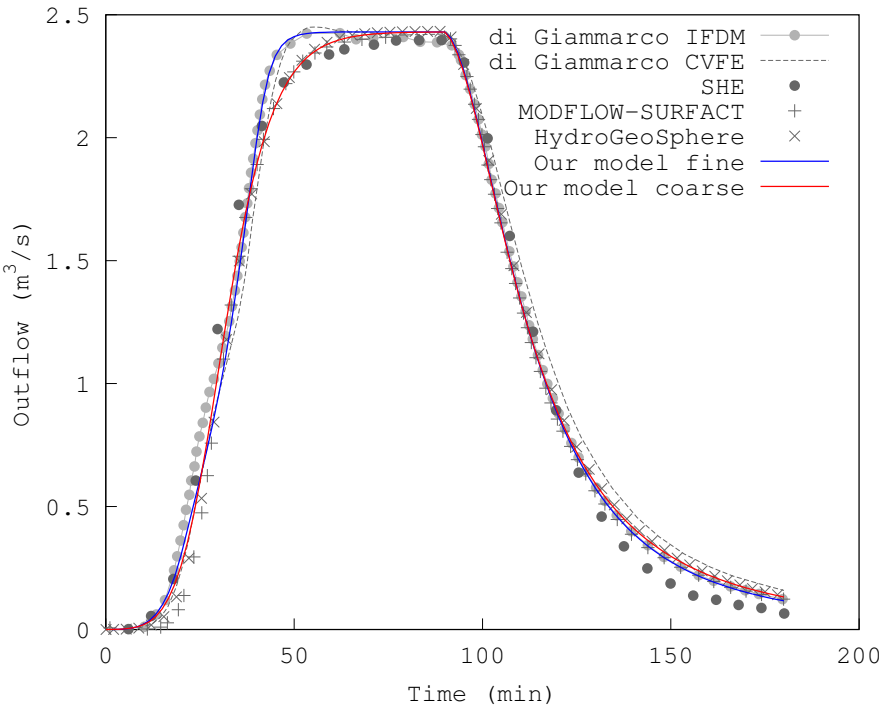


Figure 4.1: The V-catchment test case assesses the ability of a model to handle the simplest valley-shaped geometry. The outflow of our model is within the range of other model outflows presented in [58] and [18]. The fine/coarse discretizations are of 10/100m in space and 1/100s in time.

Coupled model – Hortonian runoff

Kollet and Maxwell (K&M) [62] present several modeling examples that focus on the surface-subsurface coupling. Their runoff production by excess infiltration test case (Hortonian runoff) consists in a 1D surface slope of 0.0005 spread over 400m and characterized by a Manning coefficient $n = 0.019872$ sm$^{-1/3}$. The underlying soil is uniform and is characterized by $\theta_s = 0.4$, $\theta_r = 0.08$, $\alpha = 1$, $\beta = 2$. The parameter K_s is set either to 1.1574×10^{-7} or 1.1574×10^{-6} m/s, to assess two different behaviors. The initial water depth is set to 1m below the surface. A rain of 5.5×10^{-6} m/s is applied during 200 min. We kept the same discretization as K&M, which is of 80 m along the slope and either 1 or 5 cm vertically.

An analytical solution of this problem exists but only for an impermeable surface. In order to use that solution, we therefore perform simulations without any infiltration. The resulting hydrograph is provided as a reference as it represents the situation without infiltration [104]. Our model reproduces accurately the beginning of the event and the beginning of the recession, with a discrepency before reaching the steady state (see Fig. 4.2). With the same discretization, our surface model has the same accuracy as the model of K&M. K&M showed that the analytical profile can be approached with refinement of the spatial resolution [62].

We then consider the case with the lower soil conductivity (Fig. 4.3), for which most of the rain input leaves the domain without infiltrating in the soil. When spatial and temporal discretization are refined, our model converges towards a slightly different solution than K&M, which can be explained by the different coupling approaches. The value of α_{FOEC} chosen by K&M is not specified but it is expected to be smaller than our value of σ_B in the 1cm case and larger in the 5cm case. Indeed, the infiltration is less important in their 1cm case, due to a weaker coupling. The opposite can be seen for the 5cm case. This is seemingly due to the use of a similar value of α_{FOEC} for both their 1cm and 5cm cases, resulting in different fluxes (see section 4.4.3 for details). Our model is less sensitive to a coarser spatial discretization, as the σ_B adapts itself to the subsurface element size. The coarse temporal resolution of 180s is less appropriate for our explicit subsurface model, but it can still be handled with additional sub-iterations ($q = 10, m = 2$).

The case with the higher soil conductivity is shown in Fig. 4.4. A space-time resolution of 1cm/180s has proven to be too coarse in time for the explicit solver. We found that a 1cm/10s resolution was required, which increased the number of sub-iterations to $q = 10$ and $m = 2$. Our model is in good agreement with the K&M one, except at the end of the simulations where a change of regime occurs. This is due to a complete saturation of the soil, which means that runoff is non-Hortonian at the end of the simulation. The soil pressure head has therefore to switch from an infiltration profile to an incompressible water table profile, which are very different. The explicit model needs extra iterations when this abrupt change occurs, to produce the same results in the 5cm cases. The discretization of 1cm was too fine to converge towards this regime change with a reasonable number of iterations. It can be seen again that our model is less sensitive to the coarser discretization than the K&M model, seemingly for the same reasons as above.

Coupled model – Sandbox

Abdul and Gillham [107] have studied the effect of the capillary fringes over the overland flow generation. The model domain consists in the sandbox shown

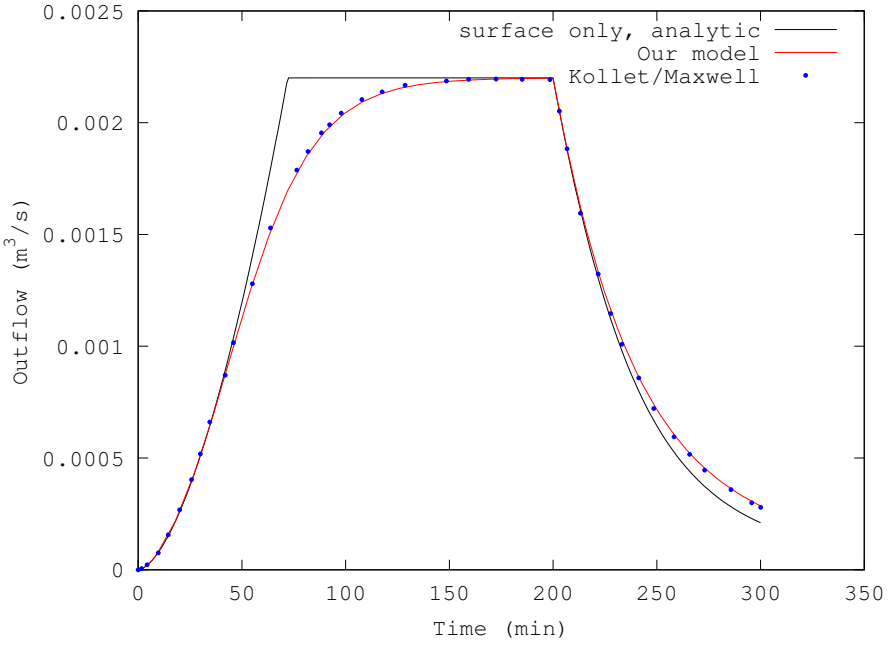


Figure 4.2: Analytical solution of a rain outflow over a simple impermeable slope and its resolution by our model and the Kollet & Maxwell model. The models results use coarse time and space discretization (180s, 80m), which can explain the divergence from the analytical solution.

in Fig. 4.5 with its related parameters. The geometry allows the use of a 2D model for subsurface flow and 1D model for surface flow. The water table is initially at a steady state and at the same height as the outlet located at the bottom of the surface slope. A rainfall is applied uniformly for 20 min at a constant rate of 4.3 cm/h. The discharge is monitored from the beginning of the rainfall to almost the end of the recession period.

The 2D mesh is unstructured and made of 252 triangles with a higher resolution near the surface. The characteristic lengths of the elements goes from 2.4 to 25.1 cm. The 1D surface model is discretized according to the topmost segments of the 2D mesh. A constant time step of 0.1 s is used. The beginning of the simulation requires an increased number of iterations for ψ, so m is manually set to 100. After 200 s, m is set to 10. During the whole simulation, the maximum number of iterations for the filter q is set to 4.

Different models are presented for comparison in Fig. 4.6. The temporal and spatial discretizations are of 10s and 1 cm×4 cm quadrilaterals for ISWGM

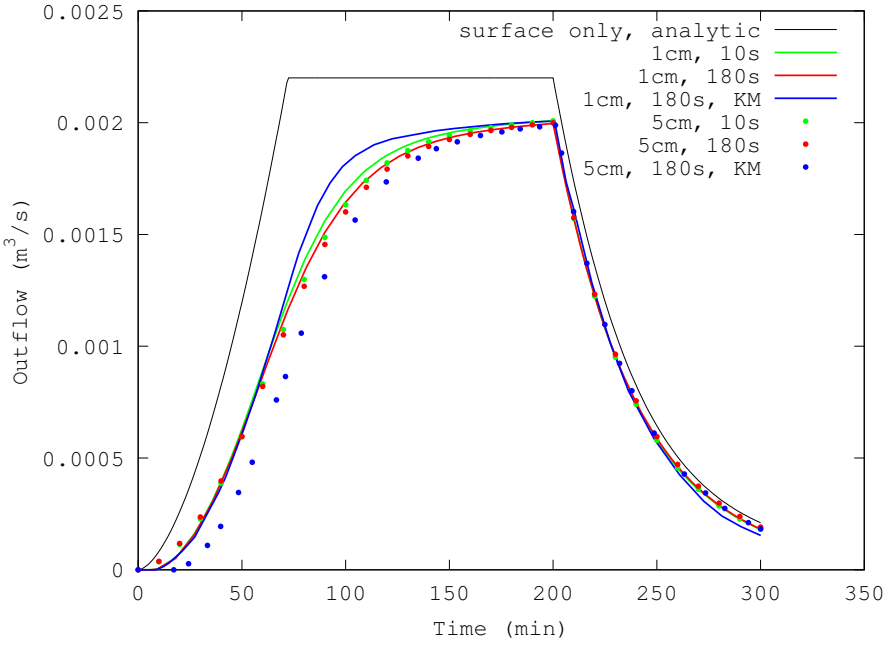


Figure 4.3: Outflow for a simple slope over homogeneous soil with $K_s = 1.1574 \times 10^{-7}$ m/s. The results obtained by K&M [62] are shown in blue. The other results are from our model. In those runs, q and m are kept to 1 excepted for the 1cm/180s case which needed more convergence with $q = 10$ and $m = 2$. It can be seen that our model converges towards a slightly different solution than the K&M one as the spatial or temporal resolutions are refined. Our results are less influenced by the coarser spatial resolution. The analytical solution of Fig. 4.2 is kept for comparison.

[106], of 0.1 to 10s and 2 cm×2.8 cm quadrilaterals for Cast3M [19], of 10 s and 1 cm×2 cm quadrilaterals for Parflow [62] and of 60 s (max) and 2 cm sided squares for InHM [58]. Fig. 4.6 shows that our model is consistent with experimental data and other published models results. The origin of the discrepancy between the models results is difficult to assess as the model resolutions are different. It should be noted that all models reach a steady state slightly ahead of the real system. K&M suggested that this is due to the presence of air phase compression in the experiment [62]. As the simulated flow profile is very similar to those already published, it is not shown here.

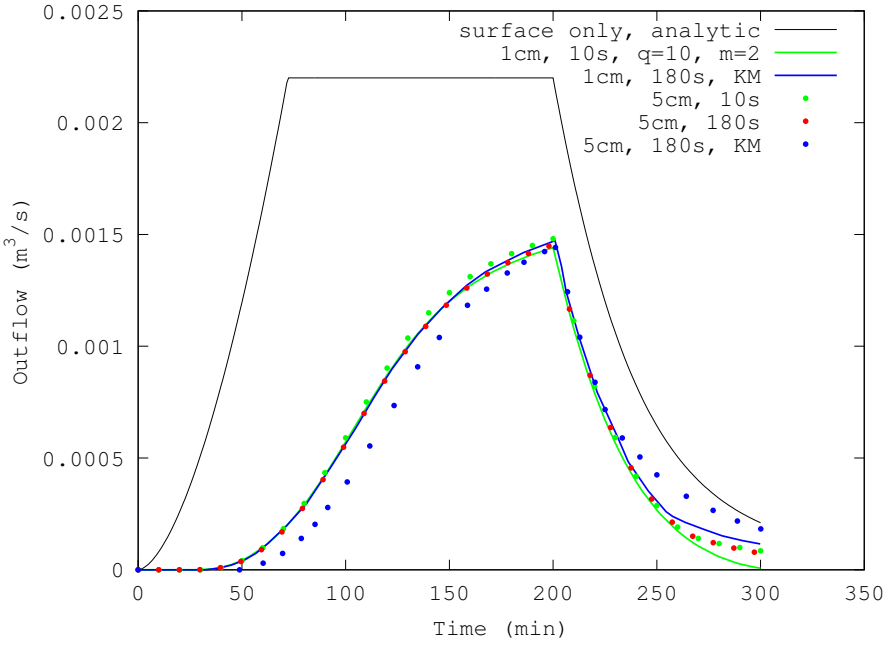


Figure 4.4: Outflow for a simple slope over homogeneous soil with $K_s = 1.1574 \times 10^{-6}$ m/s. The results obtained by K&M [62] are shown in blue. The other results are from our model. The q and m for the 5cm/10s and 5cm/180s cases are 1 excepted between 258-267 min where they were manually set to 100 to achieve convergence. Our model results are close to the fine-resolution result of K&M, even for the coarser resolution. The analytical solution of Fig. 4.2 is kept for comparison.

4.4.2 Integrated 3D modeling example

A field-scale experiment has been performed by Abdul [109] in a small catchment of the Canadian Forces Base Borden, Ontario, Canada. The area of 1070 m^2 is grass covered except for a channel approximately 60 cm wide, located 22 cm below the streambed. The sandy soil is characterized by the following parameters: $\theta_s = 0.34$, $\theta_r = 0.0612$, $\alpha = 6$ m^{-1}, $\beta = 1.9$. $\mathbf{K}_s$ has an isotropic value of 10^{-5} m/s. The initial water table is at a depth of 20 cm below the channel.

The 3D mesh is constituted of triangular prisms vertically extruded over 6 layers from the 2651 triangles that constitute the surface mesh, for a total of 95.436 subsurface DOF's (see Fig. 4.7). The surface mesh is refined in and near the channel while the subsurface mesh is refined near the soil surface. The time step is set to 5 s. The parameters m and q that specify the number of iterations

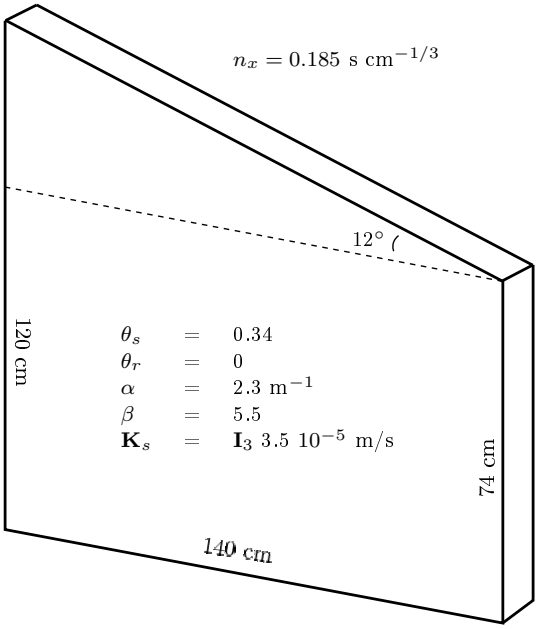


Figure 4.5: Geometry and physical properties of the sandbox test case. A uniform rainfall is applied on the top and the unique outlet is situated at the top right corner of the domain.

are set to 4 and 1, respectively. A uniform rain of 5.55×10^{-6} m/s is applied during 50 min.

Most of the surface-subsurface models use this test case as benchmark. The four we are showing use finite difference (MODFLOW) or CVFE (Groundwater, Hydrogeosphere and Vanderkwaak) schemes in space and Euler implicit schemes in time. They are all using the non-inertia approximation of the shallow water equations. The spatial mesh of the surface is the same for all models. The vertical resolution at the surface is coarser in our case: 10 cm compared to 1 cm, which leads to 15 layers for the other models.

In this test case, the FOEC coupling described by Eq. (4.12) is used. Indeed, it appeared that the hybrid coupling underestimated the surface flux, as most of the rain infiltrates. This is the physical behavior with the present parametrization for a dry soil, as the rain flux is below the saturated conductivity. Therefore the rain has to infiltrate until non-Hortonian overland flow occurs. As experimental data show a non-negligible amount of surface water, we deduce that something slows down the infiltration. It may be the vegetation or some surface

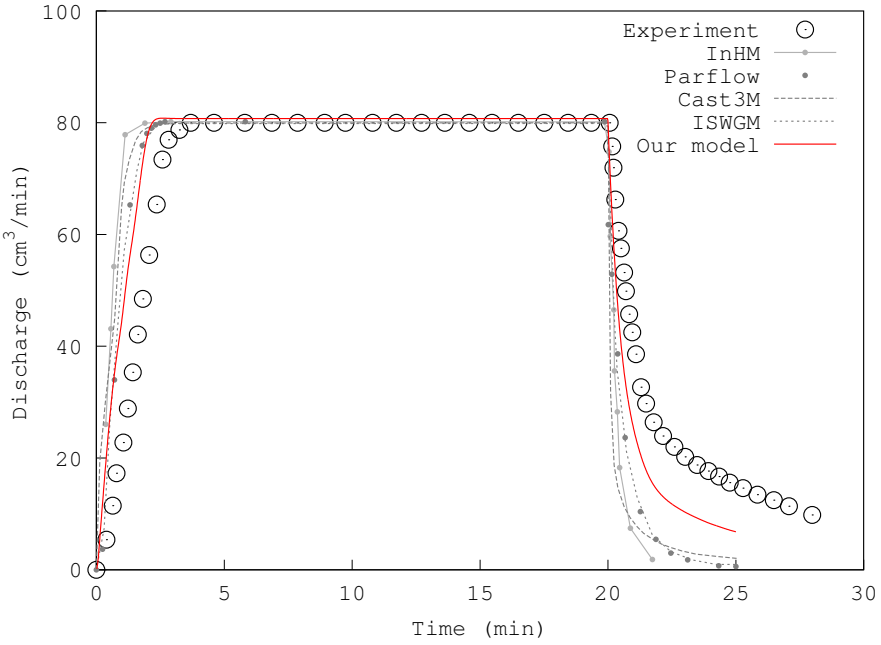


Figure 4.6: Derived and simulated hydrographs of the sandbox test case and our model results are in good agreement with observed data.

compaction of the soil. We therefore had to calibrate the model outflow with a specific value of a_s to compute α_{FOEC}, as suggested by Ebel et al. [100]. Manual calibration leads us to take $a_s = 0.25$ m. This value is around 0.1 m for Hydrogeosphere [18], and 10^{-4} m for GroundWater and InHM [64, 58]. These discrepancies for a_s may be due to additional parameterizations for the surface model as rill storage or microtopography, and from the depth difference of the first layer of discrete elements (see Section 4.4.3). Despite the coarser vertical resolution, the hydrograph obtained with our model is close to observed and modeled values (see Fig. 4.8).

The origin of the early delay between most of the simulated results and the observations is difficult to assess. A possible explanation is that a local surface storage is present (as done in [64, 18]) and has to be filled before any outflow can occur, or that the initial rain is overestimated. Both cases can delay the measures. Another possible explanation is that the runoff is non-Hortonian. In this case, it it possible that the initial water table depth is above the specified value, or that the subsurface water is not at equilibrium and more water is present in the unsaturated area [96], or that the rain is underestimated. In such

cases, it is possible to reach a non-Hortonian runoff with the hybrid coupling earlier in the simulation and hence explain the absence of outflow during the infiltration.

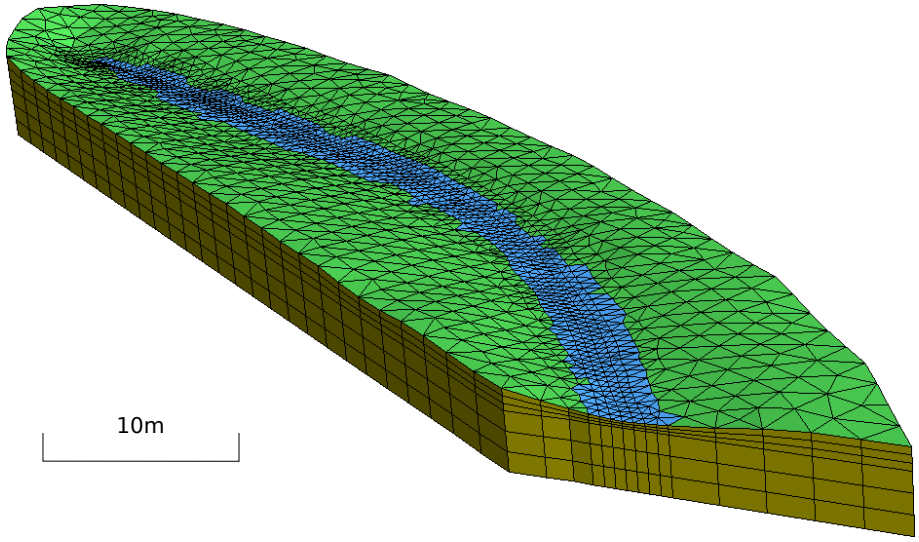


Figure 4.7: The 3D mesh used for the Borden test case. The channel is displayed in blue, grass land in green and the subsurface part in brown. The surface mesh is refined inside the channel. The first three of the six vertical layers are refined to represent the larger fluctuations present in the unsaturated zone.

4.4.3 Sensitivity analysis

Two issues were experienced during the model implementation. Both were linked to the spatial discretization. The first one is related to the CVFE method used for the surface model that is not well suited to anisotropic elements. The second one concerns the FOEC method used to add extra physics at the soil-surface interface that is strongly dependent of the subsurface discretization. In this section, we describe those issues and present the solutions that we implemented in our model.

Surface model – anisotropic elements

The above V-catchment test case produce unphysical results when the CVFE method is applied with a coarse discretization of quadrangles. We have there-

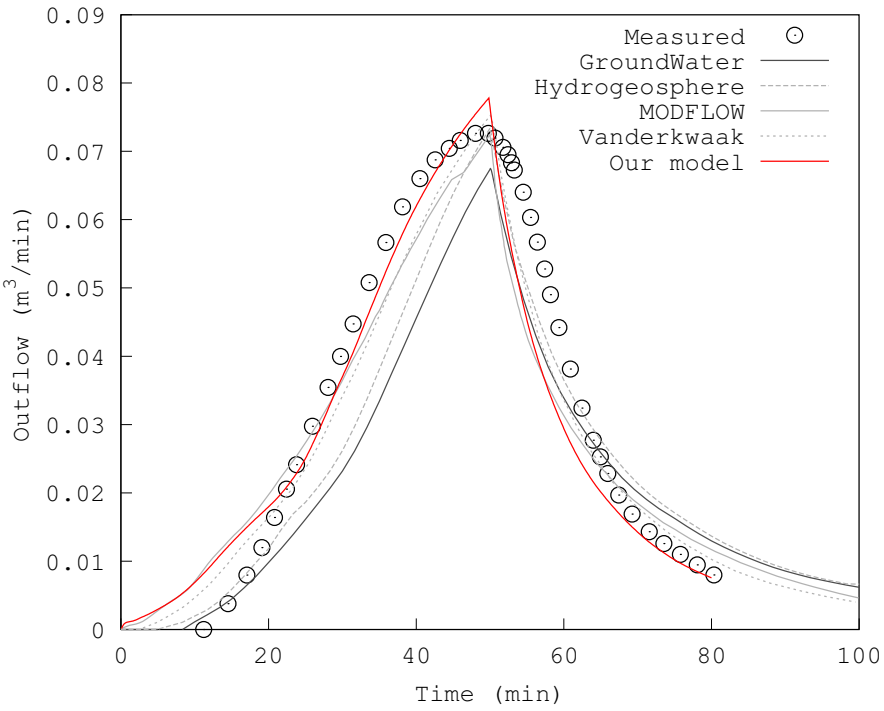


Figure 4.8: Hydrographs obtained after a 50 min rainfall event over the Borden catchment. Our model results are in good agreement with observed values and other state-of-the-art models.

fore studied the effect of different mesh resolutions on the model results. The time discretization error is negligible as the time step is set to 1 s.

When the mesh resolution is larger than 10 m, we have to use rectangular elements in order to still be able to represent the 10 m wide channel of the V-catchment. The anisotropy of the mesh elements and the non-linearity of the equations can produce unphysical fluxes going in the wrong direction if the influence coefficient method is not modified. Fig. 4.9 shows that for the 40 m discretization the approximate CVFE method keeps a correct value for the discharge when the mesh is coarsened, which is not the case with the original CVFE method. This is highlighted for the 100 m discretization, where the result is just slightly different with the modified method, but totally wrong with the classical method for which unphysical fluxes are clearly visible in the channel (not shown). The results – with and without the approximation – for the mesh with a discretization of 10 m are not differentiable graphically as the

mesh is entirely made of square elements (*i.e.* isotropic) and can therefore not produce negative coefficients.

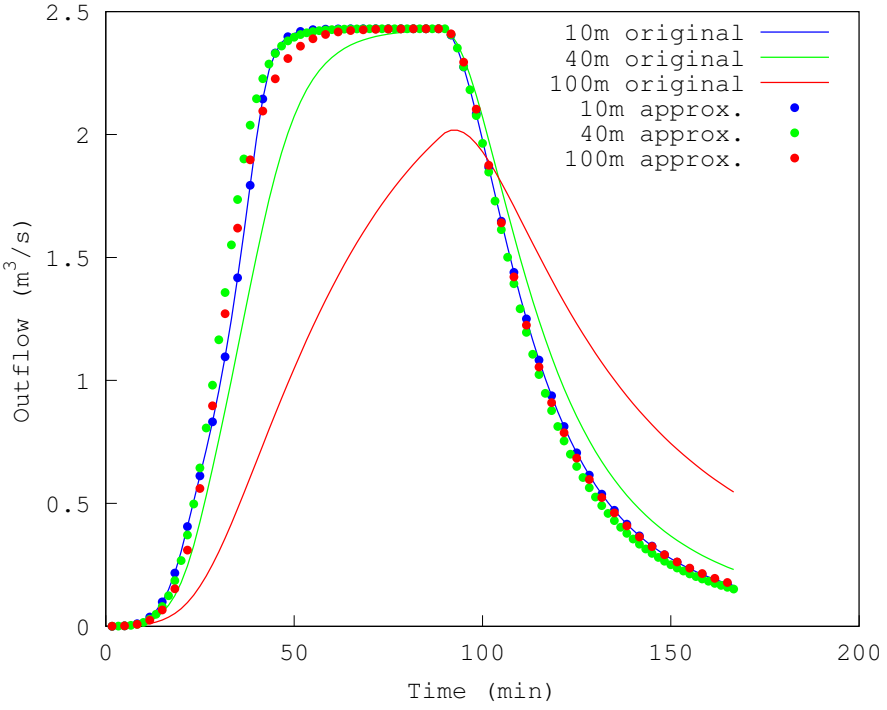


Figure 4.9: The outflow of the V-catchment experiment is greatly influenced by anisotropic elements when using the original CVFE method. The proposed approximation, in which the negative influence coefficients are set to zero (*i.e.* only positive coefficient are kept), greatly improves the results on coarser meshes as compared to the original CVFE formulation.

Coupled model – Effect of α_{FOEC}

It has been observed in the Borden test case that the first layers of the subsurface discretization have a large influence on the infiltration. To highlight this aspect of the FOEC coupling, a 1D vertical example has been considered, where the surface water can only flow in the soil or accumulate. The same properties as in the Borden test case are used, *i.e.* the depth, rain and initial conditions correspond to the ones of a column in the middle of the 3D channel. A uniform vertical discretization is used with different resolutions ranging from 1 cm to 80 cm. The surface water depth is monitored during 75 min (see

Fig. 4.10). When refining the subsurface discretization and keeping α_{FOEC} constant, infiltration decreases, and the surface water depth thus increases. This effect vanishes when the grid size reaches 5 cm and the infiltration front is sufficiently resolved. However, the use of a large vertical discretization is desirable for large-scale and/or long term studies. One simple explanation to the phenomenon is that large subsurface elements have a larger storage capacities. They are therefore less sensitive to a given influx than smaller ones. The coupling flux based on a nearly constant value of the pressure head is also nearly constant, and therefore remains important. To circumvent this issue it is possible to increase the value of α_{FOEC} and to use it as a fitting parameter like in the Borden test-case. Its optimal value to fit experimental measurements is then obviously problem and mesh-dependent.

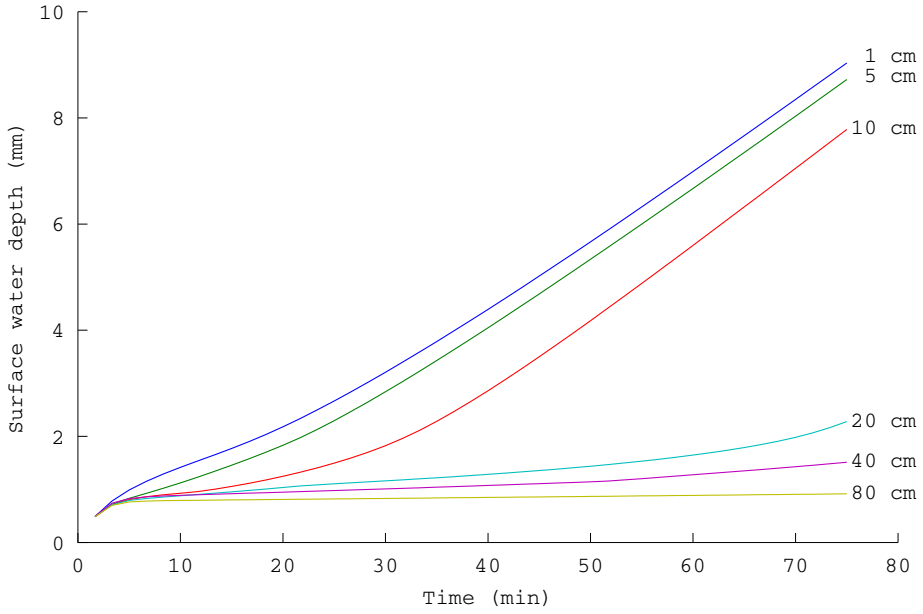


Figure 4.10: The surface water height obtained with a single 1D infiltration model without surface fluxes strongly depends on the vertical grid resolution. The grid size has to be fine enough to capture the infiltration front otherwise the $h - \psi$ difference thus the coupling flux stay important.

Effects of the near-surface vertical resolution on the results have already been discussed for subsurface models. It has been observed in the case of a 2D finite volume model that the vertical discretization highly affects the results

[110]. Downer and Ogden [23] point out the necessity to use fine vertical discretization at soil surface (∼1cm) in a finite difference model with a Dirichlet (when ponding) or Neumann (when dry surface) boundary condition coupling. They showed in particular that the amount of infiltration is highly dependent on the element size. It increases as the grid size decreases. They highlight the importance of the evaluation of K in their coupling. Indeed as the value of K in initially dry soil is very low, classical boundary fluxes relying on local K values are strongly under-estimated. Their observations are however not relevant to our issues as in our case the amount of infiltration *decreases* as the grid size decreases. We already circumvented their issue by taking the upwind value of K in interface fluxes. This means that $K = K_s$ at the surface when infiltration occurs and the surface is thus considered as either fully saturated or dry.

4.4.4 Model efficiency

The model efficiency with the hybrid coupling is assessed based on strong and weak scaling test cases. The former highlight the ability of a model to use small computational domains per nodes, which lead to an increasing domain surface/domain volume ratio. That implies an increase of the intra-nodes communication then a decrease of the scalability. A model still scalable with a large ratio has a good strong scalability. The latter is used to assess the capacity of a model to extend its computational domain still keeping a good scalability. This is certainly a good property for large-scale simulations.

Strong scaling

The first 50 s of the Borden test case where taken as benchmark to assess the model strong scaling. For those computations, two additional meshes of 4.48×10^5 and 1.37×10^7 DOF's have been used. With 192 processors, the load per processor is then of 497 DOF's for the mesh previously used and respectively of 2.9×10^3 and 7.15×10^4 DOF's for the two larger ones. The 6 layers of depth were kept, leading to a subsurface/surface DOF number ratio of 7. To limit the side effect of memory-sharing between the processors of a same node, we set a whole node of 12 processors as the reference. The partitioning of the whole domain into sub-domains has been done for the surface model. This sub-domain and the soil column under it are associated to a unique processor core.

The results in Fig. 4.11 show a good scaling for the larger problem, a correct one for the medium problem and a poor scaling for the small problem. The identification of the scaling performance for the different parts of the algorithm highlights the source of the poor scaling. Although the solution of subsurface model is the most expensive, it scales optimally as shown in [6]. However, the surface part relies on an implicit solver which requires many inter-processor

communications, especially when the surface water is at rest. Therefore the scaling of the surface model decreases when to the number of DOF's per processor goes down. The summation of these two effects does not preclude an overall good scaling, as long as the load associated with the surface model is below the load of the subsurface model, which is one order of magnitude larger on one processor in this test case.

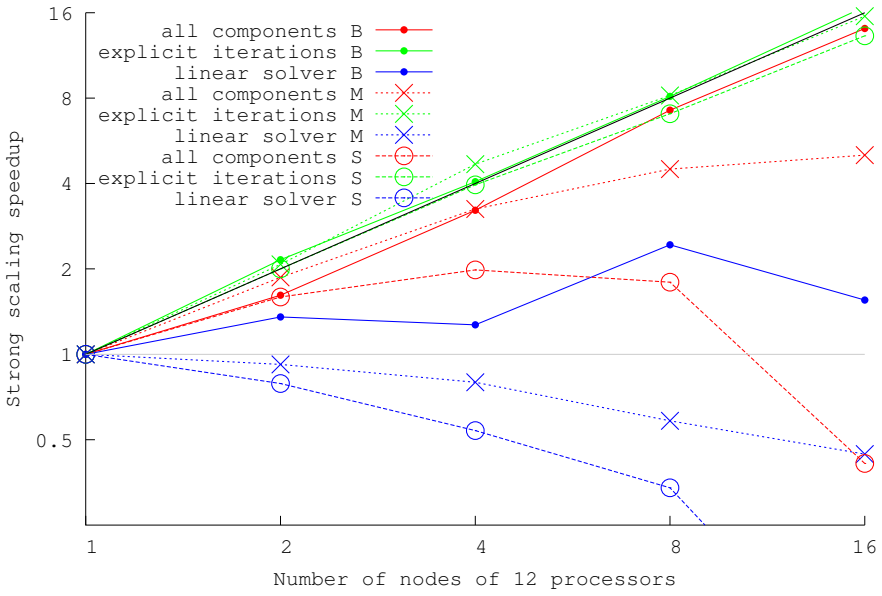


Figure 4.11: The black line represents an optimal strong scaling, *i.e.* doubling the computational power doubles the speed. The letters are for the small mesh (S), the medium mesh (M) and the big mesh (B). Two components of the model are highlighted: the explicit iterations from the subsurface model and the linear solver from the surface model. The model strong scaling is optimal for the explicit discretization of the Richards' equation, but not for the implicit discretization of the non-inertia shallow water equation, as expected. The latter reduces the scaling of the overall model but can handle much bigger time steps than an explicit discretization of the non-inertia equation.

Weak scaling

The weak scaling is interesting as it does not benefit from the extra resources coming from the additional nodes, as each new node has to handle the same computational burden as the others. The scaling is then restricted to a maximum of 100%. Obtaining a good weak scaling is often sufficient for fine res-

olution and/or large scale applications that allocate a large sub-domain per processor. Maxwell [54] has presented a non-dimensional test case optimized for weak scaling. It consists in a sinusoidaly-shaped layer of homogeneous soil. We slightly differ from this test case by considering that $\tan(x) \simeq x$ when x is close to zero, resulting in the topography formula $z = \sin(x/5)/2 - \cos(y/5)/2$. The problem size per processor is decreased from a 100×100 to a 50×50 square, with a constant depth of 25. The regular mesh is composed of hexahedrons with 2×2 horizontal sides and a depth of 1. In this test case, all quantities are dimensionless. The number of DOF's per processor has then been decreased from 5×10^5 to 1.25×10^5.

The soil properties are set to $\mathbf{K} = 0.25 \times \mathbf{I}_3$, $\theta_s = 0.25$, $\theta_r = 0.025$, $\alpha = 1$ and $\beta = 3$. The initial ψ is set to -10 at the ground surface, hydrostatic equilibrium being imposed vertically *i.e.* the water table is following the topography, and is then initially out of equilibrium. No rain is applied, which means that the surface domain remains completely dry. The duration of the simulation is set to 10 with a time step of 2.

Fig. 4.12 shows the weak scaling result for this test case. Our model exhibits a nearly-optimal weak scaling. The negative effect of the surface model on the scaling is absent as its solution is trivial when the surface is dry. As this test does not show the ability of the whole coupled model to achieve a good weak scaling, we have added a homogeneous time-dependent rainfall $r(t) = 0.2 + 0.01t$, which triggers a surface flow. The resulting scaling is depicted in Fig. 4.12, and like with the strong scaling test case, the weak scaling of the coupled model is negatively impacted by the surface model.

4.5 Conclusion

In this study, we have presented a new coupled model for the surface-subsurface water interaction. The subsurface flow is described by the Richards' equation, which has been discretized in time with an Euler explicit scheme and in space with a first-order DG FE scheme. The surface flow is described by the non-inertia approximation of the shallow water equations. It is discretized with an implicit scheme in time and a control volume finite element scheme in space. Both models are coupled by weakly imposing the continuity between surface and subsurface pressures.

The CVFE method is well designed for the non-inertia shallow water equation as it naturally handles the zero-depth issue. However, the influence coefficients of this method can become negative for anisotropic elements. Negative coefficients lead to unphysical fluxes between mesh nodes, which can severely impair the model accuracy if left unchecked. We have shown that by simply setting negative coefficients to zero we can avoid that issue.

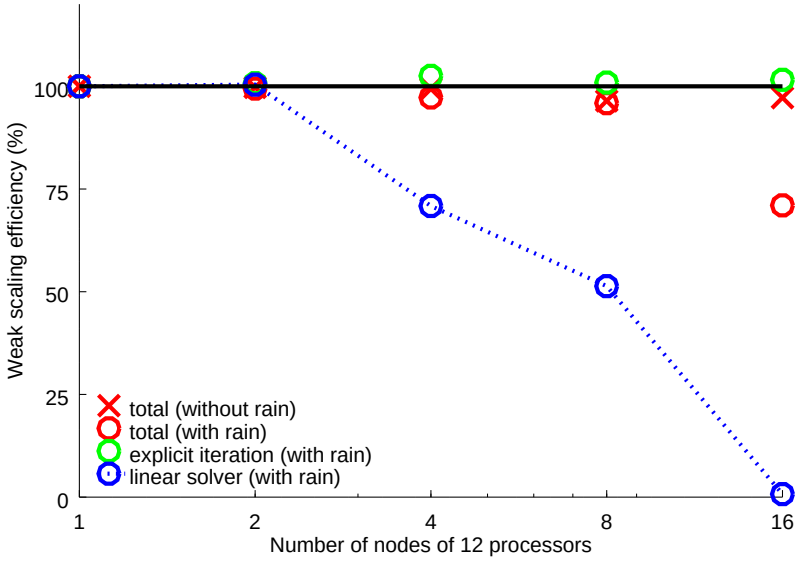


Figure 4.12: Weak scaling of the model over up to $16 \times 12 = 192$ processors. The black line represents an optimal weak scaling. In the absence of a surface flow, the model weak scaling is optimal as expected from the explicit time discretization of the subsurface model. When there is a surface flow, the model overall scaling is impacted by the suboptimal scaling of the linear solver used for the implicit time discretization of the surface model. The poor scaling of the surface model is particularly clear when more than 96 processors are used.

The proposed coupling lies between the pressure continuity coupling and the FOEC coupling methods. Indeed, it uses both a weak imposition of the continuity between the surface hydrostatic pressure and the subsurface pressure head, improving the scheme robustness, and it reduces to the pressure continuity coupling when the surface and subsurface pressures have converged. The FOEC can be used to add sub-scale surface-subsurface interactions as a slower interface conductivity, which is not the case of our hybrid coupling. However the FOEC flux can be discretization-dependent if the infiltration front is not fully resolved, which means that it has to be used with caution for coarse vertical discretization, as it is often the case in large-scale and/or long-scale studies. With the hybrid coupling presented in this study, the coefficients are element-size dependent and are coherent with the inter-element treatment of the rest of the subsurface domain. The proposed coupling comes handily along with the DG FE discretization without having to fix a free parameter value, and seems more robust for coarser subsurface discretization. This coupling also uses a

unique time step which is designed for both surface and subsurface models. Indeed, an explicit scheme is well suited to the slow dynamics of the subsurface flow, while the fast surface flow requires an implicit solver. By coupling then two sub-models together, we can use a unique nearly-optimal time step for the global model.

The explicit discretization of the Richards' equation leads to a perfect strong scaling. The surface model being implicit, it cannot achieve the same optimal scaling as the subsurface model. However, as the computational load of the non-inertia equation is generally one order of magnitude smaller than the load of the Richards' equation, the negative influence of the implicit scheme on the overall scaling remains limited. As a result, the proposed coupled model shows both good weak and strong scalings. We believe that such a modeling approach will prove useful in the future as the current trend in high-performance scientific computing is to favor large scale parallel architectures that require scalable models. However, to become an operational large-scale and long-term model, several improvements are still mandatory. First, a purely elliptic solver should be used in the saturated areas, such as the multi-grid method. That would slightly degrade the scaling, but it is necessary to handle large water tables. Secondly, an adaptive time stepping scheme which also adapts the number of sub-iterations is required to automatically optimize the performances over time, while keeping a specified accuracy. Eventually, evapotranspiration will have to be taken into account for long term studies, as it is a key component of the terrestrial water cycle.

CHAPTER 5

Implicit-explicit predictor-corrector subsurface model

Abstract

Richards equation models are used for many applications in soil sciences and groundwater engineering. However, this equation is highly non-linear. Computing a numerical solution is therefore very challenging, especially for full-scale simulations. A model, to be able to handle large domains throughout a long time period must optimize five factors: the precision, the robustness, the sequential speed, the scaling on parallel infrastructures and mass conservation. In this chapter, we describe a two-step algorithm to solve Richards equation while trying to maximize those five factors. The first step is an approximation of the pressure head with a unique implicit sub-iteration. The second is the application of that result on the water content through an explicit iteration. The resulting model is close to a fully-explicit model and hence scales well. The cost of one time step is reduced compared to a fully-implicit model and the model robustness is increased by avoiding non-linear sub-iterations. Mass is conserved at machine precision. The limiting factor to further improve the scaling is the linear solver. Our model combines several advantages: the precision is similar to existing models; it scales both weakly

and strongly, up to 192 processors; it performs robustly; its speed on a generic framework is appreciable. We believe that our approach could significantly improve the efficiency of subsurface flow models based on the Richards equation.

5.1 Introduction

Richards Equation (RE) is mainly used to model the underground hydrodynamics. It combines three physical processes in one equation: the advection-dominated infiltration fronts (from rain), the diffusion-dominated unsaturated area (capillarity) and the incompressible saturated area below the water table. The pressure head ψ and the water capacity θ are commonly used as variables, the first one to describe the dynamics, the second to reflect mass conservation. When RE is fully described in terms of θ, it is known to be efficient in unsaturated areas (advection/diffusion), but it cannot be used in saturated areas. On the other hand, when RE is expressed only in terms of ψ, it can be used everywhere but it is no longer mass-conservative. The elliptic behavior (*i.e.* incompressibility) in saturated areas imposes the use of an implicit time integration scheme.

Many state-of-the-art models are using the modified Picard method [27, 20], a mixed-form of RE including both ψ and θ. It consists in a slight modification of the time integration scheme and the non-linear solver. This method is similar to the ψ-form but asymptotically conservative, it is compatible with any spatial integration scheme and is easy to implement.

Some predictor-corrector RE models are using consecutively the ψ and θ variables [30, 16, 41, 111]. Their general idea is to find a prediction for ψ – valid everywhere – then use it to compute the fluxes with θ. The resulting unsaturated dynamics is more efficient and shows a better mass conservation. An explicit time integration scheme is used either for the predictor or the corrector as it is computationally cheaper. Another explicit model for RE has shown a good behavior for unsaturated areas and infiltration fronts, a perfect scaling and machine-precision mass conservation [6]. However, the dynamics in the saturated zone is strongly impacted by the approximation used to solve explicitly the elliptic part of the RE.

Explicit time integration schemes have more constraints on the time step than implicit ones. However, with the RE, implicit models are nonetheless constrained to use reasonable time step values to ensure the convergence of the non-linear solver. In contrast, explicit iterations are cheap and much more easily scalable. When correctly implemented, explicit schemes can yield a perfect scaling [6]. Indeed, their inherent simplicity and – with mass lumping – the absence of a linear system solver reduces the communication overhead between processors. When first-order, either explicit or implicit schemes yield the same

level of accuracy. The issues of explicit schemes with RE is the elliptic behavior and the transition towards that behavior, which can lead water content to exceed the saturation limit.

Despite all the research performed on RE models, their scalability on parallel infrastructures is often poor. It is a difficult task as the physics occur on a broad range of scales in both space and time. Areas with a fast dynamics require more computational power than areas evolving very slowly. This load can be hard to balance between the workers[1] of a computer cluster.

A model with the advantages of an explicit time integration scheme (fast, scalable) and those of an implicit one (well suited to elliptic problems and allowing large time steps) is of course desirable. Here, we want to relax the time constraint on the explicit model and to reduce the computational burden of the implicit model. A perfect model should also be precise, mass conservative and robust.

In this chapter, we present a discontinous Galerkin (DG) finite element (FE) model build with an Euler-implicit linearized predictor for the pressure, followed by an Euler-explicit corrector for the water content. Mass conservation is ensured at machine-precision and the precision is similar to the one achieved in other studies. The use of exactly the same fluxes in both equations allows the simplification of the explicit scheme into a simple vector operation. Moreover the explicit solution is close to an implicit one (*i.e.* it is implicit if mass balance is reached within the predictor), relaxing the constraint on the time step.

5.2 Mathematical formulation

The 3D subsurface model relies on the three-dimensional RE, which can be expressed as follows:

$$\frac{\partial \theta}{\partial t} = \nabla \cdot (\mathbf{K} \cdot \nabla(\psi - z)) + s, \tag{5.1}$$

$$\theta = f_\theta(\psi), \tag{5.2}$$

where θ [$\mathrm{L^3L^{-3}}$] is the volumetric soil water content, ψ [L] is the pressure head, z [L] is the upward positive vertical coordinate, s [$\mathrm{T^{-1}}$] a sink-source term, $\mathbf{K}$ [$\mathrm{LT^{-1}}$] the water conductivity tensor and f_θ [–] the retention curve. Eqs. (5.1)

[1] As it is difficult to put a single name on a unit of computation, we will use the word worker. Indeed, such a unit can use several threads, or a fraction of a processor, or half of a core, etc. The word CPU is also known to be ambiguous.

and (5.2) are complemented with appropriate initial and boundary conditions:

$$\psi = \psi_0, \qquad \text{on } \Omega, \quad t = 0 \tag{5.3}$$

$$\psi = \psi_D, \qquad \text{on } \Gamma_D, \ t \in [0, t_{end}] \tag{5.4}$$

$$-(\mathbf{K} \cdot \nabla(\psi - z)) \cdot \mathbf{n} = J_N, \qquad \text{on } \Gamma_N, \ t \in [0, t_{end}] \tag{5.5}$$

with $\mathbf{n}$ [–] the outward normal vector, t_{end} [T] the simulation duration, Ω the computational domain, Γ_D the Dirichlet part of the boundary (where the value ψ_D is imposed) and Γ_N the Neumann part (where the flux J_N is imposed). The constitutive relations defining $\theta = f_\theta(\psi)$ and $\mathbf{K}$ are derived from van Genuchten [102] and Mualem [103]:

$$S_e = \frac{\theta - \theta_r}{\theta_s - \theta_r}, \tag{5.6}$$

$$S_e = \begin{cases} (1 + |\alpha\psi|^\beta)^{-\nu} & \text{if } \psi \leq 0, \\ 1 & \text{if } \psi > 0, \end{cases} \tag{5.7}$$

$$\mathbf{K} = \mathbf{K}_s \ S_e^{l_p} \left(1 - (1 - S_e^{1/\nu})^\nu\right)^2, \tag{5.8}$$

where S_e [–] is the effective saturation, θ_r [$\mathrm{L^3L^{-3}}$] is the residual volumetric water content, θ_s [$\mathrm{L^3L^{-3}}$] is the saturated volumetric water content, $\mathbf{K}_s$ [$\mathrm{LT^{-1}}$] is the anisotropic saturated water conductivity tensor, α [$\mathrm{L^{-1}}$] is a parameter related to the air-entry pressure value, β [–] is a parameter related to the pore-size distribution, and $\nu = 1 - 1/\beta$ [–].

Eq. (5.1) can be expressed in term of ψ only:

$$C\frac{\partial \psi}{\partial t} = \nabla \cdot (\mathbf{K} \cdot \nabla(\psi - z)) + s, \tag{5.9}$$

where the capillary capacity $C = \partial\theta/\partial\psi$.

5.3 Space and time discretizations

Eqs. (5.1) and (5.9) are discretized in space with the DG FE method. By partitioning the domain Ω into N non-overlapping elements Ω_e with interfaces Γ_e, the spatially and temporally continuous model variables θ and ψ can be approximated by the discrete variables θ^h and ψ^h as

$$\theta(x, y, z, t) \simeq \theta^h(x, y, z, t) = \sum_{j=1}^{N_d} \theta_j(t)\phi_j(x, y, z), \tag{5.10}$$

$$\psi(x, y, z, t) \simeq \psi^h(x, y, z, t) = \sum_{j=1}^{N_d} \psi_j(t)\phi_j(x, y, z), \tag{5.11}$$

where N_d [–] is the total number of degrees of freedom (DOF) and ϕ_j [–] are piecewise first-order Lagrange polynomials defined on each element Ω_e. The jump $[\cdot]$ and averaging $\{\cdot\}$ operators on the interface Γ_e are defined as:

$$[x] \triangleq x^+ - x^-, \qquad \{x\} \triangleq \frac{x^+ + x^-}{2}, \tag{5.12}$$

where the '+' and '–' superscripts indicate the trace value taken either on one or opposite side of Γ_e. At the boundaries, both operators are defined in terms of an external value derived from the Dirichlet boundary condition. The discrete equations are obtained by deriving the Galerkin formulation of Eqs. (5.9) and (5.1) with the test functions ϕ_i $(1 \leq i \leq N_d)$. The interior penalty DG method is then applied. By using a matrix notation and capitalizing vectors variables, we obtain

$$\tilde{C}M\frac{d\Psi}{dt} = K\Psi - Z + P, \tag{5.13}$$

$$M\frac{d\Theta}{dt} = K\Psi - Z + P, \tag{5.14}$$

where the matrices M and K are obtained by assembling the following element-wise matrices, with n_e the number of nodes per element and $1 \leq i, j \leq n_e$ over each element e and its neighbors:

$$M_{ij}^e = \langle \phi_i \phi_j \rangle, \tag{5.15}$$

$$K_{ij}^e = \langle \mathbf{K} \cdot \nabla\phi_i \cdot \nabla\phi_j \rangle - \langle\langle \{(\mathbf{K} \cdot \nabla\phi_j) \cdot \mathbf{n}\} [\phi_i] \rangle\rangle, \tag{5.16}$$

where $\langle \cdot \rangle$ is defined as $\sum_{e=1}^{N} \int_{\Omega_e} \cdot \, d\Omega$ and $\langle\langle \cdot \rangle\rangle$ as $\sum_{e=1}^{N} \int_{\partial\Omega_e} \cdot \, d\Gamma$.. The vectors Z, P are assembled in the same way, with $1 \leq i \leq n_e$:

$$Z_i^e = \langle \mathbf{K} \cdot \mathbf{z} \cdot \nabla\phi_i + s\phi_i \rangle - \langle\langle \{(\mathbf{K} \cdot \mathbf{z}) \cdot \mathbf{n}\} [\phi_i] \rangle\rangle, \tag{5.17}$$

$$P_i^e = \sigma \left\langle\left\langle [\phi_i] \left[\psi^h\right] \right\rangle\right\rangle, \tag{5.18}$$

where the penalty parameters σ [T^{-1}] and σ_θ [LT^{-1}] are defined as:

$$\sigma = \frac{4}{l_e} \, (n_0 + 1)(n_0 + 2) \, K_{sc}, \tag{5.19}$$

$$K_{sc} = (\mathbf{K} \cdot \mathbf{n}) \cdot \mathbf{J} / \|\mathbf{J}\|, \tag{5.20}$$

$$\mathbf{J} = \mathbf{K} \cdot \nabla(\psi - z), \tag{5.21}$$

with n_0 [–] the order of the FE approximation (in our case $n_0 = 1$), K_{sc} [LT^{-1}] the normal flux-oriented scalar conductivity, $\mathbf{J}$ [LT^{-1}] the water flux, and l_e [L] a characteristic length of the two adjacent elements.

To increase the stability and efficiency of the model, the mass matrices in Eqs. (5.13) and (5.14) have been lumped. Following other model designs, the

diffusivity tensor $\mathbf{K}$ has been linearized over each element in terms of its nodal values [20, 18]. The reasons for the lumping and the linearization are explained with more details in sections 2.4.1 and 2.4.2.

Eq. (5.14) is discretized in time with an explicit Euler scheme, and Eq. (5.13) with an implicit Euler scheme, but with all the parametric functions evaluated at the last time step. Indeed, we are using the implicit time integration scheme only to solve the elliptic part of RE in saturated areas, where parameters are *de-facto* most of the time constant. That linearized predictor is close to a fully explicit model, and such a scheme has been proven to be accurate outside saturated areas [6]. A solution for that linearized equation can be obtained without the use of a non-linear solver (or with only one iteration of such a solver as the problem is linear by definition). As a classical issue is the lack of convergence of the non-linear solver, the robustness of our model, *i.e.* its ability to go through a whole complex simulation without being stopped, is better than for fully-implicit models.

5.3.1 Summary of the numerical solution procedure

The overall solution procedure for the coupled model is summarized below. For a given parametrization, an initial state (Ψ^0, Θ^0) and a time step Δt, we iterate over the following steps:

1. Solve the equation for the intermediate step Ψ^*, given Ψ^n:

$$\tilde{C}M\frac{\Psi^* - \Psi^n}{\Delta t} = K\Psi^* - Z + P, \tag{5.22}$$

   where $\tilde{C}$, K, Z and P depend on Ψ^n. This is done with a unique non-linear solver step.

2. Solve the equation for Θ^{n+1}, given Θ^n and Ψ^*:

$$M\frac{\Theta^* - \Theta^n}{\Delta t} = K\Psi^* - Z + P, \tag{5.23}$$

$$\Theta^{n+1} = \text{limit}(\Theta^*), \tag{5.24}$$

   where K, Z and P depends on Ψ^n. This is important to keep the same values than in the previous iteration to keep the implicit stability property. The limiter function balances the water content across element nodes, keeping nodal values between the maximum and minimum mean values of neighbor elements. Its algorithm is described in section 3.3.5.

3. Compute Ψ^{n+1} by combining Ψ^* and the retention curve relationship:

$$\Psi^{n+1} = \begin{cases} f_\theta^{-1}(\Theta^{n+1}) & \text{if both } \psi < 0 \text{ and } \theta < \theta_s, \\ \Psi^* & \text{elsewhere.} \end{cases} \tag{5.25}$$

The limiter is mainly used to bound the values of θ. This avoid undershoots on small values that are very sensitive to K and C, which can change very quickly. It certainly also contributes to improve the stability of the whole algorithm. An additional limiter for ψ has been tested but showed no additional benefit in terms of stability. As a limiter increases the numerical diffusion, we choose to remove it from the solution procedure. Consequently, it is possible to see non-monotonic behavior for ψ (some undershoots) due to the high non-linearities of the RE.

5.3.2 Synchronization of the variables

In specific cases, desynchronization between ψ and θ is possible. The simplest example is evaporation over an initially fully saturated soil by the means of a Dirichlet BC. During the first time step, the ψ-form is purely elliptic and instantaneously adapts ψ to the BC over the whole profile. Indeed, the mass factor C is zero in that case and is kept constant during that first time step, keeping the elliptic behavior even for negative values of ψ. On the other hand, θ keeps its initial value in the absence of fluxes, and thus does not match ψ according to the retention curve. The subsequent time step can then produce unexpected results as the new properties computed with ψ are unphysical. A solution would be to let C and K depend on the value of ψ taken at the next time step (in a fully implicit fashion) in saturated areas where those properties are most of the time constant. In that case, additional non-linear iteration(s) would be necessary, but only for such a situation.

5.3.3 Minimizing artifacts at the saturated/unsaturated interface

In areas with a sharp transition to saturation, the use of an explicitly computed C leads to overshoots for θ. Indeed, when ψ is switching between negative and positive values, the factor applied to the mass variation inside an element C also switches from a significant value to a negligible one. Here, during that transition C remains significant as it is kept constant during the time step. That leads θ to exceed its physical limit. Fully-implicit model avoid that issue by computing C with the new ψ value, but it can lead to stability and mass balance issues (see 2.4.3 for more details).

In this work, we minimize those artifacts by using a switch on the value of C. When the element is saturated ($\theta \geq \theta_s$) and has at least one node with a positive pressure head, we force C to be at saturation in the whole element (zero in our test cases). Indeed, in saturated zone ψ and θ are independent and $C(\psi)$ is then not anymore equivalent to $C(f_\theta^{-1}(\theta))$. The errors obtained are then minimal and do not affect the general behavior.

Other predictor-corrector models modify the flux or use post-processes redistribution of the excess of water [111]. In the fully-explicit model of [6] an artificial diffusion is applied on θ. It should be noted that both those operations are fully mass conservative. Any mass balance error in the classic ψ-forms of the equation is translated into conservative under- or overshoot. Applying the corrector transforms the hidden mass loss into visible mass excess.

Mass conservation and mass excess

Two different physics are present inside the RE: water-unsaturated areas driven by pore capillarity and water-saturated areas driven by water incompressibility. The pressure head switches from a suction if negative to a hydraulic head if positive. The water content switches from variable to constant values. The combination of those two very different physics produces highly non-linear relationships. Any discretization of this equation has to deal with the moving interface between them. It is common for any model of RE to have convergence issues at the interface, where the retention curve (that connects the two variables) has a nearly-discontinuous slope. That steep relationship along this interface can transform any convergence error into substantial mass balance errors.

In this work we do not seek to find the exact solution of the ψ-form of RE in our predictor. Indeed, in front of the linearization of the functions C and $\mathbf{K}$ with an extreme variability of values inside some elements, literally respecting this discretized equation is not meaningful for the physics involved. Since we proved in Chapter 3 that an Explicit Euler RE model produces results as accurate as Implicit Euler models (both first-order), we similarly used the relationships computed from the previous state in the predictor of this RE model (which is implicit). That statement predictably produces mass balance error in the predictor as we do not enforce the relationship between the pressure and mass variables. Instead of improving the coherence between variables in the predictor, we enforce it with the explicit corrector. The error, instead of producing a mass gain or leak, slightly changes the computed fluxes. During the transition of an element from the vadose zone to the saturated zone, some mass excess appears.

In our opinion, such mass excess is not critical for the simulation results. Indeed the errors mainly come from infiltration fronts, which are often fully driven by the BC, especially when they are constant like in most test cases. For instance a Neumann BC imposing a constant flux J results in a constant pressure head ψ behind the front (with $J = K(\psi)$). The front speed is determined from the start and any mass excess above $f_\theta(\psi)$ should be pushed forward without loss of precision: the amount of water coming inside the domain is determined at the domain boundary and pushes the whole profile down to the infiltration

front. Such a logic can also be applied in the case of a Dirichlet BC: the front position mainly depends on the BC, not really on the discretization scheme precision. In case of a manual displacement of the mass excess towards the front, the shape of the front can see little differences. However it does not really change the model results as this front is already very noisy in reasonable spatial discretizations. To support this idea, it is interesting to see that one of the newest RE models focuses only on the advective part, neglecting the diffusive term, to achieve a fast computation of infiltration fronts with under 1% of additional error on the solution values [112].

How to deal with the mass excess

The drawback of our quasi-explicit model is the transformation of the mass balance error into a mass excess. In this work, a large mass excess is avoided by using a switch on C. That can also be achieved by limiting the time step. However such logic is in opposition with one of our goals, which is to relax time step constraints to allow a faster computation. We propose three solutions for future works.

The first is already tested in Chapter 3, where an artificial diffusion equation is applied on the elements over saturation in an explicit iteration, maximizing the diffusivity parameter. The mass excess is then progressively transfered to unsaturated elements via the jump terms, or spread amongst other mass-excess elements until reaching a border where it can exit the domain. The drawbacks of this solution are the time step constraint linked to the size of the smallest mesh element, the computation of an additional equation and the intermediate state where the mass excess is spread but still present.

The second possible solution is to use the same idea with a quasi-elliptic equation on the saturated elements, by the means of an implicit method. It is not possible to use a fully elliptic equation or the concept of mass transfer is lost (the mass term would then vanish). If a weak Dirichlet BC is then set at the domain boundaries with $\theta = \theta_s$, the water is quasi-instantaneously transported to any available surrounding space. The incompressibility of the flow is preserved and the time step constraint is released. The drawback is the cost of an additional linear solution that is not expected to scale well.

A third possible solution is more pragmatic and similar to the 1D solution proposed by Lai et al. [111], but extended to higher dimensions. In their case, the extra water is simply transfered to the downwind element. The case where that element would already be saturated is however not considered. Such a practice seems quite rough but can make sense in the context of an infiltration front where mass excesses appear. Indeed, in advective physics the next state of a variable in an element mostly depends on its state in the upwind element. In a pure advective system, this upwind flux has to be taken as such. In our

model a mass excess appears when this flux brings more water volume than the space available in the element. Since the advective flux cannot be changed, the mass excess has to be put somewhere. The logical place is to pass it to a downwind element: the incoming flux then passes through the element directly towards the next one. On the other hand, the RE physics is far from purely advective. For instance, in the case of an infiltration front reaching a water table, the physics in the front change dramatically from advective to elliptic. In such an extreme case, this third solution will not work and an alternative procedure should be found. A possibility is to find a smaller time step to match the exact time when there is a balance between the remaining space and the incoming water volume. The switch of physics is then done exactly at this moment. Overall, this third solution bypasses the equation solution to keep the global mass movement, with some additional error in the shape the infiltration front.

5.4 Submodeling evaporation with the Kirchoff transform

The transient evaporation test case described in Section 5.5.1 raises an interesting numerical challenge. Indeed, a very strong gradient is produced at the surface boundary, leading to model instabilities. We decide to handle that case by using an element-wise submodel.

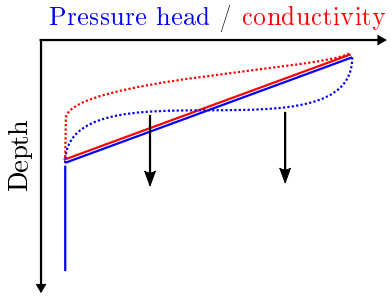
Submodels adjust the main model parameters using a specific knowledge at sub-element scale. They are often used to set other inputs/outputs. For instance, RE is often linked to root takeoff, interactions with chemicals, interactions with a clone of the model (dual-porosity), etc. Here we focus on a submodel improving the model itself, without additional inputs, by better handling non-linearities inside the elements. A method for unsaturated flow building a conductivity from a sub-scale grid with 4 times finer resolution is presented in [113]. We propose a method dedicated to evaporation, for which the conductivity is pre-computed on the original mesh.

When RE is used to model strong gradients with the FE method, the physical property values often spread over several orders of magnitude within the same element. Those large disparities impose unbalanced fluxes for nodes inside a same linear element: this flux is $\mathbf{J} \simeq K\nabla\psi$ with $\nabla\psi$ constant (since we use first-order elements) and K varying over several order of magnitude.

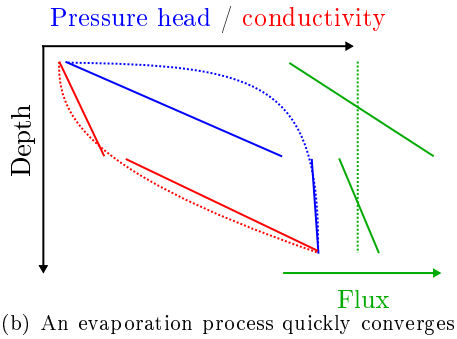
In the case of infiltration fronts, that disparity does not influence much the solution as advection is driven by upwind properties, which are always dominating ($K \simeq \exp(-\psi)$, see Fig. 5.1(a)). Also, the water content is expected to quickly increase, so imbalance inside the element is expected. On the other hand, during an evaporation process the flux along the soil profile tends to

be constant or progressive, resulting in only small changes in the water content. This is not reflected in the product $K\nabla\psi$ (see Fig. 5.1(b)). Moreover, in transient evaporation, the flux is stronger at the outlet, but the discretized situation shows a flux gradient in the other direction. To compensate, a DG FE solution adapts itself with large jumps between elements and in extreme cases the model can crash.

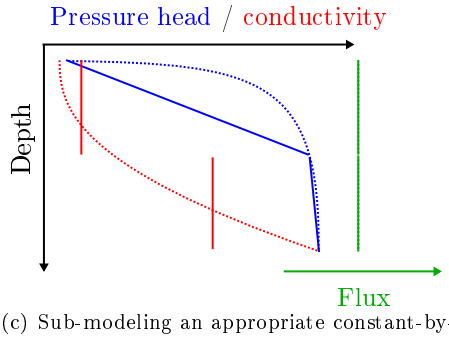
The proposed solution is to apply a constant-per-element diffusivity K_{cst} when strong gradients occur in an evaporation process. The value of K_{cst} is computed through the Kirchhoff transform with the assumption of a constant-per-element flux J_{cst}. The flux is then optimal for steady evaporation (see Fig. 5.1(c)). Note



(a) An infiltration front is often poorly resolved inside linear elements, especially at beginning. An appropriate spatial discretization is difficult to find as the front is moving.



(b) An evaporation process quickly converges to a constant flux ($K\nabla\psi = \text{cst}$). A linear numerical solution cannot approximate properly this state. In a classical transient state the flux gradient should decrease with depth.



(c) Sub-modeling an appropriate constant-by-element conductivity can solve that issue.

Figure 5.1: Mathematical (dotted lines) and numerical (continuous lines) typical profiles for infiltration (left) and evaporation (right). K is linearized on each element.

that a transient evaporation naturally has a stronger flux at the surface but the flux differences inside an element are small. Overall, it is better to produce a constant flux than a progressive flux varying in the wrong direction.

We consider a constant flux J_{cst} along a rectilinear path aligned to this flux (in 1, 2 or 3D). The Kirchoff transform is a simple reformulation of the 1D Darcy flux to obtain the distance between two points with the negative pressures ψ_1 and ψ_2, placed on the path of a constant and homogeneous flux J_{cst}:

$$\text{Kirchhoff}(\psi_1, \psi_2, J_{cst}) = -\int_{\psi_1}^{\psi_2} d\psi \left(\frac{J_{cst}}{K(\psi)} - 1 \right)^{-1} \tag{5.26}$$

This flux J_{cst} is iteratively found from the two extreme solution values in the element (ψ_{min} and ψ_{max}), and the distance l_e between those values[2]. It should be noted that the Kirchoff function must be slightly modified if an non-vertical axis is considered. A Newton solver is used on a log scale to ensure a quick convergence. The peudo-code reads:

$$\text{while } |\exp(e) - 1| \geq \text{convergence criterion :} \tag{5.27}$$
$$f_\varepsilon = \text{Kirchhoff}(\psi_1, \psi_2, \exp(\log(J_{cst}) + \varepsilon)) \tag{5.28}$$
$$\Delta f = (\log(f_\varepsilon) - \log(\Delta z))/\varepsilon \tag{5.29}$$
$$J_{cst} := \exp(\log(J_{cst}) - e/\Delta f) \tag{5.30}$$
$$\Delta z = \text{Kirchhoff}(\psi_{min}, \psi_{max}, J_{cst}) \tag{5.31}$$
$$e = \log(\Delta z) - \log(l_e) \tag{5.32}$$

where ε is used to numerically compute the function slope ($\varepsilon = 10^{-8}$ is often used with double precision). J_{cst} is initialized with a guess of the final value, determining also the initial values for Δz and the error e. Once J_{cst} converged, the conductivity[3] is found with $K_{cst} = J_{cst}/(\partial\psi/\partial z - 1)$.

A constant-per-element capillary capacity C_{cst} value is applied to be consistent with K_{cst}. It is set such as if $K_{cst} = K(\psi^*)$ then $C_{cst} = C(\psi^*)$. ψ^* is found with another log-scale Newton solver. The algorithm to find ψ^* is similar to Eqs. (5.27)-(5.32) with l_e replaced by K_{cst}, the Kirchhoff function replaced by $K(\psi)$ and J_{cst} replaced with ψ^*.

The choice of ε in this second solver is quite important. Indeed, as shown in Fig. 5.2, the classical value of 10^{-8} shows numerical instabilities in the slope computation. The Newton solver can then easily diverge from the solution. In our cases, a value of $\varepsilon = 10^{-2}$ has been used for C_{cst}. In future models, it is recommended to use a polynomial approximation of the function K to avoid such numerical instabilities, as $log(K(\psi))$ is nearly linear.

[2] For a linear 1D solution, the values for ψ are the nodal values and l_e is the length of the element

[3] in 1D, in the vertical direction

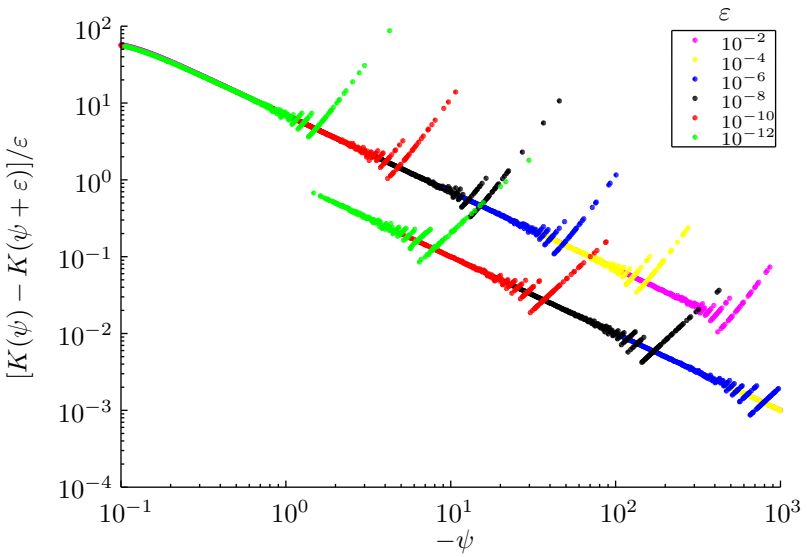


Figure 5.2: Influence of the choice of ε in the numerical evaluation of the Van Genuchten conductivity function derivative. The jumps are due to the application of a difference of very close values, highlighting the double precision numerical granularity in the evaluation of K. Mathematically, $dK/d\psi$ is continuous.

The trigger to activate this algorithm is the presence of a strong gradient for ψ and a decrease of the mean value of θ in the element over time, which characterizes evaporation or other outflow process.

5.5 Model validation and performances assessment

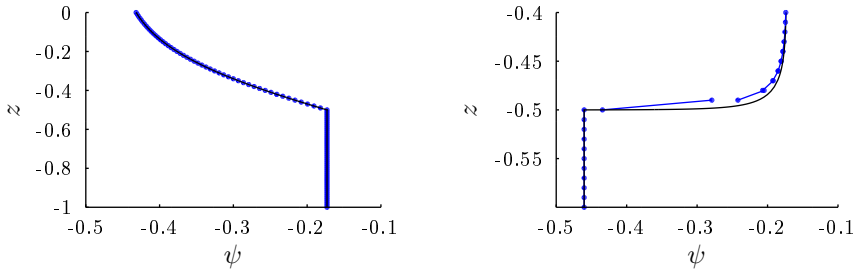
The test case have been split in three groups. The first is about model validation with a set of 1D benchmarks that have an analytical solution [114]. The second deals with 3D simulation of a ditch infiltration experiment [1, 115] with two types of finite elements. The third is centered on the parallel behavior of the model with strong and weak scaling test cases.

5.5.1 1D unsaturated analytical benchmarks

Vanderborght et al. [114] have gathered a set of analytical solutions for RE and the tracer transport equation. In this section, simulation results are produced for all their pure RE test cases. They deal only with the unsaturated

Table 5.1: Unsaturated analytical benchmarks dynamics

	Infiltration	**Evaporation**
Steady	1a, 1b, 1c	2
Transient	3a, 3b, 3c	4a, 4b, 4c, 4d



(a) 1a: infiltration, loam over sand

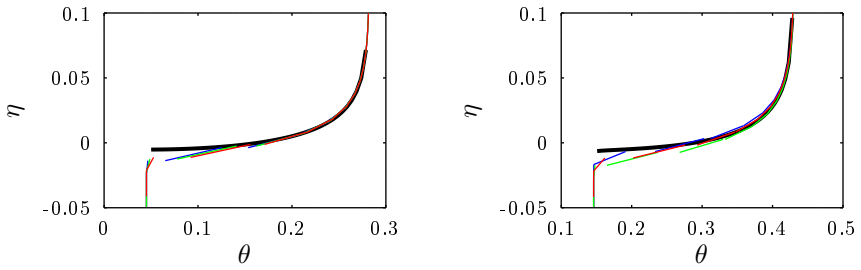
(b) 1b: infiltration, sand over loam

(c) 1c: infiltration, clay over sand

(d) 2: evaporation, loam

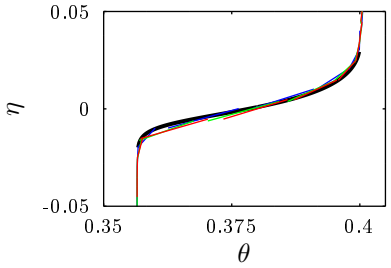
Figure 5.3: Steady infiltration and evaporation test cases from [114]. The analytical result is displayed in solid black lines while the results are in blue segments enclosed by dots. Overall those test cases see a very good fit. The discontinuity of the DG FE scheme is visible when the spatial resolution is not sufficient to handle the problem.

dynamics. The tests are identified with a number for the dynamics (gathered in Table 5.1) and a letter for different parametrizations. A summary of the different simulations is given in Appendix A.6 with the soil properties used through all simulations (Table A.1), the detailed parametrization for each simulation (Table A.2) and their initial/boundary conditions (Table A.3). The details about those simulations and their analytical solutions can be found in [114].



(a) 3a: sand at 0.1 0.2 and 0.3 day

(b) 3b: loam at 0.2, 0.5 and 1 day

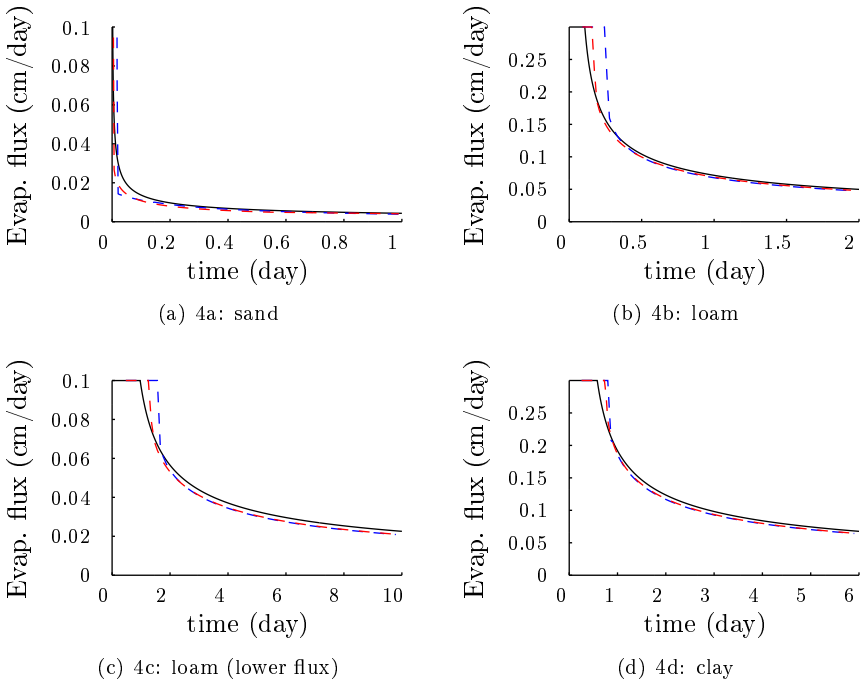


(c) 3c: clay at 0.1, 0.2 and 0.5 day

Figure 5.4: Transient infiltration test cases. The variable η is the relative analytical position of the infiltration front, used to compare the fronts at 3 different times. The black lines are the analytical solutions while the color segments are the model results, respectively in blue, green and red. Although some inevitable numeric diffusion is visible on numerical results, it is comparable to another models for the same discretization.

The results for all simulations (Figs. 5.3, 5.4 and 5.5) show good agreement with analytical solutions. When the spatial discretization is not sufficient to approximate sharp variations, jumps appear between the DG finite elements (like in Figs. 5.3(b), 5.4(a), 5.4(b) and 5.4(c)). That information could be used by an adaptive mesh method, or to refine the mesh according to the physics to improve precision. It should be noted that the tests are given with a fixed mesh resolution to allow model inter-comparison.

Due to its soil properties, the test case 3c is numerically unstable. Indeed, the slope of the conductivity function is very important and numerically discontinuous when $\psi \to 0$. This extreme slope and its discontinuity make the convergence towards a stable solution very complex for any numerical method, especially in the case where the boundary condition $\psi = 0$ is applied. As an



(a) 4a: sand

(b) 4b: loam

(c) 4c: loam (lower flux)

(d) 4d: clay

Figure 5.5: Transient evaporation test cases. The black lines represent the analytical results, while dashed lines the numerical ones, with a spatial resolution of 1 cm in blue and 0.25 cm in red. The coefficient of determination (R^2; see [114]) for the evaporation simulations in red are respectively of 0.639, 0.969, 0.978 and 0.986. The evaporation fluxes are quite sensitive to the spatial resolution. The presented model shows good agreement with analytical solutions considering the numerical challenge of the test 4a.

alternative, we have applied a 3rd order polynomial approximation on the Van Genuchten functions to obtain the result displayed in Fig. 5.4(c).

The transient evaporation test cases produces very strong gradients at the surface boundary. The conductivity and capillary capacity values at nodes inside the same element can be orders of magnitude different. To improve the precision, we change the parametric function for those test cases. The idea is to use the well-known Kirchhoff transform to deduce the exact flux (considered constant through the element) then transform K and C accordingly to reflect that flux. That sub-modeling process is fully described in section 5.4.

5.5.2 Ditch infiltration test case

In [1, 115], the authors measure and model an infiltration experiment from a ditch. This test case is particularly interesting because of the in-situ measurements available in all three dimensions, allowing a validation of 3D model results with observations. Indeed, in practice most 3D models are validated with respect to other models or sparse measurements [116].

We first present a 1D simulation under the ditch, along with the measures (which should show a lower infiltration rate) and Hydrus1D as a reference model. Both 1D and 3D meshes are used to show the influence of the 3D discretization on the result. Eventually, the full 3D simulation with the original soil measurements (given in Table 5.2) is compared with the measures and discussed.

Table 5.2: Soil properties for the ditch test case

Depths	α (m^{-1})	β	K (m/s)	θ_s	θ_r
0-0.9 m	13.6	1.296	1.90×10^{-4}	0.37	5.1×10^{-5}
0.9-2.2 & 3.5-6 m	2.4	1.572	1.03×10^{-5}	0.33	5.5×10^{-4}
2.2-4.5 m	5.2	2.279	6.75×10^{-5}	0.31	5.7×10^{-4}

1D simulation

As a preliminary study for the 3D simulation, a simulation is performed taking a 1D water column below the ditch. The geometry, boundary and initial conditions are identical to the experiment of 2002 [115]. However, this reduction of dimensionality changes the upper boundary condition into an homogeneous Dirichlet condition over the whole surface (*i.e.* an infinite ditch bottom without borders). The infiltration front is thus expected to be faster, as the water cannot be transported horizontally.

The simulation only lasts for 1.5h as reaching a fully saturated state is not the purpose here. To ensure a correct time resolution, the time step ranges from 0.001s at the start to 5s after 2min of simulation. The column is composed of 54 elements ranging from 0.05m at the top to 0.2m at the bottom.

First, we compare the results of a 1D simulation with a simulation of Hydrus1D as reference. The results are displayed in Fig. 5.6 and show a very good agreement between simulations.

Secondly, a fake 3D case, set-up to match the 1D case, is simulated to detect any effect of 3D elements. That 3D mesh is identical to the one used in the next section, removing all elements above the ditch depth to apply the homogeneous Dirichlet boundary condition at the same elevation. On the borders, a no-flux

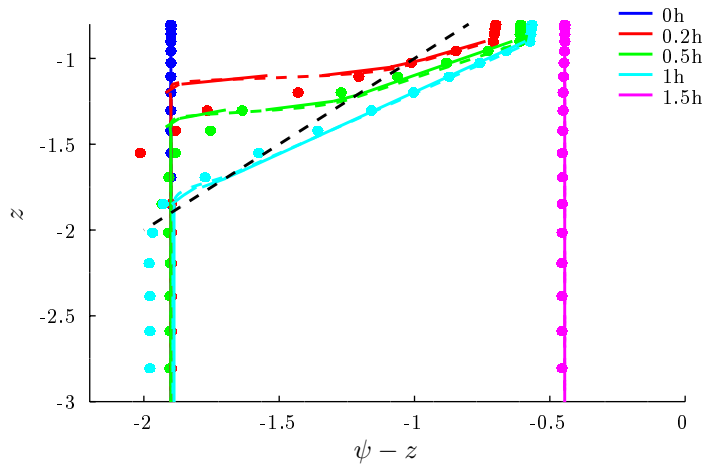


Figure 5.6: Comparison of 3 simulations at different times: Hydrus1D (dashed lines) and our model in 1D (solid lines) and in pseudo-3D (dotted lines). The values below the black dashed line are at saturation ($\psi > 0$). The Hydrus1D simulation is mostly hidden by the 1D simulation results. No side effect is observed due to the use of a 3D mesh. It is seen that the model matches quite well.

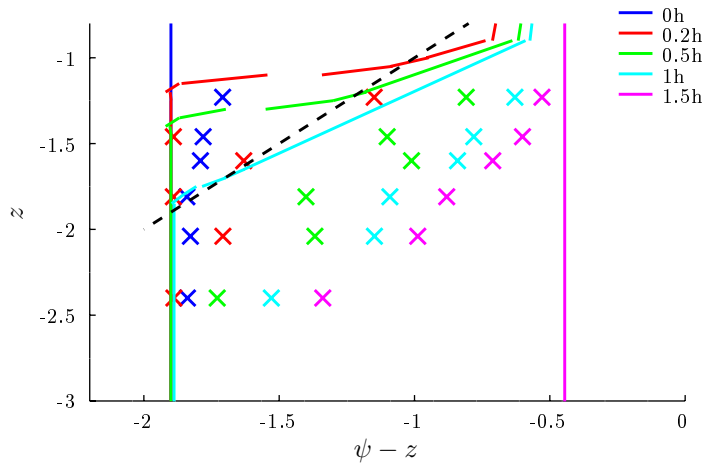


Figure 5.7: Comparison of the 1D simulation (continuous lines) with measurements ($\times$). The values below the black dashed line are at saturation ($\psi > 0$). Measures show an unexpected behavior: the measured 3D infiltration bulb is faster than the 'plane' infiltration simulated by the 1D model, for with the flux is not dispersed horizontally and therefore should go down faster.

boundary condition is applied. It can be seen on Fig. 5.6 that the 3D simulation, with a coarser mesh resolution in the z direction, is slightly more diffusive and exhibits undershoots at infiltration front. The 3D physics matches the reference on unstructured and multi-scale meshes.

As already mentioned, the measurements right under the ditch should show a slower infiltration front. However, it can be seen in Fig. 5.7 that it is much faster. The full saturation is reached just after 1h of simulation and between 1/4 and 1/2h in the tensiometer measurements. The piezometer measurements (not displayed here) show a quasi-instantaneous response to the water surge.

This large discrepancy is probably due to preferential flows reaching the water table by bypassing the unsaturated area below the ditch. A strong work of calibration/validation was done in Dages et al. [1]. Several numerical tests to reach the measurements were done by tuning the water conductivity. The main change in all their scenarios is a 3 to 5.2 fold increase of K for the layer below the ditch, reducing the latency before filling the water table. The authors mention that it was not possible to find an adequate set of soil hydraulic properties for simulating precisely both the piezometric heads and the hydraulic gradients (*i.e.* the tensiometer results). This is an additional argument in favor of preferential flows as the water table and the unsaturated areas seem decoupled. This hypothesis is supported by other studies on ditches from the same area, presenting large differences between soils in ditches and in their surrounding. It is shown that ditch soils present higher conductivity and a stronger concentration of preferential paths [117, 118].

As our model cannot represent preferential flows, the calibration of the 3D model for this test case was left aside.

Full 3D simulation

The 2002 experiment of [115] is simulated with the measured soil parameters. As discussed in the last section, no additional calibration is performed. The initial and boundary conditions are similar to those in [1]: no-flux for all the surfaces except the vertical ones far from the ditch (for which a Dirichlet BC is applied with the initial values for ψ) and the active part of the ditch. There is a small difference on the application of the pressure head on the surfaces of the ditch. In SWMS_3D a switch is done between a Dirichet BC with the water height above the node and a no-flux BC when the node is dry [119]. In our model the same switch is applied at the element level.

The geometry shown in Fig. 5.8 is approximated with a 3D mesh composed of 4086 tetrahedra in the upper layer and 12,615 triangular prisms in the lower, for a total of 92,034 DOF. The vertical prisms are well-designed for water tables as they fit well classical profiles (like low-slope water table and infiltration

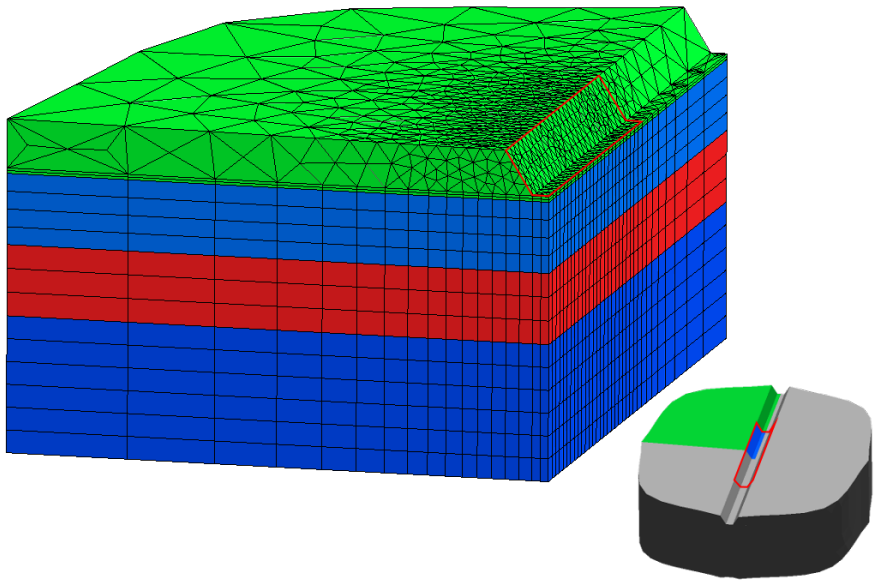


Figure 5.8: 3D mesh of the ditch composed of tetrahedra and triangular prisms, with a color for each soil type. The two visible vertical faces are symmetry planes of the experimental setup, as shown in the small overview. The active area of the ditch is only the first 5/12th of the total length, delineated in red in both views.

fronts). The prisms can easily fill any complex geometry and do not present a preferential direction for gradients, which is convenient for a water bulb. The mesh resolution is fine near the ditch and coarse further away from it to reduce the computational cost.

The 8-hours simulation is discretized into time steps of 1s before $t = 360$s and of 5s afterwards. Greater time steps are stable but lead to greater overfilling above the saturated water content. As we have not implemented a correction algorithm for those artifacts, we limit them with a smaller time step experimentally chosen.

Results are shown in Figs. 5.9 and 5.10. For visualization purpose, a L_2 projection to continuous P_1 elements has been applied to the DG results. It can be seen that the model correctly represents the key physical processes. An infiltration bulb develops around the ditch during the first hour, reaching the water table under the ditch center after about 1.5h. Then the saturated water column pushes the water table away, creating a visible piston effect in (d) and (e): the pressure gradient partially points upwards, bringing water to unsat-

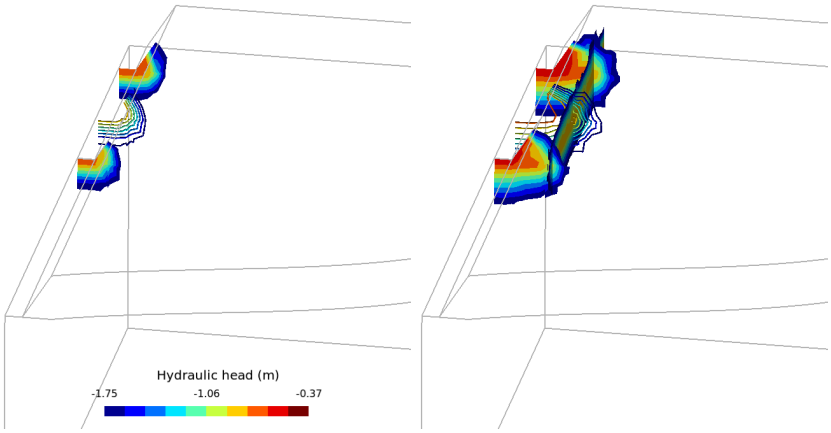
urated areas bypassing the infiltration bulb. Eventually the infiltration stops and the system slowly comes back to equilibrium in (f).

As expected after the assessment of the 1D results, the simulation values are far from the experimental data. In Fig. 5.11 the tensiometers values are compared with the hydraulic head (*i.e.* $\psi - z$) at 2.8m depth, right under the ditch and at two other points on a perpendicular axis from it. The infiltration profiles are coherent with the previous 1D simulation and show very similar results as the 'a-priori' scenario of [1]. However, the front seems to reach the water table a bit earlier and, as a consequence, their simulation have more positive hydraulic head values around 4-5h, going further than the measurements. The model values are closer to measurements at the end of the simulation.

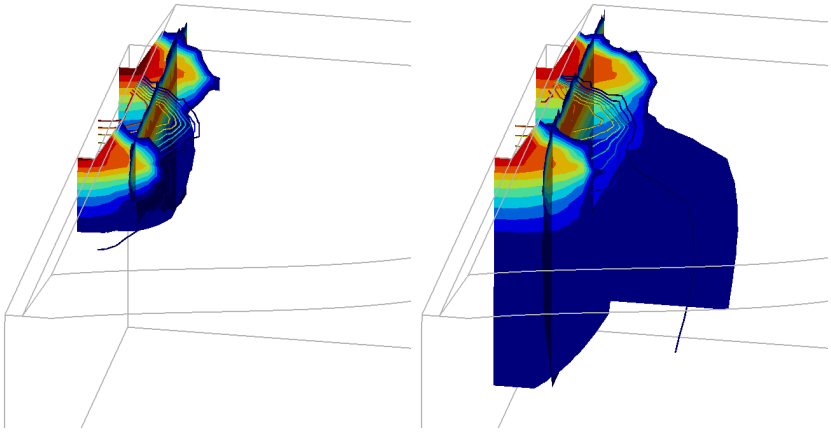
The time evolution of the piezometer values, shown in Fig. 5.12, are again highly similar to the 'a-priori' scenario, with a slightly longer delay before reaching the water table. As mentioned before, the very quick reaction of the water table is probably induced through preferential flows. The piezometer values along the ditch distance in Fig. 5.13 are again far from the model values. At the end of the simulation the water level tends to be homogeneously spread. This suggests that the computational domain should be extended, or that the use of a free-BC condition on the far vertical planes should be more appropriate than a Dirichlet one. A free-BC consists in imposing a slope to the variable instead than a fixed value.

The small differences with the 'a-priori' test case of [1] can be explained by three factors. First the mesh is fully composed of tetrahedra, which are less adapted to vertical infiltration fronts and show large vertical extent just under the ditch. The second difference is the SWMS_3D model which uses strong Dirichlet boundary conditions. When the boundary conditions are applied, the boundary nodes are instantaneously set at the given value. As a result, the elements linked to those nodes instantaneously change their water content, at the concurrence of roughly $\frac{3}{4}$ of their volume for tetrahedra with one face on the ditch. In contrast, the weak Dirichlet boundary conditions applied in our model constrains the flow much more slowly, allowing the water to flow inside the element proportionally to the soil conductivity. The third difference is the averaging of the soil properties of SWMS_3D, which have a non-negligible effect when a small layer of very conductive soil is present just under the ditch.

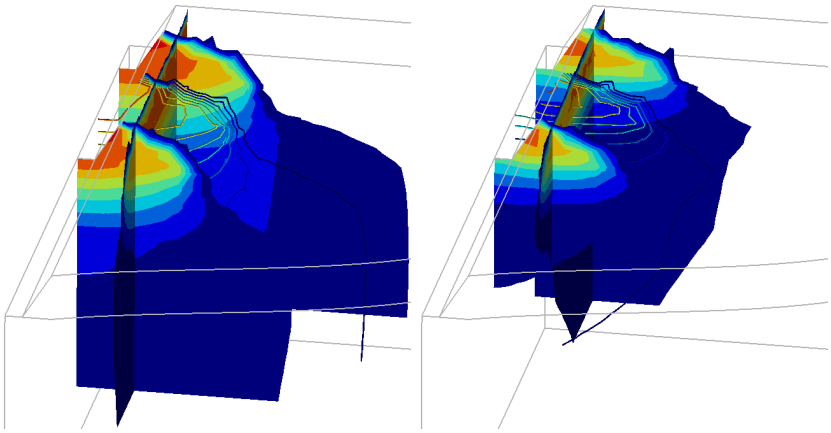
To conclude this test case, we shall mention again that building a numerical model with good properties is very challenging. Once the model is implemented, the next important task is to parametrize it correctly. In the absence of a preferential flow module in like dual porosity [120, 121] or fracture simulation [122, 123], parametrizing our model is very challenging as it misses a key physical process. We thus prefer to leave this test case as such, with a correct qualitative dynamics results but an incorrect quantitative comparisons.



(a) 0.2h (b) 1h



(c) 1.5h (d) 2.5h



(e) 4.5h (f) 8h

Figure 5.9: Evolution of the ditch simulation infiltration bulb displayed on four planes ($x = \{-1, -3, -5\}$ and $y = -1$). The hydraulic head is represented by iso-values. The middle plane is kept transparent for visualization purpose. In (a) and (b) a pure infiltration bulb emerges. The infiltration front reaches the water table at the ditch center in (c), then reaches it everywhere under the ditch in (d). In (e) the flooding stops and the pressure head bulb is at its maximum extent. At the end of the simulation in (f) the system slowly comes back to equilibrium.

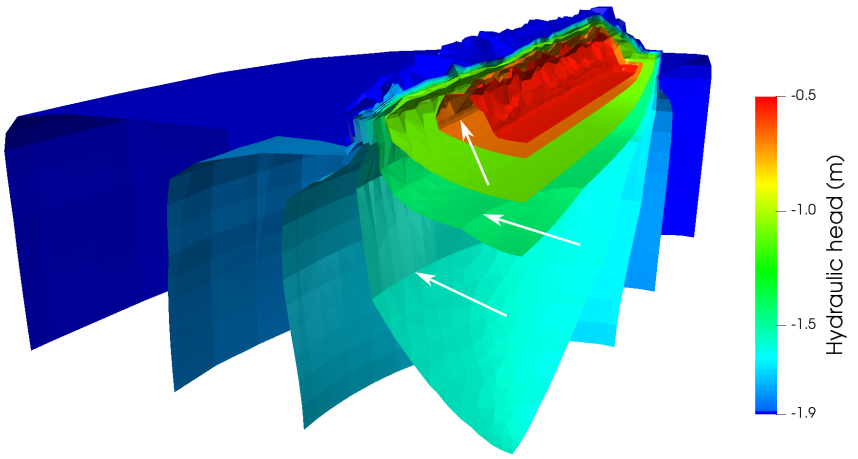


Figure 5.10: Iso-values of the hydraulic head from the ditch simulation at 3.6h. The abrupt changes of slope pointed by arrows are due to a change of soil type. The upper tetrahedral part of the mesh produces noisy values as the infiltration front is only discretized with 1 or 2 elements in depth.

5.5.3 Model efficiency

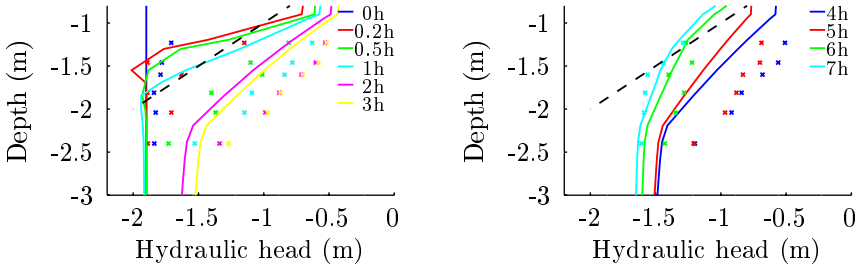
A model efficiency is determined by its precision, its robustness, and its execution speed. This section is dedicated to the last point. The model computation time on a parallel infrastructure is:

$$t_{tot} = \frac{t_{it} \times n_{it}}{n_w \times p_e}, \tag{5.33}$$

with t_{it} the computational time of one iteration if the model is run sequentially with one worker, n_{it} the number of iteration (depending on the time step), n_w the number of workers and p_e the parallel efficiency (itself dependent of n_w). All those variables are discussed in this section, the first two variables without quantitative results as it is not the focus of the present work, the last two with illustration of strong and weak scalings.

The High Performance Computing infrastructure used to produce those results is the NIC4 cluster, part of the CECI consortium[4]. When we performed those tests, in January 2018, it had the following characteristics: 2048 CPU at 2.0 GHz with each 4GB of RAM, QDR Infiniband.

[4]CECI webpage: `http://www.ceci-hpc.be`



(a) Infiltration (b) Return to equilibrium

Figure 5.11: Tensiometer profiles below the ditch, with model values (solid lines) and measurements (x). Each color represents a point in time. The values below the black dashed line are at saturation ($\psi > 0$).

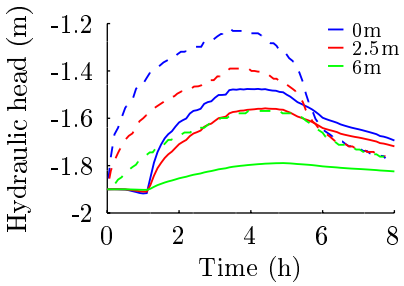
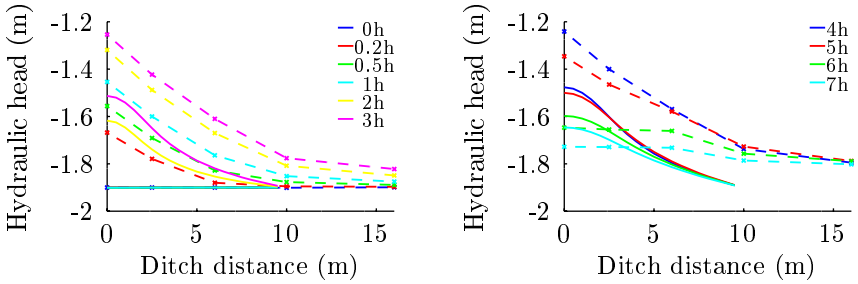


Figure 5.12: Piezometer values profiles in time at different distances from the ditch. Model values are in continuous lines and measurements in dotted lines.

Iteration cost

Our model is compared with classical implicit models. The computational time of one iteration is increased by the corrector, the slope limiter and the synchronization between the two variables. It is decreased by the use of a single iteration in the non-linear solver. That single iteration is sufficient to solve the non-linear problem as we are taking C and $\mathbf{K}$ from the last time step, linearizing the equations. The solution is close to an explicit one, except that the elliptic part of the equation (*i.e.* where $C \simeq 0$) can be solved. The computational cost of that single non-linear iteration can be further decreased by truncating the linear solver convergence, but only with caution as that degrades the precision.

The corrector procedure formally consists in applying an explicit time step. However, with the identical right-hand sides of Eq. (5.23) and Eq. (5.22) and



(a) Infiltration (b) Return to equilibrium

Figure 5.13: Piezometer values along the horizontal axis. The water table level rise is less important in the simulation (solid lines) than in the measurements (dashed lines), indicating a lower global amount of infiltrated water. The water receding is more important in the simulation, due to its boundary condition at 9m from the ditch.

the mass-lumping diagonalizing the mass matrix, this corrector step can be reduced to a simple vector operation without communication between nodes (see Appendix A.7 for more details). However, as we are using a general purpose framework, this optimization has not been implemented in this work.

Explicit constraint on the time step

The procedure described in 5.3.1 lead to a mixed formulation formed by an implicit predictor and an explicit corrector. What distinguishes it from other predictor-corrector models [30, 16, 41, 111] is the use of exactly the same right-hand side (*i.e.* the divergence of the flux) in both equations. This difference improves the stability of the time integration scheme.

If mass balance is reached in Eq. (5.22), *i.e.* if $C = (\theta^{n+1} - \theta^n)/(\psi^{n+1} - \psi^n)$, the predictor is mathematically identical to the corrector, and the so-called explicit time step gives the same result as if it was implicit. In that ideal case, our procedure achieves 'unconditional' stability like other implicit time integration schemes.

However, that ideal case is generally not reached because C has to be approximated (see details in section 2.4.3). In our formulation C is evaluated at ψ^n, which is of the same order of precision as if it was evaluated at ψ^{n+1}, but is not closer to mass conservation. The purpose of the corrector is to balance it.

In practice, we have observed that with our procedure that the time step can be chosen out of the bounds of the classical CFD constraints for advection-

diffusion problems, which is determined by the smallest element in the mesh. To support this observation, it is however not possible to produce the classical theoretical demonstration – a convergence analysis – for the whole algorithm. In Appendix A.5, we propose a simplified von Neumann analysis. Its validity can be discussed as we neglect the steps described by Eqs. (5.24) and (5.25), and because such an analysis is derived from linear differential equation theory, while RE is highly non-linear. It should be noted that the last point is assumed by every implicit models of the RE. Yet the von Neumann analysis gives a hint on the algorithm objective, which is to avoid or greatly reduce the explicit time step constraint for solution stability.

Adaptive time stepping indicator

The time step in our model is user-defined. However, it is well known that time scales of subsurface hydrology vary over several orders of magnitude, *e.g.* infiltration front vs. evaporation. A model cannot be used on a long-term time frame without an adaptive time stepping, otherwise the majority of the computation time is spent on steady fluxes. Here we propose an indicator of the mass error of the predictor involving mass excesses on the corrector. This indicator can be used to adapt the time step along with other more classical indicators such as the non-linear solver error, the number of sub-iterations of the linear solver, temporal error, etc. This indicator is obtained by comparing the ideal case (fully converged predictor) and the real iteration. As it has been seen that a fully converged ψ-form is equivalent to the chord-slope approximation (see Eq. (2.19)), we have:

$$\text{Ideal:} \qquad \frac{\theta^{n+1}-\theta^{n}}{\psi^{n+1}-\psi^{n}}\frac{\psi^{n+1}-\psi^{n}}{\Delta t} = \frac{\theta^{n+1}-\theta^{n}}{\Delta t} = \nabla\cdot\mathbf{J}, \tag{5.34}$$

$$\text{Actual:} \qquad C(\psi^{n})\frac{\psi^{n+1}-\psi^{n}}{\Delta t} = \nabla\cdot\mathbf{J}, \tag{5.35}$$

$$\epsilon_m = C(\psi^{n}) - \frac{\theta^{n+1}-\theta^{n}}{\psi^{n+1}-\psi^{n}}, \tag{5.36}$$

$$= \left.\frac{\partial\theta}{\partial\psi}\right|^{n} - \frac{\theta^{n+1}-\theta^{n}}{\psi^{n+1}-\psi^{n}}, \tag{5.37}$$

with ϵ_m the error indicator. That error is transformed by the corrector into actual fluxes. Indeed, Eq. (5.34) is the ideal case for the ψ-form, but also the corrector of our method. At the saturated/unsaturated interface those additional fluxes can fill elements above saturation. It has been observed that the high values of this indicator are very local, often highlighting some nodes in infiltration fronts. We therefore suggest to use a L_∞ norm in order to take the

worst scenario into account when adapting the time step. A classical time stepping adapter using heuristics like the number of the non-linear/linear iterations or computing temporal truncation errors can include the minimization of such an error indicator in its search for an optimal time step. Such an algorithm is not developed in this document.

Implementation

Our models are based on the FE framework used by the model SLIM[5]. It implements matrix assembly and non-linear solvers, handles the mesh generation with GMSH, partitioning with METIS and can connect any linear solver from PETSc. It aims to maximize the number of matrix operations (very efficient with the BLAS library), can handle high order CG/DG discretization for many elements types and is optimized for parallel computation.

The core of the SLIM code is written in C++ but the users parametrize it through python scripts (the code is wrapped by SWIG). The script for the 3D ditch test case is given as example in Appendix A.8.

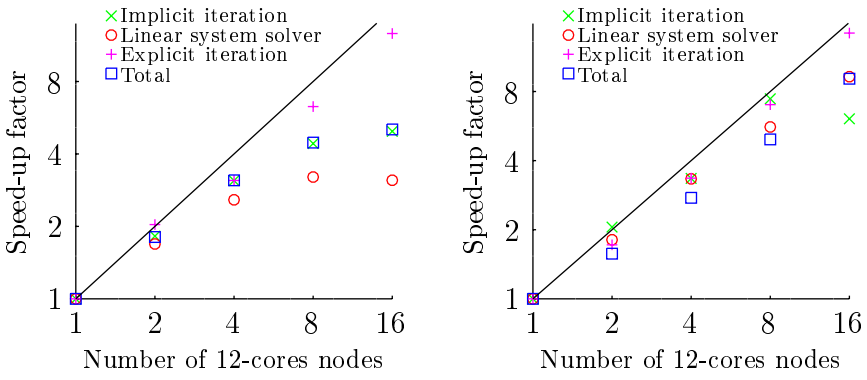
The integration of the our models in the SLIM code is quite efficient. Reducing the generality of the framework can lead to interesting optimizations, like a straight linearization of K and C or specializing the code for first-order elements. Such optimizations are interesting to reduce the matrix assembly cost, which can be substantial for explicit models. However in this case the limiting factor is the linear solver for the elliptic part of the equation, and emphasis should be put on improving that aspect.

Strong scaling

The previous 3D ditch infiltration test case is used with 3 different meshes generated by GMSH. The smallest one is similar to the one used in the previous section (92,034 DOFs), the medium and largest respectively have 429,440 and 3,408,000 DOFs. The partitioning between different workers is done with the METIS library which optimizes the volume-surface ratio of each partition while keeping a similar number of elements per worker. The scaling test has been done on a integer number of nodes of 12 CPU (see 2.6.1 for more details), up to 196 CPUs, with one worker per CPU.

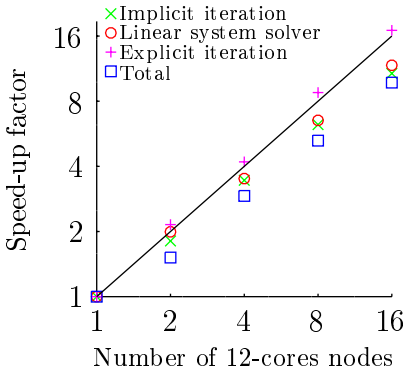
As expected for the strong scaling test case, a clear effect of the number of DOF per worker is visible in Fig. 5.14. Indeed, decreasing the number of DOF in the volume and increasing it at the partition boundaries decreases the computation over communication ratio, which is at the heart of a good scalability. In our case, the performance drops under 5000 DOF per worker (which corresponds

[5] https://www.slim-ocean.be



(a) Small mesh

(b) Medium mesh



(c) Large mesh

Figure 5.14: Strong scaling efficiency results for meshes with 9.2×10^4 DOF (small), 4.3×10^5 DOF (medium) and 3.4×10^6 DOF (large). The total computational time includes the time spent in explicit and implicit iterations. The Implicit iterations time includes the time spent by the linear system solver.

to about 1000 elements). The scaling jump between 1 and 2 nodes with the large mesh is likely due to the addition of the network latency. After 2 nodes our model shows an almost perfect scaling.

The total computational time has been split into each of the model components to assess the scaling of each part separately. As expected, the explicit part of the model scales very well [6]. The limiter and ψ update are negligible in the computational time and scale as well. The implicit solution is the most computationally expensive part of the model. It scales quite well until it reaches

the critical amount of DOF per worker. As it is already dominating, its scaling degradation is quickly reflected on the whole model.

Inside the implicit part of the model, the drop in parallel efficiency is concentrated around the linear solver, as it is often observed in other models [57, 69, 55]. We are using the GMRES generic solver from the PETSc library, preconditioned with a block Jacobi method on ILU sub-preconditioner. This is one of the most scalable generic methods as the preconditioner is fully local. However, it is certainly possible to further enhance the convergence or the scaling with more specialized methods. It should be interesting to assess the number of iterations of the GMRES linear solver in function of the number of workers. If it is increasing to a great extend, a better-suited but maybe costlier preconditioner should be considered.

The algebraic multigrid (AMG) method seems promising as it is particularly efficient in solving elliptic problems and does not require special geometries as with the classic multi-grid method [55]. Recent studies have shown that this method can be scalable on very large infrastructures [124, 52, 53]. The BoomerAMG solver from the Hypre library was tested in our model, but the results are not presented as the scaling was very poor. We believe that the scaling of our model could be further improved with the use of an appropriate AMG method (like AGMG [53]).

The simulations are run up to 5 times each. A overhead of nearly 25% on the running time is noticed between the best and the worst cases. That addtional time is equally spread between all the communicating parts of the model. We suggest that either some heterogeneity is present between the nodes (one slower node can impact the whole stack) or that the network was sometimes saturated by other runs. Therefore, it is likely that the model can perform better on a dedicated infrastructure.

We realized afterwards that the current partitioning treats both tetrahedra and prisms with equal weights. The scaling efficiency could thus be further improved by refining the algorithm to keep a similar number of DOFs instead than a similar number of element per worker.

Weak scaling

The weak scaling test case is described in section 4.4.4. Here, we replace the surface model by a Neumann surface boundary condition with a constant flux of 0.02 m/s. The first mesh is the same as in that previous section, with 62,500 hexahedra or 500,000 DOF per worker. In Fig. 5.15, we can see that the explicit part of the algorithm scales optimally when the implicit part is reduced to bring an overall result for the model of 96.6% parallel efficiency with 8 nodes (96 workers). The run with 192 workers can not be displayed because in the

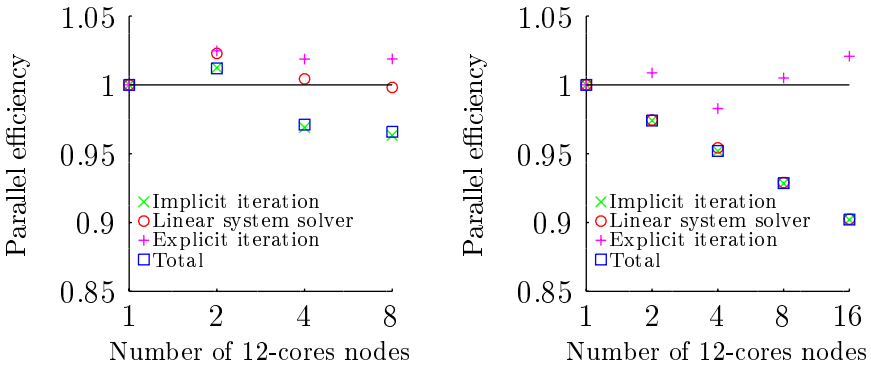
current implementation the whole mesh is read by each worker at initialization and saturates the node memory, hence stopping the model.

A second test case has been simulated with a smaller mesh, of half the size in the horizontal and with a space discretization 2.5 times larger in all directions. That brings a mesh of 1000 hexahedra or 8000 DOF per worker, just above the limit observed with the strong scaling test case. As expected, the results show a slightly degraded scaling, with a parallel efficiency of 90.2% for 16 nodes (192 workers).

This algorithm only uses very basic and computationally inexpensive arithmetics. In most cases, it will only need zero or one synchronization between nodes and then should scale well. A possible drawback of this algorithm is a possible repetition of the "bad luck" case in simulations of a strong rain event reaching a water table. However, such a case is difficult for any model.

5.6 Conclusion

We have presented a new model of RE that takes the pressure form as a predictor and the mixed form as a corrector. The predictor is solved with an implicit time integration scheme. The expected mass imbalance is then canceled with an explicit corrector. The predictor is linearized, which allows a one-step "nonlinear solution". The corrector uses exactly the same fluxes as the predictor, relaxing the explicit time step constraint. The corrector iteration can be re-



(a) Large mesh (500,000 DOFs/worker) (b) Small mesh (8000 DOFs/worker)

Figure 5.15: Weak scaling efficiency results. The vertical axis has been shrink to enhance the results visibility. The efficiency results for the total simulation are often hidden by implicit iteration results.

duced to a computationally inexpensive vector product. The model presents good strong and weak scalings up to 192 workers.

Our RE model is mass conservative to machine-precision. It is valid in both unsaturated and saturated areas. It produce accurate results, both against other models and analytical solutions. Its robustness is enhanced by not waiting for the convergence of a non-linear solver and by using slope limiters. It is fast with a quasi-explicit iteration (only one linear system solution per iteration), the vectorization of the corrector and the release of the time step constraint. Eventually, it shows a good scaling for both weak and strong test cases, with performances only degrading below 5000 DOF per worker, which is quite low.

The only known limitations are small mass excess that can appear at infiltration fronts and the possible desynchronization of θ and ϕ for draining saturated cases. Three post-process procedures to correct the former are suggested, one pragmatic solution to the latter is proposed.

This work can be at the beginning of a new line of RE models. Already a simple reimplementation of what is proposed, dedicated to the RE with discontinuous first-order elements can bring many improvements and optimizations, as the vectorization of the explicit iteration, a better linear solver or an adaptive time stepping.

A weakly scalable RE model can be used for instance to simulate in reasonable times a full-scale watershed, limiting the influence of the choice of adapted boundary conditions. A good strong scaling is appreciable to reduce the computational time of simulations at smaller scales, allowing the simulation over longer periods. With a more reasonable computing time, techniques of inverse modeling or other Monte-Carlo based parametrizations can be put into practice with appropriate resolution, even in 3D. A robust and perfectly mass-balanced model is appreciable to any groundwater modeler, and it makes its use easier for any newcomers.

CHAPTER 6

Conclusion

6.1 Summary

The long-term objective behind this work is developing a terrestrial water model able to interact with river, lakes, estuaries and sea. Most of the biogeochemical cycles linked to life are present in that land-sea continuum, and the main driver for all those cycles is water (see Fig. 6.1). The advantages of a continuous modeling is the possibility to study the interactions between each sub-model (river, groundwater, runoff, lake, etc.) and to avoid boundaries close to the domain of interest. Indeed, in a sub-model used alone strong assumptions and/or input data are necessary at the domain boundaries, which often strongly affect the results. If those boundaries could be moved further away by using larger domain or another sub-model (when the physics changes), assumptions and input data would have a reduced impact on the region of interest. To build such a continuous model, it is necessary to have working sub-models beforehand. However, we found that existing 3D Richards equation (RE) models were not efficient enough on large scales and often subject to convergence issues. They were especially not well designed for parallel computing infrastructures, although there have been some improvements over the last few years.

In this work, a first RE model is developed, perfectly scalable but with some caveats (Chapter 3). It is then coupled to a surface model (Chapter 4). Even-

Figure 6.1: Illustration of the water cycle. From the raindrop to the sea, the water cycle is home of important biogeochemical interactions, which can be impacted by human activity. The integration of all the terrestrial water components (river, lake, groundwater, etc.) within a unique numerical model allows for quantitative studies of those interactions.

tually, a second RE model is presented, which corrects the caveats of the first one by trading a little of its scalability (Chapter 5).

Strong design constraints were imposed on the first model: conservation, scaling and robustness. They drove the research towards an unexplored area of RE: a fully explicit integration time scheme. Such a scheme requires to make some approximations in saturated areas. The elliptic behavior is reduced to a very diffusive one, and the time step is limited by the explicit stability constraint. The resulting parallel scaling is perfect as the communication is limited between neighboring elements. The use of two variables (the pressure head and the water content) allows the solution of the RE in both saturated and unsaturated areas. Finally, mass is conserved to machine precision. The physical discontinuity of the water content between different soil layers is handled in a natural way by adapting the jump terms of the DG FE method. That discretization method is also well suited for the advection-dominant physics present in infiltration fronts and allows the use of standard techniques to easily modify the flux between elements or the values inside them, like slope limiters. The approximation in saturated area is reasonable as long as that area remains shallow, with a small

number of elements in the vertical direction. Under that condition, the model produces precise results.

The first model is then coupled to a surface runoff model to fully simulate intensive rain events that can generate surface flows. A 2D CV FE method is used for the NI approximation of the SWE. The zero-depth issue present in other methods is then bypassed, but the model can lead to unphysical fluxes when anisotropic elements are used. Modifying some coefficients to non-zero values avoids this issue efficiently and does not degrade the results for the presented test cases. The proposed coupling between the two sub-models is a simple extension of the DG FE method from the subsurface model. More precisely, the term of the equation for the inter-elements flux in the subsurface model is extended to the surface model. That term shares properties with two common methods: the first order exchange term (flux proportional to the difference of pressures) and a full coupling (equality between the pressures). It does not rely on an user-defined parameter like the former and is more flexible than the latter. A unique time step is used for both sub-models. The surface model have fast dynamics handled by an implicit integration time scheme, because any explicit constraint on the time step would be overly restrictive. On the other hand, the subsurface model, even on infiltration events, shows a much slower dynamics. An explicit time integration scheme is then used to improve the parallel efficiency. Overall, the minimal time steps for both sub-models are close. Therefore the global model is efficient without requiring a different time step for each sub-model, which brings more complexity. The parallel efficiency is only impacted by the surface model. The computational time for this sub-model represent a fraction of the total time in sequential run, but become dominant on a large number of workers.

Eventually, a second model of the RE is developed to mitigate some issues of the first model, while trying to preserve its good scaling, robustness and mass conservation. An implicit time integration time scheme predictor is used on the ψ-form of RE to handle its elliptic part. However this form is known to produce mass balance error. Therefore an explicit integration time scheme corrector based on the mixed-form of RE is used to ensure an exact mass balance. As the primary goal of the predictor is to solve the elliptic part which is linear, only one sub-iteration of the Newton non-linear solver is used. The non-linear functions are then evaluated from the last time step like in fully explicit time integration scheme, which has been proved to work well in the first model. As the corrector uses the same fluxes as the predictor, it is possible to compute the final result through a simple vector operation. The explicit stability is enhanced by the use of those implicitly-computed fluxes. At the end, that allows larger time steps. The principal caveat of this model is a small mass excess that can appear in strong infiltration fronts. Three post-processes are suggested to limit or remove it.

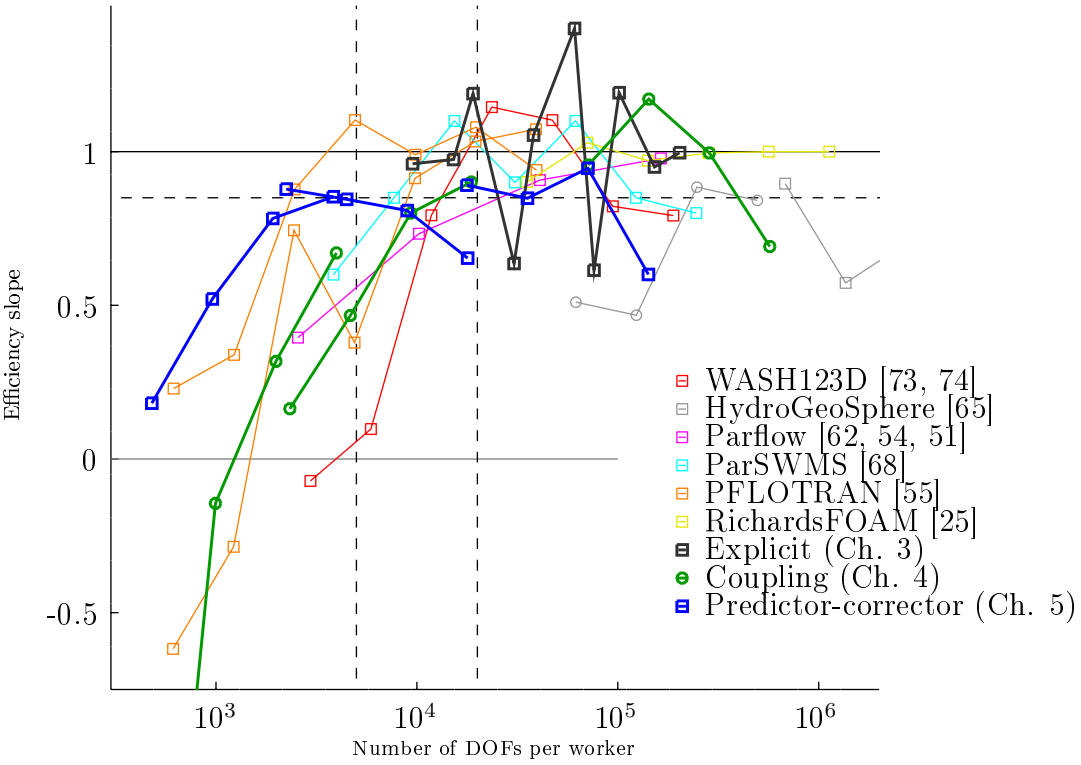


Figure 6.2: Strong scaling efficiency slopes for models of the Richards equation (squares) or models coupling surface and subsurface (circles). A slope of one represents a perfect scaling. It is expected for any model to see its performances strongly dropping under a given number of DOF/w. See Section 2.7 for more details.

A comparison of scaling efficiency (described by the efficiency slope as described in Section 2.7) is displayed in Fig. 6.2. It can be seen that the scaling breakpoint of our predictor-corrector model is below 5000 DOF/w, slightly outperforming other scalable models. Our coupled model, which does not have a real point of comparison for strong scaling, shows a breakpoint at 10,000 DOF/w. Despite a low maximal number of workers (196) compared to other studies (up to 262k), it shows promising scaling properties for much larger HPC infrastructures.

After iterating through a large range of model designs for the RE (more than the two presented here, *e.g.* see Appendix A.1), we end up with a promising model. It is efficient for sequential and parallel computations, precise, robust,

conservative. It thus presents almost all the properties expected by end-users. There is no need to check the mass balance, no non-linear convergence failure allowed, an almost unconditionally stable time integration scheme and a discretization optimized for modern HPC clusters. The current implementation shows a quite simple user interface written in Python (see Appendix A.8). We have shown that such a model for the soil can easily be coupled with a surface runoff model. It should also be the case for rivers, lakes, estuaries and seas, allowing to eventually achieve a model valid along the whole land-sea continuum.

6.2 Limitations and perspectives

Subsurface water flow models are often based on RE for its relative ease of use (apart from the usual numerical issues). However, RE is based on assumptions that can be constraining in specific cases. In unsaturated porous media, RE makes the assumption of water continuity. The mean vertical extent of such a continuity of filled pores has been estimated to be under 10m in natural unsaturated conditions [4]. It is thus questionable to solve the equation in larger discrete elements without the assumption of a change of physics. Some authors propose to keep the equations with upscaled parameters [57], other to use a finer discretization at large scale [125]. The retention curve is mostly used as a single-valued function in RE but in practice it can be multivalued due to the hysteresis effect. This approximation can lead to strong errors in specific wetting/drying regimes. Preferential paths, macropores or even soil fractures are not taken into account in RE. Additional elements like dual porosity or independent fracture models are usually used to handle those cases. The RE supposes a homogeneous fluid, but in practice is it often heterogeneous: there are density gradients due to salinity or temperature, trapped air, other fluid phases like oil, etc. In those cases, RE is no longer sufficient and it is necessary to switch to multiphase models (air, oil) or to a density-driven RE coupled with a tracer model (for salinity or temperature).

There are several methods available to parametrize RE such as inverse modeling and Monte Carlo methods, but they are often difficult to apply efficiently at large scale. Given the high variability of the soil properties, it is tempting to allow a large set of parameters during a parametrization. However, due to the sparse measurements, such parametrization with a large number of degrees of freedom can quickly over-fit the data, being only valid for a specific dataset or allowing the same precision with very different set of parameters. It is therefore recommended to fix their number reasonably, and to search for values in a range close to the property measurements. A good way to model soil heterogeneity at the macroscale has yet to be found, maybe with sub-scale modeling. Automated

detection of preferential flows to the model should be investigated as they can strongly influence such parametrization.

The current limitation of our second RE model is the small excess in mass that can appear when the constitutive relationships show a strong variability. In that model, the classical convergence and/or mass-conservation issue for RE is transformed into that small mass excess, which is kept instead of lost. In fact, to avoid mass loss, either an "exact" solution should be obtained from the non-linear solver or we accept such excess. The real precision of a fully converged solution is questionable considering the linearization of the highly variable functions over several order of magnitude inside some (linear) elements. For many applications, a small and localized excess of mass is acceptable. In that case, the excess can be handled either by doing nothing (as it will disappear when the area is unsaturated once again) or by spreading it with conservative fluxes. As the mass inside the domain is mostly driven by the BC, those mass excesses must end up somewhere and the most logical place is the nearest non-full element available.

Most of the time, model development for the RE focuses on the most challenging dynamics: infiltration fronts, often over initially dry conditions. This is numerically complex, but in long-term simulations it represents only a small part of the in-simulation time. Slower dynamics should be studied as well. The second RE model we derived should be tested in such principally diffusive conditions, where the time step is expected to be much larger. Evaporation can also lead to some issues when switching from elliptic to parabolic behavior. Likewise, the surface model possesses some interesting issues like steady water (nothing moves but the model is computationally costly) and it should be deactivated when no water is present at all. Eventually, a good adaptive time step algorithm should be chosen with the right balance between the resources use and the numerical error.

Water fluxes are important for many application, but what they transport is even more interesting. To study biogeochemical dynamics, the inclusion of a Lagrangian or Eulerian tracer model is required. The inclusion of either of them should not be more difficult than in other RE models and is let to further research. The DG FE properties of local conservation of the fluxes will prove very useful in this task.

Concerning the performances of the RE model, an optimized implementation should improve the sequential speed and the choice of a specialized linear solver should increase both sequential and parallel efficiency. A reduction of the computing time is appreciable for model inversion or other parametrizations, for large scale simulation (*e.g.* a whole watershed) and for long-term simulations (*e.g.* several years).

About the current research around RE, adaptive meshes are very promising for RE as the fastest physics is often very local and very transient. Second-order time integration schemes have sometime shown better results but, in our opinion, they are not not necessary as either the model stays nearly-steady either it shows non-polynomial discontinuous profiles in time. High order space integration schemes are interesting in non-advective areas where large elements are used. They could thus be linked with meshes adaptation in a *hp* approach (h for mesh resolution and p of approximation order).

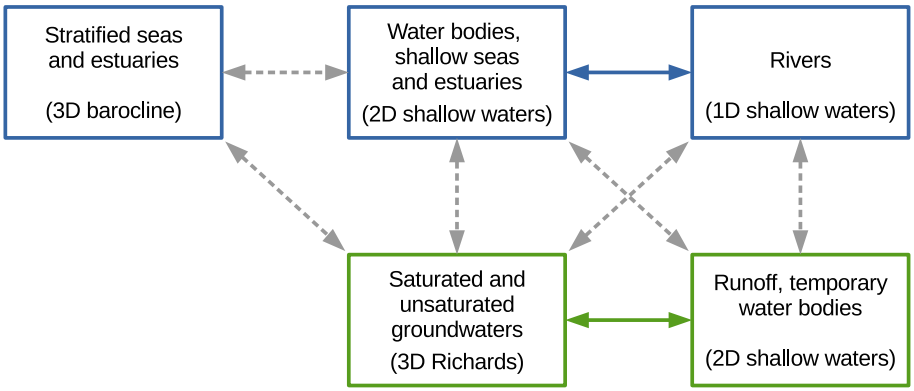


Figure 6.3: Models of the SLIM framework (boxes) and couplings (arrows). Existing couplings are shown with solid arrows while possible couplings to reach the land-sea continuum are depicted in dashed arrows. This document presents the green elements. The limits between the different surface models is not absolute, for instance a 3D model can be used for a deep lake or 2D flows for large rivers.

Guidelines to reach a model of the entire land-sea continuum are given in Appendix A.9. Even if the hydrodynamic models (river, estuary and sea in 1D-2D-3D, surface runoff, saturated and unsaturated subsurface flow) are now all available inside the SLIM framework (see Fig. 6.3), a lot of work is still necessary for environmental modeling. Tracers are mandatory for the bio-geochemical interactions (nutrients, pollutants, biological material, etc.) and couplings between each sub-models have to be designed to achieve both good physical exchanges and computing performances (coupling method, time steps, optional additional parametrization, etc.). The parametrization of the whole is challenging as some inter-model couplings have rarely been considered and because the set of input data should cover all the components. The measurement of properties at large scales like in-depth water content or flux exchange between entities should be investigated. Overall, all this remaining work should be viewed as a good challenge to assess the validity of the physical models, as

the main numerical difficulties often lie inside the hydrodynamic model components and could therefore be considered as mostly handled.

Applications of such a macro-model are just waiting: the nuclear leak in Japan, the effect of the sea level rise with salt water intrusion in estuaries and freshwater resources, long-range impact of the field fertilizers and pesticides, the carbon balance in trees/soils/seas, etc. Results of such simulations would not only be interesting for the environmental studies... but also for society as a whole!

APPENDIX A

A.1 RE as a system of equations

Many RE models rely on mixed FE methods. Overall, their general idea is to produce a continuous representation of the velocity field by adding it as a variable, and thus solving a system of two equations: one for the mass conservation equation and one for the Darcy flux. Here we try to use a system of equation for another purpose: keeping the mixed formulation intact, without the need to eventually express the problem with a unique variable, like in the modified Picard method. It simply reads:

$$\begin{cases} \dfrac{\partial \theta}{\partial t} = \nabla \cdot [\mathbf{K} \cdot \nabla(\psi - z)] \\ \theta = f_\theta(\psi) \end{cases} \tag{A.1}$$

with the same variables as described in the previous chapters. This formulation is fully mass-conservative and valid in both the saturated and unsaturated areas. The obvious negative points are the use of twice the number of DOFs compared to a classical problem and the natural discontinuity of θ between different soil properties.

This method was discarded after convergence issues with linear solvers. Fully saturated of fully unsaturated problems were easily solved, but the addition of one of those states in one domain fully dominated by the other state produced ill-conditioned matrices. This ill conditioning can be explained as follows: when

we apply a classical Newton non-linear method, it produces the following Jacobian:

$$\begin{pmatrix} M/\Delta t & L \\ M & \partial f_\theta \end{pmatrix} \tag{A.2}$$

with M the mass matrix, L the matrix representing the Darcy flux and ∂f_θ a diagonal matrix of the capillary capacity ($C = \partial\theta/\partial\psi$). The latter is zero (or near-zero) in saturated areas and very dry areas, and non-zero otherwise. It is observed that when the quantity of zero increases, the linear solver shows increasing difficulties. The precise mathematical reason was not found in this work.

A mitigation to this issue in dry areas is to use a variable transform as described in Section 2.5.2. Indeed, adapted to the soil properties, it transforms ∂f_θ into the identity matrix in unsaturated soils. It is however linked to the soil type, only allowing homogeneous soils, and does not solve the issue for saturated areas. As we do not found a solution, this development was discarded.

A.2 Modified Van Genuchten-Mualem

Following Vogel and Cislerova [88], the retention function and the conductivity functions are described as follows:

$$f_\theta(\psi) = \begin{cases} \theta_r + (\theta_s - \theta_r)(1 + |\alpha\psi|^\beta)^{-\nu}, & \psi < \psi_s \\ \theta_s, & \psi \geq \psi_s \end{cases} \tag{A.3}$$

$$\mathbf{K} = \begin{cases} \mathbf{K}_k \sqrt{\dfrac{\theta - \theta_r}{\theta_k - \theta_r}} \left(\dfrac{F(\theta_r) - F(\theta)}{F(\theta_r) - F(\theta_k)} \right)^2, & \psi \leq \psi_k \\ \mathbf{K}_k + \dfrac{\psi - \psi_k}{\psi_s - \psi_k}(\mathbf{K}_s - \mathbf{K}_k), & \psi_k < \psi < \psi_s \\ \mathbf{K}_s, & \psi \geq \psi_s \end{cases} \tag{A.4}$$

with

$$F(x) = \left[1 - \left(\frac{x - \theta_a}{\theta_m - \theta_a} \right)^{1/\nu} \right]^\nu \tag{A.5}$$

where θ_s, θ_r, α, β and ν are described in Section 3.2,$\theta_m, \theta_a, \psi_k, \psi_s, \mathbf{K}_k$ are additional empirical parameters and $\theta_s = f_\theta(\psi_s)$, $\theta_k = f_\theta(\psi_k)$.

A.3 Control volume finite elements derivation

Following the notations of section 4.3.2, the CVFE method modifies the following spatially discretized diffusive term [58, 18]:

$$\left\langle n\nabla\left(\sum_j u_j\phi_j\right)\cdot\nabla\phi_i\right\rangle \tag{A.6}$$

First, we recall that $\sum_i \phi_i = 1$ and therefore $\phi_i = 1 - \sum_{j\neq i}\phi_j$. If the gradient operator is applied $\nabla\phi_i = -\sum_{j\neq i}\nabla\phi_j$. Using this last relation, we obtain:

$$\begin{aligned}
\nabla\sum_j u_j\phi_j &= \sum_{j\neq i} u_j\nabla\phi_j + u_i\nabla\phi_i && \text{(A.7)}\\
&= \sum_{j\neq i} u_j\nabla\phi_j - u_i\sum_{j\neq i}\nabla\phi_j && \text{(A.8)}\\
&= \sum_{j\neq i}(u_j - u_i)\nabla\phi_j && \text{(A.9)}\\
&= \sum_{j}(u_j - u_i)\nabla\phi_j && \text{(A.10)}
\end{aligned}$$

By inserting this relation inside Eq. (A.6):

$$\begin{aligned}
\left\langle n\nabla\left(\sum_j u_j\phi_j\right)\cdot\nabla\phi_i\right\rangle &= \left\langle n\left(\sum_j(u_j - u_i)\nabla\phi_j\right)\cdot\nabla\phi_i\right\rangle && \text{(A.11)}\\
&= \sum_j(u_j - u_i)\left\langle n\nabla\phi_j\cdot\nabla\phi_i\right\rangle && \text{(A.12)}
\end{aligned}$$

Until now, only mathematical operations have been done. To complete the CVFE method and obtain Eq. (4.36), an approximation is done on the diffusion coefficient n: it is computed at the upwind node to handle some numerical issues, like avoiding the negative depths that can occur with the NI SWE. This coefficient is then taken out of the integral. It should be noted that the sum over j in Eq. (4.36) is implicitly done via the matrix notation.

A.4 Equivalence of the specific storage coefficient

Sometimes, the following form of RE is used to account for medium compressibility:

$$\frac{\partial\theta}{\partial t} + S_s S_w\frac{\partial\psi}{\partial t} = \nabla\cdot(\mathbf{K}\cdot\nabla(\psi - z)), \tag{A.13}$$

with S_s the specific storage coefficient (specific to material) and $S_w = \theta/\theta_s$ the water saturation. The ψ-form reads:

$$(C + S_s S_w)\frac{\partial \psi}{\partial t} = \nabla \cdot (\mathbf{K} \cdot \nabla(\psi - z)), \tag{A.14}$$

with $C = \partial\theta/\partial\psi$ the specific moisture capacity. This form looks like the classic ψ-form with a modified C. Let us denote $C^* = \partial f_\theta^*/\partial\psi = C + S_s S_w$ the modified specific moisture capacity with $f_\theta^*(\psi)$ the modified retention curve. We have ($\theta = f_\theta(\psi)$):

$$f_\theta^*(\psi) = \int_{-\infty}^{\psi} \left(\frac{\partial f_\theta(\psi)}{\partial \psi} + \frac{S_s f_\theta(\psi)}{\theta_s} \right) d\psi \tag{A.15}$$

$$= f_\theta(\psi) - f_\theta(-\infty) + \frac{S_s}{\theta_s} \int_{-\infty}^{\psi} f_\theta(\psi) d\psi + \text{cst} \tag{A.16}$$

$$= \begin{cases} f_\theta(\psi) - \theta_r + \frac{S_s}{\theta_s} \int_{-\infty}^{\psi} f_\theta(\psi) d\psi + \text{cst} & \psi < 0 \\ \theta_s - \theta_r + \frac{S_s}{\theta_s}\theta_s\psi + \frac{S_s}{\theta_s} \int_{-\infty}^{0} f_\theta(\psi) d\psi + \text{cst} & \psi \geq 0 \end{cases} \tag{A.17}$$

$$= \begin{cases} f_\theta(\psi) + \frac{S_s}{\theta_s} \int_{0}^{\psi} f_\theta(\psi) d\psi & \psi < 0 \\ \theta_s + S_s\psi & \psi \geq 0 \end{cases} \tag{A.18}$$

as $f_\theta(-\infty) = \theta_r$, $f_\theta(\psi \geq 0) = \theta_s$ and the constant from the integral is taken as $\text{cst} = \theta_r - \frac{S_s}{\theta_s} \int_{-\infty}^{0} f_\theta(\psi) d\psi$. This essentially adds a linear compression to the pressure in saturated areas. In unsaturated areas, compression effects are supposed to be negligible, but it is not the case for low values of ψ in materials with β approaching 1 or with large values of θ_r.

An equivalent way to represent (A.13) is

$$\begin{cases} \frac{\partial \theta^*}{\partial t} = \nabla \cdot (\mathbf{K} \cdot \nabla(\psi + z)) \\ \theta^* = f_\theta^*(\psi) \end{cases} \tag{A.19}$$

where θ^* can increases in saturated areas – which makes sense as more water is present within the increased representative volume – but can also take negative values in very dry conditions, which is not at all physical. The addition of a simple slope S_s in f_θ when $\psi > 0$, with appropriate smoothing around $\psi = 0$ should be preferred to the use of Eq. (A.18).

A.5 Von Neumann analysis for the predictor-corrector scheme

A simplified Von-Neumann analysis is produces for the proposed implicit-explicit predictor-corrector scheme of Chapter 5. The classical test equation is

modified into the following linear system of two equations:

$$C(\psi)\frac{\partial \psi}{\partial t} = \lambda\psi, \tag{A.20}$$

$$\frac{\partial \theta}{\partial t} = \lambda\psi \tag{A.21}$$

with λ a complex number. We want to study the stability of the discretization with implicit Euler on the first equation and the discretization with explicit Euler on the second one. Exactly the same function $\lambda\psi$ is applied on both equations:

$$C(\psi)\frac{\psi_{k+1} - \psi_k}{\Delta t} = \lambda\psi_{k+1}, \tag{A.22}$$

$$\frac{\theta_{k+1} - \theta_k}{\Delta t} = \lambda\psi_{k+1} \tag{A.23}$$

rearranging the terms:

$$\psi_{k+1} = \psi_k + \Delta t\frac{\lambda\psi_{k+1}}{C(\psi)}, \tag{A.24}$$

$$\psi_{k+1} = \psi_k\left(\frac{1}{1 - \Delta t\lambda/C(\psi)}\right), \tag{A.25}$$

$$\theta_{k+1} = \theta_k + \Delta t\lambda\psi_k\left(\frac{1}{1 - \Delta t\lambda/C(\psi)}\right) \tag{A.26}$$

If C is zero, ψ is zero (the time derivative cannot influence the solution) and θ remains constant. The analysis continue with the hypothesis of a positive and constant C.

The amplification factor for ψ is classical for the Implicit Euler scheme:

$$\psi_k = \left(\frac{1}{1 - \Delta t\lambda/C}\right)^k \hat{\psi} \tag{A.27}$$

with $\hat{\psi}$ the initial condition for ψ. The second equation does not influence that result in this analysis and we can then deduce the following stability condition:

$$\left|\frac{1}{1 - \Delta t\lambda/C}\right| < 1, \tag{A.28}$$

$$1 - \Delta t\lambda/C > 1 \text{ or } 1 - \Delta t\lambda/C < -1, \tag{A.29}$$

$$\Delta t\lambda/C < 0 \text{ or } \Delta t\lambda/C > 2 \tag{A.30}$$

with all the constraints on the variables ($\Delta t > 0$, $C > 0$ and $\lambda < 0$), only the first condition can be satisfied and the ψ variable is unconditionally stable if

C is positive. If C was variable, the amplification factor would be much more complex to derive, but it can be seen that the product of smaller-than-one positive numbers, whatever the numbers, will tend towards zero.

For the θ variable, we find the following series:

$$\text{with } A = \left(\frac{1}{1 - \Delta t\lambda/C}\right), \tag{A.31}$$

$$\theta_{k+2} = \theta_{k+1} + \Delta t\lambda\psi_{k+1}A, \tag{A.32}$$

$$= \theta_k + \Delta t\lambda\psi_k A + \Delta t\lambda\psi_k AA, \tag{A.33}$$

$$= \theta_k + \Delta t\lambda\psi_k A + \Delta t\lambda\psi_k A^2, \tag{A.34}$$

$$\theta_{k+3} = \theta_{k+3} + \Delta t\lambda\psi_{k+2}A, \tag{A.35}$$

$$= \theta_k + \Delta t\lambda\psi_k A + \Delta t\lambda\psi_k A^2 + \Delta t\lambda\psi_k A^3, \tag{A.36}$$

$$\ldots \tag{A.37}$$

$$\theta_k = \hat{\theta} + \Delta t\lambda\hat{\psi}\sum_{i=1}^{k} A^i \tag{A.38}$$

with $\hat{\theta}$ the initial condition for θ.

It can be seen that the series diverges if the absolute value of A is greater than one. The stability condition for θ and ψ is identical and the overall scheme is unconditionally stable.

The largest difference between this analysis and the applied scheme is the application between each time steps of $\psi_{n+1} = f_\theta^{-1}(\theta_{n+1})$ in unsaturated areas. That leads to complex feedback loops which are too complex to study with a simple Von Neumann analysis.

A.6 Parametrization of the Vanderborght et al. test cases

Here follows the description of the Vanderborght et al. test cases concerning the RE alone (without tracers) [114].

Table A.1: Soil properties for analytical benchmarks

	α (m^{-1})	β	K (m/s)	θ_s	θ_r
Sand	15	3.0	1.157×10^{-4}	0.43	0.045
Loam	4	1.6	5.787×10^{-6}	0.43	0.080
Clay	1	1.1	1.157×10^{-6}	0.40	0.100

Table A.2: Parametrization for analytical benchmarks

	Soil type	thickness	discr.	duration	time step
1a	loam/sand	0.5 m/1.5 m	1 cm	1 h	30 s
1b	sand/loam	0.5 m/1.5 m	1 cm	1 h	60 s
1c	clay/sand	0.5 m/1.5 m	1 cm	1 h	60 s
2	loam	0.54 m	1 cm	5 h	30 s
3a	sand	2 m	1 cm	7.2 h	25 ms→5 s
3b	loam	2 m	1 cm	1 day	1 s
3c	clay	2 m	1 cm	12 h	1 s
4a	sand	2 m	1 cm&0.25 cm	1 day	1 s
4b	loam	2 m	1 cm&0.25 cm	2 day	5 s
4c	loam	2 m	1 cm&0.25 cm	10 day	30 s
4d	clay	2 m	1 cm&0.25 cm	6 day	10 s

Table A.3: Initial and boundary conditions for analytical benchmarks

	initial condition	top b.c.	bottom b.c.
1a	ψ=−2 m	J_N= 5.787 10^{-8} m/s	J_N= K_{zz}
1b	ψ=−2 m	J_N= 5.787 10^{-8} m/s	J_N= K_{zz}
1c	ψ=−2 m	J_N= 5.787 10^{-8} m/s	J_N= K_{zz}
2	ψ=−z−0.54 m	J_N= −5.787 10^{-8} m/s	ψ_D= 0
3a	ψ=−4 m	J_N= 1.157 10^{-5} m/s[a]	J_N= K_{zz}
3b	ψ=−4 m	J_N= 1.157 10^{-5} m/s[a]	J_N= K_{zz}
3c	ψ=−4 m	J_N= 1.157 10^{-5} m/s[a]	J_N= K_{zz}
4a	ψ=−0.4 m	J_N= −3.472 10^{-8} m/s[b]	J_N= 0
4b	ψ=−2 m	J_N= −3.472 10^{-8} m/s[b]	J_N= 0
4c	ψ=−2 m	J_N= −1.157 10^{-8} m/s[b]	J_N= 0
4d	ψ=−2 m	J_N= −3.472 10^{-8} m/s[b]	J_N= 0

[a] When $\psi < 0$; otherwise, $\psi = 0$
[b] When $\psi > -100$ m; otherwise, $\psi = -100$ m

A.7 Applying the corrector in one vector operation

As the same right-hand term is applied on ψ- and mixed-forms of RE, it is easy to obtain:

$$C^n \frac{\psi^{n+1} - \psi^n}{\Delta t} = \nabla \cdot (\mathbf{K}^n \cdot \nabla(\psi^{n+1} - z)) + s, \tag{A.39}$$

$$\frac{\theta^{n+1} - \theta^n}{\Delta t} = \nabla \cdot (\mathbf{K}^n \cdot \nabla(\psi^{n+1} - z)) + s, \tag{A.40}$$

$$C^n(\psi^{n+1} - \psi^n) = (\theta^{n+1} - \theta^n) \tag{A.41}$$

$$\theta^{n+1} = \theta^n + C^n(\psi^{n+1} - \psi^n) \tag{A.42}$$

witch is a simple vector-wise operation of known values. It has to be noted that in absence of mass-lumping, this expression becomes more complex as the time derivative terms are not anymore diagonal matrices. In case of a DG FE method it can still be solved per block.

A.8 Example of python script: the 3D Ditch test case

Below, we present an example of Python script used in the SLIM framework. This is the 3D ditch test case (see Section 5.5.2) which has only been slightly formated and commented to be self-descriptive.

```
from Richards import *
import utilities as u

# The units used in this script are SI (m, s, kg, ...)

# These custom C++ functions are compiled on-the-fly.
# They can also be written in python at a slight performance cost.
# - ditchHead returns the height difference between the ditch
#   water level and the nodal z coordinate.
# - ditchSwitch returns a positive value if a Neumann BC
#   is imposed inside the active part of the ditch and a
#   negative one for Dirichlet BC.
# - farHead returns the initial condition. It is imposed as
#   a Dirichlet BC at the far sides of the domain.
# - hydraulicHead speaks by itself
CFunctions = '''
#include "fullMatrix.h"
#include "function.h"
#include "dgConservationLaw.h"
extern "C" {

  void ditchHead(dataCacheMap *, fullMatrix<double> &out,
                                 fullMatrix<double> &coord,
                                 fullMatrix<double> &time) {
    double head = 0.0;
    const double t = time(0, 0) / 3600.;
```

```
    if(t<=0.00000) head = 0.00000;
    else if(t<=0.03330) head = 0.07000;
    else if(t<=0.10000) head = 0.13000;
    ... // rem: file truncated here
    else if(t<=4.91670) head = 0.07100;
    else if(t<=5.08330) head = 0.03300;
    else head = -1.00000;
    for (size_t i = 0; i < out.size1(); i++) {
      double depth = -coord(i, 2);
      double nodeHead = head + depth - 0.8;
      out.set(i, 0, std::max(0., nodeHead) );
    }
  }

  // if positive head in BC, Dirichet; otherwise Neumann no-flux
  void ditchSwitch(dataCacheMap *m, fullMatrix<double> &out,
                                    fullMatrix<double> &ditchHead) {
    size_t nE = m->nElement();
    size_t nP = m->nPointByElement();
    for (size_t iFace = 0; iFace < nE; ++iFace) {
      size_t iFirstPt = iFace * nP;
      bool havePositive = false;
      for (size_t iPt = iFirstPt; iPt < iFirstPt + nP; iPt++)
        havePositive |= ditchHead(iPt, 0) > 0;
      for (size_t iPt = iFirstPt; iPt < iFirstPt + nP; iPt++)
        out.set(iPt, 0, havePositive ? -1 : 1); // positive = neumann
    }
  }

  void farHead(dataCacheMap *, fullMatrix<double> &out,
                               fullMatrix<double> &coord) {
    for (size_t i = 0; i < out.size1(); i++) {
      out.set(i, 0, - coord(i, 2) - 1.9); // rem: same as BC !!
    }
  }

  void hydraulicHead(dataCacheMap *, fullMatrix<double> &out,
                                fullMatrix<double> &h,
                                fullMatrix<double> &xyz) {
    for (size_t i = 0; i < out.size1(); i++) {
      double z = xyz(i, 2);
      out.set(i, 0, h(i,0) + z);
    }
  }

}
''',
u.makeLib(CFunctions, "dylibs/tmp_ADag3D.dylib")
# -- end of on-the-fly compilation --

# User-defined functions from compiled code
TIME = function.getTime()
COORD = function.getCoordinates()
ditchHead = functionC(tmpLib, "ditchHead", 1, [COORD, TIME])
```

```
ditchSwitch = functionC(tmpLib, "ditchSwitch", 1, [ditchHead])
farHead = functionC(tmpLib, "farHead", 1, [COORD])

# Mesh
#   The mesh is generated with GMSH, and partitionned with METIS
#   (here in 4 partitions to run on 4 processors). Tags are put
#   in the geometry definition, defining 4 volumes (for different
#   soil properties) and 7 surfaces (for the BC).
mesh = 'meshes/ditch_4p.msh'

# Parameters
duration = 3600 * 8  # simulation duration
maxDT = 5            # maximum time step
                     # => mostly the time step excepted if
                     #    exports are not aligned on it
maxDT_start = 1      # max Dt at the simulation beginning
T_end_start= 360     # 0-6min: maxDT_start, 6min-end: maxDT
printEach = 10       # print global infos each 10 iterations
exports = "360:0.1,0.2,0.3,0.4,0.5,1,5,10,15,20,40,60,120,240"
                     # export state each 6min and at specific times

# Soil properties (Van Genuchten parametrization)
#   Mesured values (parameters used in Dages et al. 2008)
soil_1  = soilVG(13.6, 1.296, 1.90e-4, 0.37, 0.000051) #684 mm/h
soil_24 = soilVG(2.4 , 1.572, 1.03e-5, 0.33, 0.000550) #37  mm/h
soil_3  = soilVG(5.2 , 2.279, 6.75e-5, 0.31, 0.000570) #234 mm/h
#   Homogeneous (another possible set of parameters)
# soil_1  = soilVG(13.6, 1.296, 5.33e-5, 0.37, 0.000051) #192 mm/h
# soil_24 = soil_1
# soil_3  = soil_1
#   Heterogeneous (another possible set of parameters)
# soil_1  = soilVG(13.6, 1.296, 9.17e-5, 0.37, 0.000051) #330 mm/h
# soil_24 = soilVG(2.4 , 1.572, 4.17e-5, 0.33, 0.000550) #150 mm/h
# soil_3  = soilVG(5.2 , 2.279, 1.25e-4, 0.31, 0.000570) #450 mm/h

# Apply soil properties and map them to volume tags of the mesh
sm = SoilMap()
sm.add("v1", soil_1)
sm.add("v2", soil_24)
sm.add("v3", soil_3)
sm.add("v4", soil_24)

# Creation of the model object, with different options
# - PETSc: library of linear system solvers. Here a AMG method is
#          commented. The chosen method is a block Jacobi used as
#          preconditioner followed by an incomplete LU factorization.
# - interpolator: method to approximate the K and C functions. Here
#                 such a method is not used.
# - h_th_link: the method to project the inverse of f_theta on psi:
#              interpolation or L2 projection
#PETSc = '-pc_type hypre -pc_hypre_type boomeramg -ksp_type preonly'
PETSc = '-pc_type bjacobi -sub_pc_type ilu'
Roptions = RichardsOptions(interpolator=None, petscOptions=PETSc)
R = RichardsPredCorr(mesh, sm, options=Roptions, h_th_link='interp')
```

```
# Boundary conditions
# - a mixed Neumann/Dirichlet BC in the active part of the ditch
# - a Dirichlet BC on the far sides of the domain
# - a no-flux Neumann BC on each remaining faces
C0 = functionConstant(0.)
R.bndMixed(ditchSwitch, ditchHead, C0, ['ditchWet'])
R.bndDirichlet(farHead, ['farVertical'])
R.bndNeumann(C0, ['top', 'bottom', 'parallel',
                  'perpendicular', 'ditchDry'])


# Non-linear solver (Newton) settings
R.constructSolvers()              # finilize initialization
R.IE.getNewton().setVerb(0)       # no verbosity for N-L iterations
R.IE.getNewton().setMaxIt(1)      # 1 N-L iteration
R.IE.getNewton().setAtol(1e-9)    # absolute tolerance


# Initial condition
#   Here defined as a system at equilibrium, with a water table
#   situated at 1.9m under the soil surface. A C++ function is
#   not useful here as the program run it once and for all.
def ini(val, xyz):
  for i in range(0, xyz.size1()):
    z = xyz.get(i, 2)
    val.set (i, 0, -z - 1.9) # 2002=1.9 ; 2004=2.0  (Dages 2008)
INI = functionPython(1, ini, [R.XYZ])
R.seth(INI, h_th_link='interp')

# Time-stepper
#   Determine the optimal time step based on the maximum allowed and
#   the requested exports
h = slimIterateHelper()
h.setInitialDate(0)
h.setFinalDate(duration)
h.setExport(exports)
h.setMaxDt(maxDT_start)
h.setNbMaxIter(1000000)
h.printExport(2)

# Exporter (outputs)
#   Special function and variable fields used for export
hydraulicHead = functionC(tmpLib, "hydraulicHead", 1, [R.HF, R.XYZ])
CGstruct = dgCGStructure(R.gc)          # continuous Galerkin struct
H_smoothed = dgDofContainer(R.gc, 1)    # CG pressure head field
W_smoothed = dgDofContainer(R.gc, 1)    # CG water content field
phi_smoothed = dgDofContainer(R.gc, 1) # CG hydraulic head field
#   Output file path
outputDir = 'output'
outputname = outputDir + "/Ditch_3D_4p"
try: os.mkdir(outputDir);
except: 0;
```

```
h.setFileName(outputname)
#   Functions and fields to export
h.addExport(R.H, "h")                       # DG pressure head
h.addExport(R.W, "th")                      # DG water content
h.addExport(R.H, hydraulicHead, "phi")      # DG hydraulic head
h.addExport(H_smoothed, "h_CG")             # CG pressure head
h.addExport(W_smoothed, "th_CG")            # CG water content
h.addExport(phi_smoothed, "phi_CG")         # CG hydraulic head
h.exportBin(True)                           # export also in binary
                                            # (default format: msh)

# Initialization of the temporal loop
i=0; t=0; dt=1; norm=0; normH=0;
u.initClock()
u.printInfos(i, dt, t, '|Mass|\t%.16g' % (R.getGlobalMass()))

# Temporal loop
while not h.isTheEnd():

  # When export is needed, pre-compute CG projections
  if h.willExport():
    CGstruct.mean(R.H, H_smoothed)
    CGstruct.mean(R.W, W_smoothed)
    phi_smoothed.interpolate(hydraulicHead, R.H)
    CGstruct.mean(phi_smoothed, phi_smoothed)

  # Iterate number, time step and time
  # Do also export files if requested
  i = i + 1
  dt = h.getNextTimeStepAndExport()
  t = h.getCurrentTime()

  # After a while, increase the time step
  if(t > T_end_start):
    h.setMaxDt(maxDT)

  # Iterate Richards equation model
  nbSub = R.iterate(dt, t)

  # Print informations about the simulation status
  if (i % printEach == 0) :
    gmass = R.getGlobalMass()
    u.printInfos(i, dt, t, '|Mass|\t%.16g' % (gmass))

u.printOnce('Last iteration :')
u.printInfos(i, dt, t, '|Mass|\t%.16g' % (R.getGlobalMass()))
```

A.9 Towards an integrated model of land-sea continuum

The original goal of this thesis was to be able to follow the water flows from the raindrop on the soil to the ocean. In our opinion, the Chapter 5 describes a model for subsurface water able to handle such a scale. However, to be

operational for environmental flows, a certain number of improvements have to be done. The current state of the SLIM framework is depicted in Chapter 6, Fig. 6.3.

Improving the subsurface water flow model

One needs to develop an algorithm to spread the excess mass of the current model. The removal of these artifacts will allow the use of larger time steps without leading to mass accumulation. A special attention should be paid to keep those additional fluxes along with the solution (or adding them to the solution fluxes), in order to keep local conservation of fluxes (*e.g.* to be used in a tracer model).

One should test the model for different types of boundary conditions, especially for long term evaporation where the dynamics is very slow and large time steps can be used. Stability issues can possibly appear at those scales of time, even if none have been observed yet.

One should add an adaptive time stepping strategy to the model. The simplest solution is to increase/decrease the time step based on heuristics like the linear solver convergence or an error indicator as in Eq. (5.36). However, we suggest to make some experiments on our predictor with the second-order time stepping presented in [45], where a unique non-linear iteration is used, like in our model. A time truncation error estimation is derived from 1^{st} and 2^{nd} order time integration schemes, and the time step is then adapted to keep this estimate under a prescribed value.

One should investigate more efficient methods to solve elliptic equations in parallel. This is a recurrent research topic and new methods are likely to emerge in near future.

One should migrate the existing code towards the production version of SLIM. Indeed, the code has been forked in 2015 and not merged with the master branch since then.

The variable transform of Section 2.5.2 could also be optimized per soil type. The discontinuities of the transformed pressure head ψ^* could then be treated with the modification to the DG method developed to handle physical discontinuities of θ (see Eq. (3.33) and Section 3.3.5).

Modeling environmental subsurface flow

One needs to develop an Eulerian tracer model for the RE. Due to the local conservation of the fluxes, it should be straightforward. It is expected that a Lagrangian tracer model will hit some issues with the non-continuity of the fluxes, which are constant by element. However, some post-processes methods

exist to handle such issues and produce continuous fluxes. Another possibility is to modify the model with a separate spatial discretization of the fluxes to make them continuous, like in the LDG method for RE [5]. The proposed predictor-corrector method applied to another DG spatial integration scheme should not see additional complication, as it mostly concerns the time integration scheme. One also needs to develop a tracer model for the NISWE. Our CVFE method is based on explicit fluxes between nodes which should facilitate this integration.

A macroscale pores model should be included to handle large scale dynamics. Indeed, it has been observed that even in our medium-scale final 3D test case (the ditch), their impact is significant on the results (split of dynamics between the water table recharge and the porous infiltration front). Dual porosity is the usual way to model macroscale pores, simply using two identical models of RE with different parametrizations, coupled together with exchange coefficients when imbalance occurs in the pressure head. Such a development should be feasible inside the present implementation.

One needs to implement a variable boundary condition (BC) including both infiltration (*e.g.* rain) and evapotranspiration (*i.e.* evaporation in air and plants transpiration). Our recommendation is to use a weak Dirichlet BC with a prescribed pressure head, which in practice is applied as a flux proportional to the pressure difference. For infiltrations, this flux is limited by the available surface water. Prescribing a flux for evaporation is quite dangerous as it is quickly limited by the soil and the model can crash if too much is required. It is also possible to model the transpiration as a sink term in the first meters of soil (up to the roots extent).

Improving the surface-subsurface coupling

One needs to test the coupling developed in Chapter 4 with our predictor-corrector RE model (Chapter 5). No additional issues are expected except that as our subsurface time step can become larger, the surface model keeps its limitations. A first solution should be to slightly degrade the surface model by limiting its diffusivity coefficient, allowing larger time steps. A second one should be to use a simpler model for the surface runoff, like the kinematic wave approximation, but then some additional work is required for the surface water reservoirs (*e.g.* lakes or swamps). If the times steps are too different, a third solution can be to split the time step, allowing several time steps for the surface water model when the subsurface model makes one step forward. In this last case, additional treatments are needed between the two models to preserve mass balance.

One should slightly modify the surface model to add extra processes via parameters such as the *local surface storage* to account for micro-topography or the *interception storage* for rain interception by vegetation or urban structures

[18]. Indeed, such processes can strongly delay actual surface fluxes during a rain event and therefore increase the final infiltrated water volume.

Coupling the subsurface/runoff model with a 1D river flow model

We consider that the rivers are placed along the edges of the 2D surface model(s). One needs to couple the river model with the runoff model. It should be quite easy by imposing the river water height as a weak Dirichlet BC for the runoff model and a source term for the river model. The river model then has to be coupled with the subsurface model. It should be done in a similar fashion as the subsurface/runoff coupling, by specifying the water height as a hydraulic head for the subsurface model. However, the difference is that this coupling should be applied on subsurface element edges only. The rivers tracks delineation should ideally be computed from a digital elevation model (DEM). Due to the very different water velocities, the temporal discretization for the SWE should be finer than the one used for subsurface flows. One should use nested time stepping or another appropriate sub-time-stepping method.

Coupling the subsurface/runoff model with a 2D free-surface model for water bodies and large rivers

One needs to couple the 2D SWE for barotropic free-surface flow with the subsurface model. This coupling should be identical to the one used for subsurface and runoff models. The 2D SWE also has to be coupled with the runoff model. If their interface is well defined at element boundaries, this coupling should be quite easy and defined as a flux determined by a weak Dirichet BC on one of the models and applied as a Neumann BC on the other model. The boundary between those two surface flow models can be fixed at the mesh level and the dry elements of the 2D SWE model handled through wetting and drying techniques.

One should determine the best method to iterate between the fast physics of SWE and the slow ones of the RE.

Coupling a subsurface/runoff model with a 3D baroclinic model for stratified flows

Although such a coupling is interesting in the conceptual point of view, it is not the most useful firsthand. Indeed, the expected exchanges are quantitatively low in front of the 3D ocean dynamics. In most cases a 2D SWE should be sufficient, even for salt water intrusion studies.

One needs to couple this 3D model with the subsurface model. This coupling should be straightforward by imposing a weak Dirichlet BC in the subsurface

model, identifying the pressure head with the water column height. Consequently, a Neumann BC is imposed to the 3D free-water model with the resulting flux.

One needs to couple this 3D model with the runoff model. The same method than for the 2D SWE/runoff should be used, but at the edge level. A similar issue with the delineation between both models arises, excepted that the difference in dimensionality potentially adds complexities.

One should consider the interface between the two components, as a special layer is expected (alluvions, silt, etc.). For instance, something similar to the *FOEC* method can be investigated. The issue with subsurface discretization found in Section 4.4.3 should be absent as large jumps are not expected at this interface. Eventually, the fluxes at this interface can be challenging to measure and parametrize.

If salt water is considered (seas, estuaries, etc.), one needs to modify the RE with a variable water density to enable density-driven flows. The consequences of such a modification on mass conservation or stability have not been investigated. A tracer model for salinity is obviously mandatory and has to be tightly coupled with this modified RE.

Coupling the tracers

The corresponding tracers models should also be coupled accordingly for each sub-model and coupling. The coupling between 1D and 2D SWE is already implemented in the SLIM framework.

Macro-model efficiency

A great challenge is to obtain coupled models efficients on HPC. To scale well, the load of each model should be equally spread on the different workers. A classical way is partitioning the computational domains into the number of workers. As the time step will probably be different for each model, choices have to be made like concurrent runs of multiple models (but with a weaker coupling), or sequential runs of all models (with probably more bottlenecks), or full coupling within one non-linear system (likely leading to convergence issues), etc. Some areas may need more computational time (*e.g.* rain over a small part of the domain) and would therefore unbalance the load if the spatial domain partitioning is kept as such. The optimal time step between such areas and the others will also be different (a single small event can impose a small time step for the whole simulation). A first recommendation is to partition spatially-close models together (*e.g.* river, runoff and soil). The coupling of those models can then mostly use information directly accessible by the worker, and the workload should be of the same order of magnitude for each model.

The number of partitions can be larger than the number of workers to allow a dynamical load balancing. As it is very difficult to achieve perfect scaling for all models, most time-consuming ones should be handled in priority.

References

[1] C. Dages, M. Voltz, P. Ackerer, Parameterization and evaluation of a three-dimensional modelling approach to water table recharge from seepage losses in a ditch, Journal of Hydrology 348 (3-4) (2008) 350–362. Cited in pages 8, 113, 117, 119, and 121.

[2] M. W. Farthing, F. L. Ogden, Numerical solution of richards' equation: A review of advances and challenges, Soil Science Society of America Journal 81 (6) (2017) 1257–1269. Cited in page 9.

[3] R. Niswonger, D. Prudic, Comment on "evaluating interactions between groundwater and vadose zone using the HYDRUS-based flow package for MODFLOW" by navin kumar c. twarakavi, jirka simunek, and sophia seo, Vadose Zone Journal 8 (3) (2009) 818–819. Cited in page 11.

[4] D. Or, P. Lehmann, S. Assouline, Natural length scales define the range of applicability of the richards equation for capillary flows, Water Resources Research 51 (9) (2015) 7130–7144. Cited in pages 11 and 137.

[5] H. Li, M. Farthing, C. Miller, Adaptive local discontinuous Galerkin approximation to Richards' equation, Advances in Water Resources 30 (9) (2007) 1883–1901. Cited in pages 11, 32, and 154.

[6] T. De Maet, E. Hanert, M. Vanclooster, A fully-explicit discontinuous Galerkin hydrodynamic model for variably-saturated porous media, Journal of Hydrodynamics 26 (4) (2014) 594–607. Cited in pages 43, 71, 73, 80, 81, 83, 95, 102, 106, 108, and 128.

[7] T. De Maet, F. Cornaton, E. Hanert, A scalable coupled surface–subsurface flow model, Computers & Fluids 116 (2015) 74–87. Cited in page 69.

[8] T. De Maet, E. Hanert, Towards a multi-scale model of the land-sea continuum, in: ENVITAM PhD day, 2010. no citations

[9] T. De Maet, E. Hanert, A material discontinuity capturing model based on discontinuous Galerkin method, in: 35èmes Journées du GFHN, 2010. no citations

[10] T. De Maet, E. Hanert, Towards a 3D continuous-discontinuous Galerkin sub-surface water-flow model, in: Joint ISSH and ICWRER Symposium, 2010. no citations

[11] T. De Maet, E. Hanert, An explicit discontinuous-Galerkin surface-subsurface water flow model, in: Fifth International Conference on Advanced COmputational Methods in ENgineering (ACOMEN 2011), 2011. no citations

[12] T. De Maet, E. Hanert, Towards an integrated model of the surface-subsurface water cycle, in: PhD Student Day ENVITAM, 2012. no citations

[13] T. De Maet, E. Hanert, Towards an integrated and multi-scale model of the land-sea continuum, in: EGU General Assembly Conference Abstracts, Vol. 14, 2012, p. 9634. no citations

[14] T. De Maet, E. Hanert, E. Deleersnijder, T. Fichefet, V. Legat, J.-F. Remacle, S. Soares Frazao, M. Vanclooster, J. Lambrechts, C. König Beatty, et al., SLIM: A multi-scale model of the land-sea continuum, in: EGU General Assembly Conference Abstracts, Vol. 14, 2012, p. 9603. no citations

[15] T. De Maet, E. Hanert, A 3D explicit discontinuous-Galerkin groundwater model, in: EGU General Assembly Conference Abstracts, Vol. 14, 2012, p. 9665. no citations

[16] H. Diersch, P. Perrochet, On the primary variable switching technique for simulating unsaturated-saturated flows, Advances in Water Resources 23 (3) (1999) 271–301. Cited in pages 19, 22, 23, 26, 31, 102, and 125.

[17] S. Panday, P. Huyakorn, A fully coupled physically-based spatially-distributed model for evaluating surface/subsurface flow, Advances in Water Resources 27 (4) (2004) 361–382. Cited in pages 19, 36, and 37.

[18] R. Therrien, R. McLaren, E. Sudicky, S. Panday, A three-dimensional numerical model describing fully-integrated subsurface and surface flow and solute transport, Groundwater Simulations Group (2012) 479. Cited in pages 19, 37, 71, 72, 80, 83, 84, 90, 106, 143, and 155.

[19] S. Weill, E. Mouche, J. Patin, A generalized Richards equation for surface/subsurface flow modelling, Journal of Hydrology 366 (1-4) (2009) 9–20. Cited in pages 20, 37, 71, 83, and 87.

[20] J. Simunek, M. Sejna, M. V. Genuchten, The HYDRUS software package for simulating the two- and three-dimensional movement of water, heat, and multiple solutes in variably-saturated media, technical manual, PC Progress, Prague, Czech Republic (2006) 1–241. Cited in pages 22, 23, 37, 80, 102, and 106.

[21] M. Islam, A mass lumping and distributing finite element algorithm for modeling flow in variably saturated porous media, Journal of the korean society for industrial and applied mathematics 20 (3) (2016) 243–259. Cited in pages 22 and 23.

[22] H. Li, M. Farthing, C. Dawson, C. Miller, Local discontinuous Galerkin approximations to Richards' equation, Advances in Water Resources 30 (3) (2007) 555–575. Cited in pages 23, 32, 71, and 72.

[23] C. Downer, F. Ogden, Appropriate vertical discretization of richards' equation for two-dimensional watershed-scale modelling, Hydrological Processes 18 (1) (2004) 1–22. Cited in pages 24 and 95.

[24] K. Rathfelder, L. Abriola, Mass conservative numerical solutions of the head-based Richards equation, Water Resources Research 30 (9) (1994) 2579–2586. Cited in pages 26 and 52.

[25] L. Orgogozo, N. Renon, C. Soulaine, F. Hénon, S. Tomer, D. Labat, O. Pokrovsky, M. Sekhar, R. Ababou, M. Quintard, An open source massively parallel solver for Richards equation: Mechanistic modelling of water fluxes at the watershed scale, Computer Physics Communications 185 (12) (2014) 3358–3371. Cited in pages 26, 36, 37, 39, 40, and 136.

[26] C. Paniconi, M. Putti, A comparison of Picard and Newton iteration in the numerical solution of multidimensional variably saturated flow problems, Water Resources Research 30 (12) (1994) 3357–3374. Cited in pages 26 and 32.

[27] M. Celia, R. Zarba, E. Bouloutas, A general mass-conservative numerical solution for the unsaturated flow equation, Water Resources Research 26 (7) (1990) 1483–1496. Cited in pages 28, 52, 71, and 102.

[28] F. Lehmann, P. Ackerer, Comparison of iterative methods for improved solutions of the fluid flow equation in partially saturated porous media, Transport in porous media 31 (3) (1998) 275–292. Cited in page 28.

[29] G. Williams, C. Miller, C. Kelley, Transformation approaches for simulating flow in variably saturated porous media, Water Resources Research 36 (4) (2000) 923–934. Cited in page 28.

[30] M. Kirkland, R. Hills, P. Wierenga, Algorithms for solving Richards' equation for variably saturated soils, Water Resources Research 28 (8) (1992) 2049–2058. Cited in pages 30, 102, and 125.

[31] L. Pan, P. Wierenga, A transformed pressure head-based approach to solve Richards' equation for variably saturated soils, Water Resources Research 31 (4) (1995) 925–931. Cited in page 30.

[32] R. Haverkamp, M. Vauclin, J. Touma, P. Wierenga, G. Vachaud, A comparison of numerical simulation models for one-dimensional infiltration 1, Soil Science Society of America Journal 41 (2) (1977) 285–294. Cited in page 30.

[33] P. J. Ross, K. L. Bristow, Simulating water movement in layered and gradational soils using the Kirchhoff transform, Soil Science Society of America Journal 54 (6) (1990) 1519–1524. Cited in page 30.

[34] P. J. Ross, Cubic approximation of hydraulic properties for simulations of unsaturated flow, Water Resources Research 28 (10) (1992) 2617–2620. Cited in page 30.

[35] P. Forsyth, Y. Wu, K. Pruess, Robust numerical methods for saturated-unsaturated flow with dry initial conditions in heterogeneous media, Advances in Water Resources 18 (1) (1995) 25–38. Cited in pages 31 and 71.

[36] K. Krabbenhøft, An alternative to primary variable switching in saturated-unsaturated flow computations, Advances in Water Resources 30 (3) (2007) 483–492. Cited in page 31.

[37] D. Baker, Applying higher order DIRK time steps to the "modified Picard" method, Ground Water 33 (2) (1995) 259–263. Cited in page 32.

[38] M. Farthing, C. Kees, C. Miller, Mixed finite element methods and higher order temporal approximations for variably saturated groundwater flow, Advances in Water Resources 26 (2003) 373–394. Cited in page 32.

[39] L. Guarracino, F. Quintana, A third-order accurate time scheme for variably saturated groundwater flow modelling, Communications in Numerical Methods in Engineering 20 (2004) 379–389. Cited in page 32.

[40] P. Solin, M. Kuraz, Solving the nonstationary Richards equation with adaptive hp-FEM, Advances in water resources 34 (9) (2011) 1062–1081. Cited in pages 32 and 42.

[41] C. Zambra, M. Dumbser, E. Toro, N. Moraga, A novel numerical method of high-order accuracy for flow in unsaturated porous media, International Journal for Numerical Methods in Engineering 89 (2) (2012) 227–240. Cited in pages 32, 102, and 125.

[42] M. Tocci, C. Kelley, C. Miller, Accurate and economical solution of the pressure-head form of Richards' equation by the method of lines, Advances in Water Resources 20 (1) (1997) 1–14. Cited in page 32.

[43] G. Williams, C. Miller, An evaluation of temporally adaptive transformation approaches for solving Richards' equation, Advances in Water Resources 22 (8) (1999) 831–840. Cited in page 32.

[44] D. Kavetski, P. Binning, S. Sloan, Adaptive time stepping and error control in a mass conservative numerical solution of the mixed form of Richards equation, Advances in Water Resources 24 (6) (2001) 595–605. Cited in pages 32 and 45.

[45] D. Kavetski, P. Binning, S. Sloan, Noniterative time stepping schemes with adaptive truncation error control for the solution of Richards equation, Water Resources Research 38 (10) (2002) 1211. Cited in pages 32 and 153.

[46] W. Huang, L. Zheng, X. Zhan, Adaptive moving mesh methods for simulating one-dimensional groundwater problems with sharp moving fronts, Int. J. Numer. Meth. Engng 54 (11) (2002) 1579–1603. Cited in page 32.

[47] W. Huang, X. Zhan, Adaptive moving mesh modeling for two dimensional groundwater flow and transport, Recent Advances in Adaptive Computation 383 (2004) 283–296. Cited in pages 32 and 42.

[48] M. Bause, P. Knabner, Computation of variably saturated subsurface flow by adaptive mixed hybrid finite element methods, Advances in Water Resources 27 (6) (2004) 565–581. Cited in page 32.

[49] C. Miller, C. Abhishek, M. Farthing, A spatially and temporally adaptive solution of Richards' equation, Advances in Water Resources 29 (2006) 525–545. Cited in pages 32 and 42.

[50] S. Müller, Y. Stiriba, Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping, Journal of Scientific computing 30 (3) (2007) 493–531. Cited in page 32.

[51] C. Burstedde, J. A. Fonseca, S. Kollet, Enhancing speed and scalability of the ParFlow simulation code, Computational Geosciences 22 (1) (2018) 347–361. Cited in pages 34, 35, 39, 40, 41, and 136.

[52] A. H. Baker, R. D. Falgout, T. V. Kolev, U. M. Yang, Scaling HYPRE's multigrid solvers to 100,000 cores, in: High-Performance Scientific Computing, Springer, 2012, pp. 261–279. Cited in pages 34 and 129.

[53] Y. Notay, A. Napov, A massively parallel solver for discrete Poisson-like problems, Journal of computational physics 281 (2015) 237–250. Cited in pages 34 and 129.

[54] R. Maxwell, A terrain-following grid transform and preconditioner for parallel, large-scale, integrated hydrologic modeling, Advances in Water Resources 53 (2013) 109–117. Cited in pages 35, 39, 40, 45, 63, 71, 97, and 136.

[55] G. E. Hammond, P. C. Lichtner, R. Mills, Evaluating the performance of parallel subsurface simulators: An illustrative example with PFLOTRAN, Water Resources Research 50 (1) (2014) 208–228. Cited in pages 36, 39, 40, 129, and 136.

[56] R. G. Niswonger, D. E. Prudic, R. S. Regan, Documentation of the unsaturated-zone flow (UZF1) package for modeling unsaturated flow between the land surface and the water table with MODFLOW-2005, Tech. rep., USGS (2006). Cited in pages 36 and 37.

[57] B. Flemisch, M. Darcis, K. Erbertseder, B. Faigle, A. Lauser, K. Mosthaf, S. Müthing, P. Nuske, A. Tatomir, M. Wolff, R. Helmig, DuMux: DUNE for multi-{phase,component,scale,physics,...} flow and transport in porous media, Advances in Water Resources 34 (9) (2011) 1102–1112. Cited in pages 36, 37, 39, 40, 45, 65, 71, 129, and 137.

[58] J. VanderKwaak, Numerical simulation of flow and chemical transport in integrated surface-subsurface hydrologic systems, Dissertation Abstracts International Part B: Science and Engineering 60 (7) (2000) 3170. Cited in pages 37, 67, 71, 72, 76, 81, 83, 84, 87, 90, and 143.

[59] G. Yeh, H. Cheng, J. Cheng, H. Lin, W. Martin, A numerical model simulating water flow and contaminant and sediment transport in WAterSHed systems of 1-D stream-river network, 2-D overland regime, and 3-D subsurface media (WASH123D: Version 1.0), US Army Corps of Engineers (1998) 1—267. Cited in page 37.

[60] M. Morita, B. C. Yen, Numerical methods for conjunctive two-dimensional surface and three-dimensional sub-surface flows, International Journal for Numerical Methods in Fluids 32 (8) (2000) 921–957. Cited in page 37.

[61] M. Camporese, C. Paniconi, M. Putti, S. Orlandini, Surface-subsurface flow modeling with path-based runoff routing, boundary condition-based coupling, and assimilation of multisource observation data, Water Resources Research 46 (2) (2010) W02512. Cited in pages 37 and 71.

[62] S. Kollet, R. Maxwell, Integrated surface-groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model, Advances in Water Resources 29 (7) (2006) 945–958. Cited in pages 37, 39, 40, 45, 63, 71, 72, 73, 83, 84, 85, 87, 88, and 136.

[63] C. Dawson, A continuous/discontinuous Galerkin framework for modeling coupled subsurface and surface water flow, Computational Geosciences 12 (4) (2008) 451–472. Cited in pages 37 and 71.

[64] F. Cornaton, Ground Water manual (version 3.2.9), University of Neuchâtel, Switzerland (2012) 422. Cited in pages 37, 71, 83, and 90.

[65] H.-T. Hwang, Y.-J. Park, E. Sudicky, P. A. Forsyth, A parallel computational framework to solve flow and transport in integrated surface–subsurface hydrologic systems, Environmental modelling & software 61 (2014) 39–58. Cited in pages 37, 39, 40, and 136.

[66] H. S. Seo, J. Simunek, E. P. Poeter, Documentation of the HYDRUS package for MODFLOW-2000, the us geological survey modular groundwater model, IGWMC-International Ground Water Modeling Center. Cited in page 37.

[67] J. D. Hughes, C. D. Langevin, K. L. Chartier, J. T. White, Documentation of the surface-water routing (SWR1) process for modeling surface-water flow with the US Geological Survey modular ground-water model (MODFLOW-2005), US Geological Survey Techniques and Methods (2012) 6–A40. Cited in page 37.

[68] H. Hardelauf, M. Javaux, M. Herbst, S. Gottschalk, R. Kasteel, J. Vanderborght, H. Vereecken, PARSWMS: A parallelized model for simulating three-dimensional water flow and solute transport in variably saturated soils, Vadose Zone Journal 6 (2) (2007) 255–259. Cited in pages 37, 39, 40, 45, 63, 71, 73, and 136.

[69] W. Wang, T. Fischer, B. Zehner, N. Böttcher, U.-J. Görke, O. Kolditz, A parallel finite element method for two-phase flow processes in porous media: Opengeosys with petsc, Environmental Earth Sciences 73 (5) (2015) 2269–2285. Cited in pages 37 and 129.

[70] R. T. Mills, C. Lu, P. C. Lichtner, G. E. Hammond, Simulating subsurface flow and transport on ultrascale computers using PFLOTRAN, in:

Journal of physics: conference series, Vol. 78, IOP Publishing, 2007, p. 012051. Cited in page 37.

[71] R. M. Maxwell, M. Putti, S. Meyerhoff, J.-O. Delfs, I. M. Ferguson, V. Ivanov, J. Kim, O. Kolditz, S. J. Kollet, M. Kumar, et al., Surface-subsurface model intercomparison: A first set of benchmark results to diagnose integrated hydrology and feedbacks, Water Resour. Res. 50 (2) (2014) 1531–1549. Cited in pages 38 and 72.

[72] S. Kollet, M. Sulis, R. M. Maxwell, C. Paniconi, M. Putti, G. Bertoldi, E. T. Coon, E. Cordano, S. Endrizzi, E. Kikinzon, et al., The integrated hydrologic model intercomparison project, IH-MIP2: A second set of benchmark results to diagnose integrated hydrology and feedbacks, Water Resources Research 53 (1) (2017) 867–890. Cited in page 38.

[73] P. C. Liu, D. S. Shih, C.-Y. Chou, C.-H. Chen, Y.-C. Wang, Development of a parallel computing watershed model for flood forecasts, Procedia Engineering 154 (2016) 1043–1049. Cited in pages 39, 40, and 136.

[74] J.-R. C. Cheng, H.-C. Lin, H.-P. Cheng, R. M. Hunter, D. R. Richards, G.-T. Yeh, Parallelization of the WASH123D code—phase I: 2-dimensional overland and 3-dimensional subsurface flows, in: Developments in Water Science, Vol. 55, Elsevier, 2004, pp. 1403–1414. Cited in pages 39, 40, and 136.

[75] T. Cheng, Z. Mo, J. Shao, Accelerating groundwater flow simulation in MODFLOW using JASMIN-based parallel computing, Groundwater 52 (2) (2014) 194–205. Cited in pages 39 and 40.

[76] Y. Dong, G. Li, A parallel PCG solver for MODFLOW, Groundwater 47 (6) (2009) 845–850. Cited in pages 39 and 40.

[77] X. Ji, D. Li, T. Cheng, X.-S. Wang, Q. Wang, Parallelization of MODFLOW using a GPU library, Groundwater 52 (4) (2014) 618–623. Cited in pages 39 and 40.

[78] Q. fei ZHANG, S. qi LAN, Y. ming WANG, Y. fu XU, A new numerical method for groundwater fiow and solute transport using velocity field, Journal of Hydrodynamics, Ser. B 20 (3) (2008) 356–364. Cited in page 45.

[79] L. Lin, J.-Z. Yang, B. Zhang, Y. Zhu, A simplified numerical model of 3-D groundwater and solute transport at large scale area, Journal of Hydrodynamics 22 (3) (2010) 319–328. Cited in page 45.

[80] P. A. Lott, H. F. Walker, C. S. Woodward, U. M. Yang, An accelerated Picard method for nonlinear systems related to variably saturated flow, Advances in Water Resources 38 (C) (2012) 92–101. Cited in page 45.

[81] M. Herbst, S. Gottschalk, On preconditioning for a parallel solution of the Richards equation, Computers & geosciences 34 (12) (2008) 1958–1963. Cited in pages 45 and 71.

[82] E. Hanert, V. Legat, How to save a bad element with weak boundary conditions, Computers & Fluids 35 (5) (2006) 477–484. Cited in pages 50 and 79.

[83] Y. Bazilevs, T. Hughes, Weak imposition of Dirichlet boundary conditions in fluid mechanics, Computers & Fluids 36 (1) (2007) 12–26. Cited in pages 50 and 79.

[84] K. Shahbazi, An explicit expression for the penalty parameter of the interior penalty method, Journal of Computational Physics 205 (2) (2005) 401–407. Cited in page 50.

[85] B. Cockburn, C. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, Journal of Scientific Computing 16 (3) (2001) 173–261. Cited in page 55.

[86] V. Aizinger, A geometry independent slope limiter for the discontinuous Galerkin method, Computational Science and High Performance Computing IV (2011) 207–217. Cited in page 55.

[87] J. Simunek, M. Sejna, M. V. Genuchten, 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 240. Cited in page 57.

[88] T. Vogel, M. Cislerova, On the reliability of unsaturated hydraulic conductivity calculated from the moisture retention curve, Transport in porous media 3 (1988) 1–15. Cited in pages 57 and 142.

[89] H. Berninger, R. Kornhuber, O. Sander, Fast and robust numerical solution of the richards equation in homogeneous soil, SIAM Journal on Numerical Analysis 49 (6) (2011) 2576–2597. Cited in page 67.

[90] E. Constantinescu, A. Sandu, Multirate timestepping methods for hyperbolic conservation laws, Journal of Scientific computing 33 (3) (2007) 239–278. Cited in page 67.

[91] C. Paniconi, E. Wood, A detailed model for simulation of catchment scale subsurface hydrologic processes, Water Resources Research 29 (6) (1993) 1601–1620. Cited in pages 70 and 71.

[92] G. Salvucci, D. Entekhabi, Hillslope and climatic controls on hydrologic fluxes, Water Resources Research 31 (7) (1995) 1725–1739. Cited in page 70.

[93] Z. Liu, E. Todini, Towards a comprehensive physically-based rainfall-runoff model, Hydrology and Earth System Sciences 6 (5) (1999) 859–881. Cited in page 71.

[94] M. Santillana, C. Dawson, A local discontinuous Galerkin method for a doubly nonlinear diffusion equation arising in shallow water modeling, Computer Methods in Applied Mechanics and Engineering 199 (23-24) (2010) 1424–1436. Cited in page 71.

[95] P. D. Giammarco, E. Todini, P. Lamberti, A conservative finite elements approach to overland flow: The control volume finite element formulation, Journal of Hydrology 175 (1) (1996) 267–291. Cited in pages 71, 80, and 83.

[96] J. VanderKwaak, K. Loague, Hydrologic-response simulations for the R-5 catchment with a comprehensive physics-based model, Water Resources Research 37 (4) (2001) 999–1013. Cited in pages 71, 72, and 90.

[97] G. Chavent, J. Roberts, A unified physical presentation of mixed, mixed-hybrid finite elements and usual finite differences for the determination of velocities in waterflow problems, Institut National de Recherche en Informatique et en Automatique (1989) 52. Cited in page 71.

[98] R. Freeze, Three-dimensional, transient, saturated-unsaturated flow in a groundwater basin, Water Resources Research 7 (2) (1971) 347–366. Cited in page 71.

[99] J. H. Fleckenstein, S. Krause, D. M. Hannah, F. Boano, Groundwater-surface water interactions: New methods and models to improve understanding of processes and dynamics, Advances in Water Resources 33 (11) (2010) 1291–1295. Cited in page 72.

[100] B. A. Ebel, B. B. Mirus, C. S. Heppner, J. E. Vanderkwaak, K. Loague, First-order exchange coefficient coupling for simulating surface water-groundwater interactions: parameter sensitivity and consistency with a physics-based approach, Hydrological Processes 23 (13) (2009) 1949–1959. Cited in pages 72, 76, and 90.

[101] J. E. Liggett, A. D. Werner, C. T. Simmons, Influence of the first-order exchange coefficient on simulation of coupled surface-subsurface flow, Journal of Hydrology 414-415 (C) (2012) 503–515. Cited in page 72.

[102] M. V. Genuchten, A closed-form equation for predicting the hydraulic conductivity of unsaturated soils, Soil Science Society of America Journal 44 (5) (1980) 892–898. Cited in pages 74 and 104.

[103] Y. Mualem, A new model for predicting the hydraulic conductivity of unsaturated porous media, Water Resources Research 12 (3) (1976) 513–522. Cited in pages 74 and 104.

[104] G. Gottardi, M. Venutelli, An accurate time integration method for simplified overland flow models, Advances in Water Resources 31 (1) (2008) 173–180. Cited in pages 75 and 85.

[105] Y. Saad, M. H. Schultz, Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on scientific and statistical computing 7 (3) (1986) 856—869. Cited in page 81.

[106] D. Tian, D. Liu, A new integrated surface and subsurface flows model and its verification, Applied Mathematical Modelling 35 (7) (2011) 3574–3586. Cited in pages 83 and 87.

[107] A. Abdul, R. Gillham, Laboratory studies of the effects of the capillary fringe on streamflow generation, Water Resources Research 20 (6) (1984) 691–698. Cited in pages 83 and 85.

[108] H. Cloke, M. Anderson, J. McDonnell, J. Renaud, Using numerical modelling to evaluate the capillary fringe groundwater ridging hypothesis of streamflow generation, Journal of Hydrology 316 (1) (2006) 141–162. Cited in page 83.

[109] A. S. Abdul, Experimental and numerical studies of the effect of the capillary fringe on streamflow generation, Ph.D. Thesis, University of Waterloo, Waterloo, Ontario (1985) 210. Cited in page 88.

[110] S. Mehl, M. Hill, Grid-size dependence of Cauchy boundary conditions used to simulate stream–aquifer interactions, Advances in Water Resources 33 (4) (2010) 430–442. Cited in page 95.

[111] W. Lai, F. L. Ogden, A mass-conservative finite volume predictor-corrector solution of the 1D Richards' equation, Journal of Hydrology 523 (2015) 119–127. Cited in pages 102, 108, 109, and 125.

[112] F. L. Ogden, M. B. Allen, W. Lai, J. Zhu, M. Seo, C. C. Douglas, C. A. Talbot, The soil moisture velocity equation, Journal of Advances in Modeling Earth Systems 9 (2) (2017) 1473–1487. Cited in page 109.

[113] X. He, L. Ren, An adaptive multiscale finite element method for unsaturated flow problems in heterogeneous porous media, Journal of Hydrology 374 (1-2) (2009) 56–70. Cited in page 110.

[114] J. Vanderborght, R. Kasteel, M. Herbst, M. Javaux, D. Thiery, M. Vanclooster, C. Mouvet, H. Vereecken, A set of analytical benchmarks to test numerical models of flow and transport in soils, Vadose Zone Journal 4 (1) (2005) 206–221. Cited in pages 113, 114, 116, and 146.

[115] C. Dages, M. Voltz, J. Lacas, O. Huttel, S. Negro, X. Louchart, An experimental study of water table recharge by seepage losses from a ditch with intermittent flow, Hydrological Processes 22 (18) (2008) 3555–3563. Cited in pages 113, 117, and 119.

[116] J. Jones, E. Sudicky, R. McLaren, Application of a fully-integrated surface-subsurface flow model at the watershed-scale: A case study, Water Resources Research 44 (3) (2008) W03407. Cited in page 117.

[117] C. Dages, A. Samouëlian, S. Negro, V. Storck, O. Huttel, M. Voltz, Seepage patterns of diuron in a ditch bed during a sequence of flood events, Science of the Total Environment 537 (2015) 120–128. Cited in page 119.

[118] J. Dollinger, C. Dagès, A. Samouelian, G. Coulouma, M. Lanoix, Y. Blanca, M. Voltz, Contrasting soil property patterns between ditch bed and neighbouring field profiles evidence the need of specific approaches when assessing water and pesticide fate in farmed landscapes, Geoderma 309 (2018) 50–59. Cited in page 119.

[119] J. Simunek, K. Huang, M. V. Genuchten, The SWMS 3D code for simulating water flow-and solute transport in three-dimensional variably-saturated media, User Guide. Cited in page 119.

[120] H. Gerke, M. V. Genuchten, A dual-porosity model for simulating the preferential movement of water and solutes in structured porous media, Water resources research 29 (2) (1993) 305–319. Cited in page 121.

[121] M. Kuráž, P. Mayer, M. Lepš, D. Trpkošová, An adaptive time discretization of the classical and the dual porosity model of Richards' equation, Journal of computational and applied mathematics 233 (12) (2010) 3167–3177. Cited in page 121.

[122] M. G. Trefry, C. Muffels, Feflow: A finite-element ground water flow and transport modeling tool, Ground Water 45 (5) (2007) 525–528. Cited in page 121.

[123] P. Brunner, C. T. Simmons, Hydrogeosphere: a fully integrated, physically based hydrological model, Groundwater 50 (2) (2012) 170–176. Cited in page 121.

[124] A. H. Baker, R. D. Falgout, T. Gamblin, T. V. Kolev, M. Schulz, U. M. Yang, Scaling algebraic multigrid solvers: On the road to exascale, in: Competence in High Performance Computing 2010, Springer, 2011, pp. 215–226. Cited in page 129.

[125] S. J. Kollet, R. M. Maxwell, C. S. Woodward, S. Smith, J. Vanderborght, H. Vereecken, C. Simmer, Proof of concept of regional scale hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources, Water Resources Research 46 (4). Cited in page 137.