https://doi.org/10.35662/unine-thesis-3190



# A Custom OpenFOAM Solver for Fully-Coupled Surface and Subsurface Hydrological Modelling

Thesis presented to the Faculty of Science of the University of Neuchâtel to satisfy the requirements of the degree of Doctor of Sciences by

ÁLVARO PARDO ÁLVAREZ

Accepted on the recommendation of:

Prof. Dr. Philip Brunner, University of Neuchâtel, Switzerland (Thesis Director)
Prof. Dr. Jan H. Fleckenstein, UFZ, Germany (Thesis Co-Director, Rapporteur)
Prof. Dr. Daniel Hunkeler, University of Neuchâtel, Switzerland (Rapporteur)
Prof. Dr. René Therrien, Laval University, Canada (Rapporteur)

**Defended:** 26th May 2025

unine
Université de Neuchâtel
Faculté des sciences

Rue Emile-Argand 11
CH-2000 Neuchâtel
doctorat.sciences@unine.ch

# IMPRIMATUR POUR THESE DE DOCTORAT

**La Faculté des sciences de l'Université de Neuchâtel autorise l'impression de la présente thèse soutenue par**

## Monsieur Álvaro PARDO ÁLVAREZ

Titre :

## "A Custom OpenFOAM Solver for Fully-Coupled Surface and Subsurface Hydrological Modelling"

**sur le rapport des membres du jury composé comme suit :**

- **Prof. Philip Brunner,** directeur de thèse, Université de Neuchâtel, Suisse
- **Prof. Jan Fleckenstein,** co-directeur de thèse, Helmholtz-Zentrum für Umweltforschung, Allemagne
- **Prof. Daniel Hunkeler,** Université de Neuchâtel, Suisse
- **Prof. René Therrien,** Université Laval, Canada

Neuchâtel, le 3 juillet 2025

Le Doyen, Prof. P. Brunner

**Citation:** Pardo-Álvarez, Á. (2025), A Custom OpenFOAM Solver for Fully-Coupled Surface and Subsurface Hydrological Modelling, Doctoral Thesis, University of Neuchâtel, Switzerland.

# Abstract

Numerical models play a crucial role in studying surface water-groundwater (SW-GW) interactions. They enable the resolution of complex hydrodynamics and associated processes, such as solute transport and heat transfer, that occur across the sediment-water interface under varying forcing conditions. While traditional modelling approaches treated SW and GW flows separately or accounted for the interfacial feedback in a unidirectional manner, advances in hydrological science and computational power have enabled the development of more sophisticated techniques that better represent the bidirectional nature of SW-GW exchanges. These include fully-coupled models (FCMs) and fully-integrated models (FIMs). FIMs solve both surface and subsurface processes within a single computational framework using a unified suite of equations, whereas FCMs use distinct, domain-specific equations and simulate their interaction through the iterative exchange of boundary conditions across their interface. Among the equations commonly used by FCMs to simulate surface flow, the Navier-Stokes equations enable the resolution of the complex three-dimensional (3D) turbulent dynamics and, therefore, allow the impact of turbulence on the SW-GW exchange to be accounted for. These equations are solved by Computational Fluid Dynamics (CFD) software, such as **OpenFOAM**. Beyond its standard CFD applications, the modular and extensible structure of this powerful, open-source C++ toolbox makes **OpenFOAM** an ideal framework for implementing user-defined FCMs. Leveraging this flexibility, the present PhD thesis develops `darcyInterTransportFoam`, a novel, fully-coupled 3D SW-GW model. The new solver builds upon `hyporheicScalarInterFoam`, addressing key modelling limitations in parameter heterogeneity, boundary conditions and transport processes, among others, that previously constrained its applicability to real-world hydrogeological settings. These improvements are achieved through newly implemented simulation features, including internal solver updates and external add-ons, which expand the model's capabilities for improved SW-GW exchange simulations. A complete description of the modelling novelties, along with an evaluation of their performance, is provided in this thesis. The latter is conducted in two steps: (1) simulating the reach-scale flow, solute transport and heat transfer processes at two different river segments of an actual alluvial system; and (2) comparing the computational demands, efficiency and simulation performance of `darcyInterTransportFoam` with those of `interFoam` (standard **OpenFOAM** FIM) by simulating the flow and solute feedback across a 3D, synthetic rippled streambed. By improving the representation of SW-GW interactions, this work contributes to advancing numerical modelling techniques and their application to hydrological and environmental research.



**Keywords:** surface water-groundwater interactions, fully-coupled modelling, fully-integrated modelling, computational fluid dynamics, conservative solute transport, heat transfer, OpenFOAM.

# Résumé

Les modèles numériques jouent un rôle essentiel dans l'étude des interactions entre les eaux de surface (SW) et les eaux souterraines (GW). Ils permettent de résoudre les hydrodynamiques complexes et les processus associés, comme le transport de solutés et le transfert de chaleur, qui se produisent à l'interface sédiment-eau sous diverses conditions de forçage. Alors que les approches traditionnelles traitaient séparément les écoulements de surface et souterrains, ou ne considéraient que des rétroactions unidirectionnelles, les avancées en sciences hydrologiques et en puissance de calcul ont conduit au développement de techniques plus sophistiquées, capables de mieux représenter la nature bidirectionnelle des échanges SW-GW. Ces techniques incluent les modèles entièrement couplés (FCMs) et les modèles entièrement intégrés (FIMs). Les FIMs résolvent les processus de surface et de subsurface dans un même cadre de calcul via un ensemble unifié d'équations, tandis que les FCMs utilisent des équations spécifiques à chaque domaine et simulent leur interaction par échange itératif de conditions aux limites. Pour simuler l'écoulement de surface, les FCMs recourent souvent aux équations de Navier-Stokes, qui permettent de représenter les dynamiques turbulentes tridimensionnelles (3D) complexes et, ainsi, de tenir compte de leur impact sur les échanges SW-GW. Ces équations sont résolues à l'aide de logiciels de dynamique des fluides numérique, tels qu'**OpenFOAM**. Grâce à sa structure modulaire et extensible, cette puissante boîte à outils C++ open-source constitue un cadre idéal pour développer des FCMs personnalisés. En exploitant cette flexibilité, cette thèse développe `darcyInterTransportFoam`, un nouveau modèle SW-GW 3D entièrement couplé. Il repose sur `hyporheicScalarInterFoam`, en corrigeant plusieurs limitations liées à l'hétérogénéité des paramètres, aux conditions aux limites et aux processus de transport, qui limitaient son application à des contextes réels. Ces améliorations sont rendues possibles par de nouvelles fonctionnalités, incluant des mises à jour internes et des modules externes, qui élargissent les capacités du modèle. Une description complète des nouveautés, ainsi qu'une évaluation de leurs performances, est présentée. Cette évaluation se fait en deux étapes : (1) la simulation des processus d'écoulement, de transport de solutés et de transfert de chaleur à l'échelle d'un tronçon de rivière dans deux segments différents d'un système alluvial réel; et (2) la comparaison des performances numériques et de la précision de `darcyInterTransportFoam` avec celles d'`interFoam` (FIM standard d'**OpenFOAM**), via la simulation des rétroactions écoulement-soluté sur un lit de rivière ondulé, synthétique et 3D. En améliorant la représentation des interactions SW-GW, ce travail contribue à l'avancement de la modélisation numérique pour la recherche hydrologique et environnementale.

**Mots-clés :** interactions eaux de surface-eaux souterraines, modélisation entièrement couplée, modélisation entièrement intégrée, dynamique des fluides numérique, transport de soluté conservatif, transfert de chaleur, OpenFOAM.

# Acknowledgements

This thesis has come to an end after six long years. Different reasons and situations made things take longer than initially expected. However, the job is now done, and this would not have been possible without the assistance of many different people around, which supported and encouraged me throughout all the process. First and foremost, I would like to thank my supervisor, Prof. Dr. Philip Brunner, for the trust you placed in me to proceed independently for the most part and the expertise you provided whenever necessary. The time spent together, and discussions we held throughout these years, encouraged me to keep going and do not give up in difficult times. Having satisfactorily concluded this thesis, it is not only my success, but also yours. I dare say that we both have learnt a lot from this period working together, and that such knowledge will help us in the future to confront similar situations. I am also grateful to Roberto Costa, who assisted me with fieldwork. Your patient guidance was indispensable every time I was in the field. Likewise, I would like to acknowledge the support from the people at the UFZ Leipzig, namely, my co-supervisor Prof. Dr. Jan H. Fleckenstein, Dr. Nico Trauth and Franz Dichgans. Jan, you were always willing to help and contribute as much as you could when requested. Your comments, sometimes overwhelming, provided me with new insights and ideas, and helped me to improve my work. Nico, everything started working together with you. You were the person who introduced me into the OpenFOAM world, my mentor at the very beginning. Franz, I am thankful to have met you at that OpenFOAM course. Our discussions and your constant input helped me to fix issues, solve doubts and ultimately improve my work. I cannot thank you enough for your contribution. I would neither like to forget to express my gratitude for the research network I was part of during my PhD, namely, the ENIGMA ITN, which also provided the financial support for this work through the EU Framework Programme for Research and Innovation Horizon 2020 (Grant No. 722028). Finally, I would like to extend my deepest thanks to Bea, who shared this whole journey with me and who also suffered my ups and downs during all these years (grazas por sosterme todos estes anos e, sobre todo, por escoitarme e apoiarme nos momentos que máis o precisaba. Son moi afortunado de terte na miña vida), to my parents and brother Héctor (gracias por acompañarme en todas las adversidades de estos años, por siempre confiar en mí y por celebrar incondicionalmente mis éxitos), as well as to my friends. They have supported me through the process, help me to think out of the box in difficult moments and to forget about work when necessary. I am lucky to have you all around me. This thesis is, therefore, also partially yours.

# Table of Contents

# List of Figures

# List of Tables

# 1

# Introduction

# 1.1. Context and Motivation

## 1.1.1. Surface Water-Groundwater Interactions

Surface water-groundwater (SW-GW) interactions are fundamental components of the hydrological cycle (Krause, et al., 2023; Lewandowski, et al., 2020). Although for a long-time surface water and groundwater were studied as separate entities, it is now widely recognized that these systems are intrinsically linked due to their continuous interaction (Winter, et al., 1998). Consequently, surface water and groundwater are considered a single hydrologic resource connected through the sediment-water interface (Schmidt & Fleckenstein, 2024; Toran, 2019).

SW-GW interactions occur across a wide range of spatial and temporal scales, influencing the availability and quality of water resources, ecosystem health and climate resilience (Boano, et al., 2014; Krause, et al., 2017; Lewandowski, et al., 2019). Their understanding is thus essential not only for managing water resources effectively, but also for the sustainability of ecosystems (Krause, et al., 2024; Nisbeth, et al., 2019; Steiness, et al., 2019). For example, river base flow constitutes groundwater discharge (Kelly, et al., 2019; Le Lay, et al., 2019). Without this critical contribution, streams and rivers would dry up during dry periods, leading to the loss of aquatic habitats and a decline in biodiversity (Cochand, et al., 2019; Poff & Zimmerman, 2010). Similarly, groundwater recharge often depends on the focused infiltration of surface water from rivers or lakes, which ensures the long-term availability of groundwater resources and turnover capacity (Han, et al., 2019; Lewandowski, et al., 2024; Mojarrad, et al., 2019; Trauth, et al., 2018). Intact wetland hydrodynamics, which are crucial for maintaining biodiversity and key ecosystem functions, retaining and trapping pollutants and providing habitat for a wide variety of species, also rely on the interplay between surface water inputs and groundwater flows (Harvey, et al., 2019; Neff, et al., 2020). The disruption of SW-GW exchanges, whether through excessive groundwater extraction, land use changes or climate change, can therefore lead to significant ecological consequences, including habitat degradation, reduced water quality and the loss of biodiversity (Blöschl, et al., 2019; Harvey & Gooseff, 2015; Meinikmann, et al., 2015; Morén, et al., 2017; Watts, 2024).

Over the past few decades, significant progress in understanding the importance of SW-GW interactions and their integrity as a hydrological continuum has been driven by advances in field observation techniques and numerical models (Brunner, et al., 2017; Fleckenstein, et al., 2010; Lewandowski, et al., 2020). In this regard, recent technological innovations have greatly enhanced our ability to observe and quantify SW-GW interactions at unprecedented spatial and temporal resolutions (Coluccio & Morgan, 2019; Hermans, et al., 2023; Krause, et al., 2023). High-resolution remote sensing

techniques, such as photogrammetry, LiDAR and thermal infrared imagery, have enabled the detailed mapping of surface water bodies and their interactions with underlying aquifers (Dugdale, 2016). These technologies have been particularly effective in identifying areas of groundwater discharge and recharge by detecting temperature anomalies or changes in the surface water level (e.g., Harvey, et al., 2019; Lewandowski, et al., 2013; Tang, et al., 2018). Advancements in geophysical methods, such as electrical resistivity tomography (ERT) and ground-penetrating radar (GPR), have provided new insights into the subsurface structures that control groundwater flow and its interaction with surface water (Binley, et al., 2015). These methods enable non-invasive, high-resolution imaging of subsurface features, revealing critical information about the aquifer properties and the connectivity between SW and GW (e.g., Blazevic, et al., 2020; Hermans, et al., 2015; McGarr, et al., 2021). Tracer tests, both natural and artificial, have also become more sophisticated, offering detailed information on the water movement between surface and subsurface domains (Brennwald, et al., 2022; Jasechko, 2019; Warden, et al., 2023). These techniques help to identify the flow paths and quantify both the mixing ratios and residence times of water, thereby enhancing the understanding of the complex processes governing SW-GW interactions (e.g., Blanc, et al., 2024; Knapp & Cirpka, 2017; Nogueira, et al., 2021a; Peel, et al., 2022; Popp, et al., 2021; Schilling, et al., 2017a). Likewise, the implementation of more direct and classic measuring methods, such as hydraulic head monitoring, seepage meters or distributed temperature sensing (DTS), have also experienced advances that allow for a better determination of the exchange between SW and GW (e.g., Le Lay, et al., 2019; Simon, et al., 2022; Tirado-Conde, et al., 2019).

Parallel to field observation advances, numerical models have evolved to integrate increasingly complex physical processes, allowing for more accurate and detailed predictions of water flow and solute transport in both surface and subsurface environments (Barthel & Banzhaf, 2016; Brunner, et al., 2017). In this way, numerical simulations can now account for the complexity of real-world conditions, such as the subsurface heterogeneity, the dynamic nature of surface water bodies or the influence of climatic variability, vegetation and land use changes on SW-GW interactions (e.g., Mahmood, et al., 2019; Nogueira, et al., 2022; Nogueira, et al., 2024; Schilling, et al., 2021; Schilling, et al., 2022; Tang, et al., 2017). Moreover, they can handle fine spatial resolutions and time steps, capturing the SW-GW exchange dynamics more effectively. This has been made possible by advances in computational power and numerical techniques, such as adaptive mesh refinement and parallel computing, which enable the simulation of large, complex systems over extended periods (Anderson, et al., 2015; Maxwell, et al., 2015). Furthermore, the calibration and validation of numerical models have also experienced substantial improvements in recent years due to the progressive incorporation of different remote sensing and field observation data types to the simulations (Schilling, et al., 2019). Techniques, such as data assimilation and highly parametrized regularized inversion (Fletcher, 2022; White, et al., 2020), allow to reduce

the predictive uncertainty of the simulations by simultaneously and systematically calibrating hundreds of parameters according to the provided observations (e.g., Delottier, et al., 2023; Li, et al., 2023; Schilling, et al., 2022; Soltani, et al., 2024; Tang, et al., 2024).

Numerical models have proven to be a suitable complement to field experiments to investigate the complex SW-GW interaction dynamics produced across the sediment-water interface (Hermans, et al., 2023; Lewandowski, et al., 2020). In contrast to in-situ studies, numerical models can provide high-resolution spatiotemporal information on flow and transport processes at a reasonable computational cost, allowing for a broader range of scenario analyses and future projections (Sobhi Gollo, et al., 2022). Moreover, they also allow to determine variables that are hard to measure with field observation techniques. This fact is particularly important for groundwater, where high-resolution data measurements are difficult to achieve (Broecker, et al., 2021). Different modelling approaches have been developed over the past few decades to study the interactions between SW and GW (Fleckenstein, et al., 2010). The differences between modelling techniques primarily lie in the degree of connectivity between the surface and subsurface processes across the SW-GW interface and the specific equations applied in each domain. A further description of each numerical modelling approach is provided in the following subsection (**Section 1.1.2**).

### 1.1.2. Preceding Model Developments

Numerical models are instrumental in identifying the influence of factors like climate change, land use and vegetation on SW-GW exchange, as well as in forecasting future needs for water resource management and environmental planning (Anderson, et al., 2015; Boano, et al., 2014; Brunner, et al., 2017). Their application to SW-GW processes has been extensively reported in the literature in recent years, ranging from catchment scale studies (e.g., Barthel & Banzhaf, 2016; Maxwell & Condon, 2016), to reach scale investigations (e.g., Broecker, et al., 2021; Chen, et al., 2018; Trauth, et al., 2015) and Darcy scale research (e.g., Carrillo, et al., 2020; Soulaine & Tchelepi, 2016).

Numerical models describe the different SW-GW processes through sets of partial differential equations. To determine flow and transport, these equations are transformed into algebraic equations via different spatial and temporal discretization schemes (Versteeg & Malalasekera, 2007). The discretized equations result in a sparse matrix equation, which is solved using iterative or direct linear solvers (Schulze & Thorenz, 2014), leading to an approximate solution that accurately reflects real-world conditions while ensuring computational efficiency.

Numerical models can simulate SW-GW exchanges either independently or in combination with other models through various techniques. These include one-way and

two-way coupling methods, as well as fully-integrated approaches, where a single model is sufficient to represent all processes (Maxwell, et al., 2014). Selecting the most suitable technique can be a challenging and sometimes confusing task. The appropriate choice will largely depend on the spatial and temporal scales of the studied case and the specific hydrodynamic processes to be simulated (Dassargues, 2018). Each technique has its own strengths and limitations; thus, aligning the modelling approach with the objectives of the simulation is crucial for ensuring accurate and reliable results (Anderson, et al., 2015).

The asynchronous linking, or one-way, coupling method (Maxwell, et al., 2014) is the most common modelling technique applied to study SW-GW interactions (e.g., Chen, et al., 2018; Jin, et al., 2011; Lee, et al., 2020; Sobhi Gollo, et al., 2022; Trauth, et al., 2015). In this approach, a surface flow model is coupled to a groundwater flow model, where the pressure distribution at the bottom boundary of the former typically serves as the upper boundary condition for the latter. Once the groundwater fields are solved, there is no feedback to the surface domain, meaning that the computed exchange fluxes do not affect the surface flow dynamics. Despite this one-way approach can often be accepted as a sound approximation of the interactions that occur across the interface, flow and transport exchanges are inherently bidirectional, which implies that there is always feedback from GW to SW and vice versa. Considering the interfacial feedback as unidirectional may result in significant errors in the water balance and solute mass conservation, especially in strongly coupled river-aquifer systems (Li, et al., 2020). To overcome this, fully-integrated models (FIM), such as **`porousInter`** (Oxtoby, et al., 2013), and fully-coupled models (FCM), such as **`HydroGeoSphere`** (Brunner & Simmons, 2012; Therrien, et al., 2010), **`ParFlow`** (Kuffour, et al., 2020) or **`MODFLOW`** (Brunner, et al., 2010; Langevin, et al., 2023), have been developed to simulate the actual two-way exchange between SW and GW. FIMs apply a global implicit coupling approach (Maxwell, et al., 2014), in which a unified set of equations governs flow and transport processes across both domains. In contrast, FCMs solve surface and subsurface processes using different equations, with coupling achieved through a sequential iterative, or two-way, scheme (Maxwell, et al., 2014). This distinction largely arises from the differing temporal variabilities between surface and subsurface variables, with surface variables typically exhibiting greater short-term fluctuations compared to the more gradual changes observed in subsurface environments (Broecker, et al., 2019). Consequently, although FIMs enable the investigation of SW-GW processes within a single computational framework, FCMs can be particularly beneficial when the distinct hydrodynamics of each domain must be precisely modelled. In such cases, these techniques may not only enhance the accuracy of the simulations but also ensure the efficient use of computational resources by applying different control parameters and modelling strategies to each domain (Kollet & Maxwell, 2008).

Building on this framework, FCMs commonly use the two-dimensional (2D) shallow water equations to solve the surface flow dynamics, while the groundwater flow is usually driven by the three-dimensional (3D) Richards' equation and Darcy's law. The effective coupling and, therefore, simultaneous solution of both surface and subsurface equations, is achieved by the two-way, iterative transfer of parameter information through the SW-GW interface. Shallow water equations are derived from depth integration of the Navier-Stokes equations. This derivation implies that they do not directly determine the actual vertical flow velocities but instead infer them from hydrostatic pressure variations. Consequently, the complex 3D turbulent flow dynamics are not directly resolved. Given that turbulence can significantly influence the magnitude of SW-GW exchange (Tonina & Buffington, 2009; Voermans, et al., 2017), its detailed computation should not be overlooked. As an alternative, Computational Fluid Dynamics (CFD) models offer a more precise approach for capturing these complex dynamics by solving the full 3D Navier-Stokes equations. CFD solvers have the potential to account for the complex 3D mechanisms of flow dynamics, including turbulence, and therefore can provide high-resolution information on flow field characteristics. Such comprehensive computation of the flow dynamics ultimately allows to get a better insight into the actual flow and transport exchanges across the sediment-water interface, as well as into the processes occurring in both surface and subsurface domains.

Among the different CFD models available to simulate 3D processes, the **OpenFOAM** software (Weller, et al., 1998) has grown in popularity in recent years due to its open-source nature and extensible code structure (e.g., Broecker, et al., 2021; Lee, et al., 2020; Sobhi Gollo, et al., 2022; Trauth, et al., 2015). This flexibility makes **OpenFOAM** an ideal modelling framework for implementing new numerical applications, such as FCMs, for the thorough study of 3D SW-GW processes. An extended description of this software and the reasons for its selection in this PhD thesis are contained in the following section of this document (**Section 1.2**).

## 1.2. The OpenFOAM Software

**OpenFOAM** (**Open**-Source **F**ield **O**peration **A**nd **M**anipulation) is a powerful, free and open-source C++ library designed to model and analyse continuum mechanics problems, most prominently including Computational Fluid Dynamics (CFD).

Applications of **OpenFOAM** can be found across most areas of engineering and science, from both commercial and academic organisations. Furthermore, its free and open-source nature encourages collaboration and knowledge sharing within its software community (e.g., cfd-online.com). In this way, the possibility to have full access to the code and its updates provides a robust platform for collaborative research. As a result, this collective expertise produces valuable software development and improved

functionality to complement the official **OpenFOAM** releases (scheduled every six months in June and December).

The **OpenFOAM** toolbox contains a comprehensive set of applications for the thorough simulation of diverse two and three-dimensional numerical scenarios (OpenCFD Ltd, 2024). These applications fall into two main categories: numerical *solvers* and pre- and post-processing *utilities*. Whereas the *solvers* are designed to solve complex partial differential equations, the *utilities* are aimed to perform tasks that involve data manipulation. The partial differential equations are discretized based on the Finite Volume Method (FVM) in space and the Finite Difference Method (FDM) in time (Schulze & Thorenz, 2014). The combination of both *solvers* and *utilities*, together with further tools and libraries, allows us to gain insights into physics phenomena, optimize designs, evaluate performance and ultimately make informed decisions on different continuum mechanics problems. The latter include complex fluid flows (incompressible and compressible) involving chemical reactions, turbulence and heat transfer to acoustics, solid mechanics and electromagnetics. Due to the nature of the SW-GW systems, only fluid flows and associated features are subject of analysis in this PhD thesis.

Beyond the standard applications included in the toolbox, the modular and extensible nature of **OpenFOAM** allows to customize and extend the software to suit specific simulation requirements (OpenCFD Ltd, 2023). In this way, new solvers, utilities, boundary conditions and numerical schemes can be created by any user with some pre-requisite knowledge of the underlying method, physics and programming techniques involved. This flexible framework enables **OpenFOAM** to provide advanced simulation capabilities, hardly achievable by other continuum mechanics software.

A short description of the software history and its different variants, the basic directory structure to run a case and the necessary steps to set up a model in **OpenFOAM** can be found in Appendix A at the end of this document.

### 1.2.1. Technical and Economic Benefits, as well as Challenges, of Using OpenFOAM

The growing use of **OpenFOAM** as a numerical tool for investigating SW-GW interactions can be attributed to several advantages over other available codes. The most important are the following:

- **No license costs:** as opposed to commercial software, that requires expensive licensing fees, **OpenFOAM** is free to use due to its GNU General Public License Version 3 (GPLv3). This license guarantees the open sourceness of the software while allowing its private and commercial usage. GPLv3 makes **OpenFOAM** accessible to anyone, ranging from heavy users of the software to private companies and research institutions making occasional use of it. This fact is

especially important for the former. Computational costs can become unacceptable if a High-Performance Computing (HPC) cluster is required to run the jobs in parallel, since additional fees typically need to be paid in order to have access to powerful supercomputers such as this. **OpenFOAM** enables to reduce the total cost of these jobs by eliminating the licensing fees.

- **Open source**: both industry and research often demand customized models to address singular problems. While some commercial software, such as ANSYS **Fluent** or **COMSOL Multiphysics**, do provide ways to implement customizations, they are limited to specific parts of the code where the developer allows the user to add new functionality. **OpenFOAM**, instead, provides full access to the source code, so that any possible customization can be made by a qualified user.
- **Large set of precompiled applications: OpenFOAM** does not have a generic solver applicable to all cases, as it occurs with other numerical tools. Instead, it contains multiple solvers and utilities designed to solve different problems in computational continuum mechanics. This means that the solver selection will depend each time on the class of problem to be addressed. **OpenFOAM** solvers are classified according to the continuum mechanics category. Related to hydrogeology, these include incompressible and multiphase flows with RANS and LES turbulence model capabilities, particle-tracking, or heat transfer, among others. Commercial software is typically limited to one or two of the previous categories. This fact limits its applicability and normally forces the coupling to a second model in order to achieve a complete solution.

Other reasons that could explain the increasing application of **OpenFOAM** to SW-GW studies could be the **unstructured polyhedral 3D mesh**, which offers great freedom in mesh generation and manipulation, in particular when the geometry of the domain is complex or changes over time; the **friendly syntax for partial differential equations**, which facilitates the reading and understanding of the code and encourages the implementation of new applications; or the **automatic parallelization of applications using OpenMPI (Open Message Passing Interface)**, which saves user's time in the computational process.

Despite the aforementioned clear advantages, **OpenFOAM** also presents some disadvantages concerning other numerical models. Probably the biggest one is the lack of the integrated GUI (Graphical User Interface) that many commercial software provides. Furthermore, the documentation is much less plentiful, so the learning curve is significantly flatter, especially for beginners. This latter point could explain the reluctance of the hydrological community to use **OpenFOAM** and gravitate more towards commercial codes that can get up and running faster. However, the simplicity of commercial codes can be sometimes misleading, where solutions can be achieved even with critical errors in the model setup. This does not happen in **OpenFOAM**, as some

previous knowledge of the model setup as well as of the underlying physics and methods involved is required in order to run the software properly.

Ultimately, OpenFOAM provides the hydrogeological community with a powerful computational tool, capable of accurately simulating diverse SW-GW processes and developing customized models, while also offering a cost advantage compared to other commercial software.

## 1.3. State-of-the-Art: Recently Developed Fully-Coupled OpenFOAM Solvers

Numerous SW-GW solvers have been developed in recent years building on OpenFOAM's flexible code structure. Among these, fully-integrated solvers, often based on adaptations of the standard `interFoam` and `pimpleFoam` solvers or custom developments like `porousInter`, are more prevalent. In contrast, fully-coupled solvers remain comparatively rare. One notable example of the latter is `hyporheicFoam` (Li, et al., 2020). This user-defined model simulates flow and biogeochemical reactions in both surface and subsurface domains, along with the two-way exchange occurring across their interface. For surface flow dynamics, the model employs the standard `pimpleFoam` solver, while a custom-coded OpenFOAM solver handles saturated groundwater flow. Full coupling between the surface and subsurface domains is achieved by iteratively mapping conservative flux boundary conditions (BCs) at their interface. However, the use of `pimpleFoam` (transient, incompressible flow solver) implies the assumption of a rigid-lid condition, where the water surface is considered flat (e.g., Chen, et al., 2018; Janssen, et al., 2012; Jin, et al., 2011; Lee, et al., 2020). In reality, river surfaces are deformable and adjust dynamically to topography. Previous studies have demonstrated that neglecting surface deformation can underestimate bedform-driven SW-GW exchange (e.g., Broecker, et al., 2019; Broecker, et al., 2021; Trauth, et al., 2014; Trauth, et al., 2015). To address this limitation, Lee, et al. (2021) developed a fully-coupled 3D model named `hyporheicScalarInterFoam`. This enhanced solver replaces the use of `pimpleFoam` with `interFoam` to simulate surface water dynamics, while retaining the same subsurface flow solver employed in `hyporheicFoam`. The `interFoam` solver (Deshpande, et al., 2012), part of the OpenFOAM suite, handles two-phase, incompressible, immiscible flows using the Volume of Fluid (VOF) method (Hirt & Nichols, 1981) to capture the interface between fluids. This approach enables the model to account for deformations of the river's free surface, thus improving the accuracy of surface water-groundwater exchange calculations. Unlike `hyporheicFoam`, however, `hyporheicScalarInterFoam` is limited to conservative solute transport in both domains.

Full coupling is maintained through an iterative scheme that transfers model-specific flux boundary conditions across the surface–subsurface interface.

Although `hyporheicScalarInterFoam` demonstrates strong potential for simulating SW-GW flow and transport dynamics, its application to real-world hydrogeological settings remains limited due to several modelling constraints:

1. The solver does not directly compute hydrological variables such as specific discharge or hydraulic head. Instead, it resolves the flow dynamics using the so-called modified pressure and the volume-averaged velocity.
2. It does not support the specification of water depth at the outlet of the surface domain, making it impossible to apply a hydrostatic pressure BC that would allow free outflow.
3. The body force required for compatibility with periodic boundary conditions is included only in the computation of the specific discharge, while the hydraulic head and groundwater flux calculations omit this adjustment.
4. Subsurface properties, such as hydraulic conductivity, porosity and dispersivities, must be defined as homogeneous and isotropic across the entire domain, limiting the representation of geological heterogeneity.
5. The model requires full hydraulic connectivity along the sediment-water interface, which must remain entirely submerged for the simulation to run.
6. Solute transport must be solved in both water and air phases within the surface domain, even when unnecessary.
7. Heat transport is not supported in either domain.
8. Domain coupling relies on interpolation at the interface, which may introduce inaccuracies, particularly when mesh resolutions differ significantly.
9. Convergence between surface and subsurface solutions is controlled by a fixed number of iterations, without verification of consistency across domains after each step.
10. The model does not support alternative interface BCs for one-way coupling mode, reducing flexibility in simulation configurations.
11. Data field reinitialization during runtime is not possible and must be handled externally through pre-processing.

Addressing these limitations through further code development will be essential for broadening the solver's capabilities and enabling its application to a wider range of SW-GW interaction scenarios across different spatial and temporal scales.

## 1.4. Scope and Outline of the Present Thesis

### 1.4.1. Aims and Objectives

The main focus of this PhD thesis is **to present an upgraded version of the fully-coupled OpenFOAM solver, `hyporheicScalarInterFoam`, designed to improve its applicability to real-world SW-GW interaction scenarios across a broader range of scales than those captured by the original implementation.** This goal is achieved in three steps: (1) describing the newly-implemented fully-coupled solver and testing its performance through a couple of test cases in a real river-aquifer system (2) comparing and evaluating the computational efficiency and performance of the novel code against a standard fully-integrated **OpenFOAM** solver across a synthetic 3D rippled streambed and (3) bridging the gap between theory and practical application through detailed code descriptions of the new solver features and the external **MATLAB** scripts developed for **OpenFOAM** file generation. All points are thoroughly developed over the different chapters of this thesis. Ultimately, it is hoped that some conclusions can be drawn to the attention of the geoscience community regarding the power and flexibility of **OpenFOAM** in developing advanced numerical tools for the detailed simulation of highly dynamic SW-GW interactions.

### 1.4.2. Workflow and Structure

The current PhD thesis is structured into five chapters. Chapters 1 and 5 comprise, respectively, the general introduction and conclusions, whereas the remaining chapters cover the different subjects addressed in this manuscript. Furthermore, Chapters 2 and 3 are article based, meaning that they follow the structure of a peer-reviewed publication. Each of them contains, therefore, an introduction to the topic, a description of the applied methodology and a separate conclusion. A version of Chapter 2 is in submission to the journal *Geoscientific Model Development* and a publication based on Chapter 3 is currently in preparation.

A brief overview of the topics covered in each article-based chapter is given below:

#### 1.4.2.1. Chapter 2 -- darcyInterTransportFoam v1.0: an Open-Source, Fully-Coupled 3D Solver for Simulating Surface Water – Saturated Groundwater Processes and Exchanges

This chapter introduces an updated and extended version of `hyporheicScalarInterFoam`, a fully-coupled, three-dimensional **OpenFOAM** solver developed by Lee, et al. (2021). The new model, called `darcyInterTransportFoam`,

boosts the applicability of the original code by improving previously existing functionalities and providing novel simulation features. In this way, beyond the original flow and two-phase solute transport simulation, the novel version allows to model, e.g., the transport of solutes in a single phase within the surface domain or the transfer of heat in both domains. Moreover, `darcyInterTransportFoam` implies an evolution in terms of domain parametrization and modelling flexibility. The former is achieved by the possibility to define, e.g., heterogeneous and anisotropic subsurface fields at the subsurface domain. On the other hand, the improved modelling flexibility is supported, among other features, by the existence of reset options to start over the computation, if necessary, at any of the domains. The theoretical framework of the aforementioned novelties, as well as of the rest of new solver features are thoroughly described in this chapter. Furthermore, the efficacy of the new OpenFOAM solver is demonstrated in a couple of test cases, in which the flow, solute transport and head transfer are simulated at two different topographical setups of an alluvial river-aquifer system in Switzerland.

#### 1.4.2.2. Chapter 3 -- A Comparison Study of a Fully-Coupled vs. a Fully-Integrated OpenFOAM Solver for Simulating Surface Water - Groundwater Interactions

In Chapter 3, the previously described fully-coupled model, `darcyInterTransportFoam` (Chapter 2), is compared to `interFoam`, a three-dimensional, transient solver for incompressible, immiscible, two-phase fluids included within the standard OpenFOAM library. `interFoam` allows the simulation of SW-GW interaction dynamics in a fully-integrated way by adding a sink term to the Navier-Stokes equations in the subsurface domain. This resistance term results in a pressure drop proportional to the energy losses caused by the porous medium in the groundwater flow. The implemented comparison aims to address the lack of studies evaluating the suitability of distinct modelling approaches for tackling SW-GW interactions. Both models were used to simulate the continuous feedback between SW and GW across a three-dimensional, synthetic rippled streambed. A brief description of the two solvers as well as details of the comparison set-up are first provided in this chapter. Thereafter, the computational efficiency and performance of both modelling approaches are compared based on the resulting flow and solute transport fields, alongside their computational demands. The results achieved ultimately support the use of numerical models as effective complementary tools to field observations.

#### 1.4.2.3. Chapter 4 -- Code Developments: Solver Upgrades and OpenFOAM Dictionary Generation Scripts

The code upgrades and novel add-ons implemented in `darcyInterTransportFoam` are detailed in this chapter, along with the MATLAB script suite developed to generate the dictionary files required to set up an OpenFOAM simulation case. These descriptions

are supported by code snippets and include the names of all relevant input parameters to enhance clarity and reproducibility, ultimately reinforcing the connection between theory and implementation.

# References


[1] Anderson, M. P., Woessner, W. W., & Hunt, R. J. (2015). *Applied Groundwater Modeling* (Second Edition ed.). Elsevier Inc. https://doi.org/10.1016/C2009-0-21563-7

[2] Barthel, R., & Banzhaf, S. (2016). Groundwater and surface water interaction at the regional-scale – A review with focus on regional integrated models. *Water Resources Management, 30*(1), 1-32. https://doi.org/10.1007/s11269-015-1163-z

[3] Binley, A., Hubbard, S. S., Huisman, J. A., Revil, A., Robinson, D. A., Singha, K., & Slater, L. D. (2015). The emergence of hydrogeophysics for improved understanding of subsurface processes over multiple scales. *Water Resources Research, 51*(6), 3837-3866. https://doi.org/10.1002/2015WR017016

[4] Blanc, T., Peel, M., Brennwald, M. S., Kipfer, R., & Brunner, P. (2024). Efficient injection of gas tracers into rivers: A tool to study Surface water-Groundwater interactions. *Water Research, 254*, 121375. https://doi.org/10.1016/j.watres.2024.121375

[5] Blazevic, L., Bodet, L., Pasquet, S., Linde, N., Jougnot, D., & Longuevergne, L. (2020). Time-Lapse Seismic and Electrical Monitoring of the Vadose Zone during a Controlled Infiltration Experiment at the Ploemeur Hydrological Observatory. *Water, 12*(5), 1230. https://doi.org/10.3390/w12051230

[6] Blöschl, G., Hall, J., Viglione, A., Perdigão, R. A. P., Parajka, J., Merz, B., Lun, D., Arheimer, B., Aronica, G. T., Bilibashi, A., Boháč, M., Bonacci, O., Borga, M., Čanjevac, I., Castellarin, A., Chirico, G. B., Claps, P., Frolova, N., Ganora, D., Gorbachova, L., Gül, A., Hannaford, J., Harrigan, S., Kireeva, M., Kiss, A., Kjeldsen, T. R., Kohnová, S., Koskela, J. J., Ledvinka, O., Macdonald, N., Mavrova-Guirguinova, M., Mediero, L., Merz, R., Molnar, P., Montanari, A., Murphy, C., Osuch, M., Ovcharuk, V., Radevski, I., Salinas, J. L., Sauquet, E., Šraj, M., Szolgay, J., Volpi, E., Wilson, D., Zaimi, K., & Živković, N. (2019). Changing climate both increases and decreases European river floods. *Nature, 573*(7772), 108-111. https://doi.org/10.1038/s41586-019-1495-6

[7] Boano, F., Harvey, J. W., Marion, A., Packman, A. I., Revelli, R., Ridolfi, L., & Wörman, A. (2014). Hyporheic flow and transport processes: Mechanisms, models,

and biogeochemical implications. *Reviews of Geophysics, 41*, 603-679. https://doi.org/10.1002/2012RG000417

[8] Brennwald, M. S., Peel, M., Blanc, T., Tomonaga, Y., Kipfer, R., Brunner, P., & Hunkeler, D. (2022). New Experimental Tools to Use Noble Gases as Artificial Tracers for Groundwater Flow. *Frontiers in Water, 4*. https://doi.org/10.3389/frwa.2022.925294

[9] Broecker, T., Gollo, V. S., Fox, A., Lewandowski, J., Nützmann, G., Arnon, S., & Hinkelmann, R. (2021). High-Resolution Integrated Transport Model for Studying Surface Water-Groundwater Interaction. *Groundwater, 59*(4), 488-502. https://doi.org/10.1111/gwat.13071

[10] Broecker, T., Teuber, K., Sobhi Gollo, V., Nützmann, G., Lewandowski, J., & Hinkelmann, R. (2019). Integral Flow Modelling Approach for Surface Water-Groundwater Interactions along a Rippled Streambed. *Water, 11*(7), 1517. https://doi.org/10.3390/w11071517

[11] Brunner, P., & Simmons, C. T. (2012). HydroGeoSphere: A Fully Integrated, Physically Based Hydrological Model. *Groundwater, 50*(2), 170-176. https://doi.org/10.1111/j.1745-6584.2011.00882.x

[12] Brunner, P., Simmons, C. T., Cook, P. G., & Therrien, R. (2010). Modeling Surface Water-Groundwater Interaction with MODFLOW: Some Considerations. *Groundwater, 48*(2), 174-180. https://doi.org/10.1111/j.1745-6584.2009.00644.x

[13] Brunner, P., Therrien, R., Renard, P., Simmons, C. T., & Hendricks Franssen, H.-J. (2017). Advances in understanding river-groundwater interactions. *Reviews of Geophysics, 55*(3), 818-854. https://doi.org/10.1002/2017RG000556

[14] Carrillo, F. J., Bourg, I. C., & Soulaine, C. (2020). Multiphase flow modeling in multiscale porous media: An open-source micro-continuum approach. *Journal of Computational Physics: X, 8*, 100073. https://doi.org/10.1016/j.jcpx.2020.100073

[15] Chen, X., M. Bayani, C., & Chen, L. (2018). Hyporheic Exchange Driven by Three-Dimensional Sandy Bed Forms: Sensitivity to and Prediction from Bed Form Geometry. *Water Resources Research, 54*(6), 4131-4149. https://doi.org/10.1029/2018WR022663

[16] Cochand, M., Christe, P., Ornstein, P., & Hunkeler, D. (2019). Groundwater storage in high alpine catchments and its contribution to streamflow. *Water Resources Research, 55*(4), 2613-2630. https://doi.org/10.1029/2018WR022989

[17] Coluccio, K., & Morgan, L. K. (2019). A review of methods for measuring groundwater-surface water exchange in braided rivers. *Hydrology and Earth System Sciences, 23*(10), 4397-4417. https://doi.org/10.5194/hess-23-4397-2019

[18] Dassargues, A. (2018). *Hydrogeology: Groundwater Science and Engineering* (First Edition ed.). Boca Raton: Taylor & Francis CRC Press. https://doi.org/10.1201/9780429470660

[19] Delottier, H., Doherty, J., & Brunner, P. (2023). Data space inversion for efficient uncertainty quantification using an integrated surface and sub-surface hydrologic model. *Geoscientific Model Development, 16*(14), 4213–4231. https://doi.org/10.5194/gmd-16-4213-2023

[20] Deshpande, S. S., Anumolu, L., & Trujillo, M. F. (2012). Evaluating the performance of the two-phase flow solver interFoam. *Computational Science & Discovery, 5*(1), 014016. https://doi.org/10.1088/1749-4699/5/1/014016

[21] Dugdale, S. J. (2016). A practitioner's guide to thermal infrared remote sensing of rivers and streams: recent advances, precautions and considerations. *WIREs, 3*(2), 251-268. https://doi.org/10.1002/wat2.1135

[22] Fleckenstein, J. H., Krause, S., Hannah, D. M., & Boano, F. (2010). Groundwater-surface water interactions: New methods and models to improve understanding of processes and dynamics. *Advances in Water Resources, 33*, 1291–1295. https://doi.org/10.1016/j.advwatres.2010.09.011

[23] Fletcher, S. J. (2022). *Data Assimilation for the Geosciences* (2nd ed.). Elsevier.

[24] Han, Z., Shi, X., Jia, K., Sun, B., Zhao, S., & Fu, C. (2019). Determining the discharge and recharge relationships between lake and groundwater in Lake Hulun using hydrogen and oxygen isotopes and chloride iIons. *Water, 11*(2), 264. https://doi.org/10.3390/w11020264

[25] Harvey, J., & Gooseff, M. (2015). River corridor science: Hydrologic exchange and ecological consequences from bedforms to basins. *Water Resources Research, 51*(9), 6893-6922. https://doi.org/10.1002/2015WR017617

[26] Harvey, M. C., Hare, D. K., Hackman, A., Davenport, G., Haynes, A. B., Helton, A., Lane, J. W. Jr., & Briggs, M. A. (2019). Evaluation of stream and wetland restoration using UAS-based thermal infrared mapping. *Water, 11*(8), 1568. https://doi.org/10.3390/w11081568

[27] Hermans, T., Goderniaux, P., Jougnot, D., Fleckenstein, J. H., Brunner, P., Nguyen, F., Linde, N., Huisman, J. A., Bour, O., Lopez Alvis, J., Hoffmann, R., Palacios, A., Cooke, A.-K., Pardo-Álvarez, Á., Blazevic, L., Pouladi, B., Haruzi, P., Fernandez Visentini, A., Nogueira, G. E. H., Tirado-Conde, J., Looms, M. C., Kenshilikova, M., Davy, P., & Le Borgne, T. (2023). Advancing measurements and representations of subsurface heterogeneity and dynamic processes: towards 4D hydrogeology. *Hydrology and Earth System Sciences, 27*(1), 255–287. https://doi.org/10.5194/hess-27-255-2023

[28] Hermans, T., Wildemeersch, S., Jamin, P., Orban, P., Brouyère, S., Dassargues, A., & Nguyen, F. (2015). Quantitative temperature monitoring of a heat tracing experiment using cross-borehole ERT. *Geothermics, 53*, 14-26. https://doi.org/10.1016/j.geothermics.2014.03.013

[29] Hirt, C. W., & Nichols, B. D. (1981). Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics, 39*(1), 201-225. https://doi.org/10.1016/0021-9991(81)90145-5

[30] Janssen, F., Cardenas, M. B., Sawyer, A. H., Dammrich, T., Krietsch, J., & de Beer, D. (2012). A comparative experimental and multiphysics computational fluid dynamics study of coupled surface-subsurface flow in bed forms. *Water Resources Research, 48*(8), W08514. https://doi.org/10.1029/2012WR011982

[31] Jasechko, S. (2019). Global Isotope Hydrogeology¯Review. *Reviews of Geophysics, 57*(3), 835-965. https://doi.org/10.1029/2018RG000627

[32] Jin, G., Tang, H., Li, L., & Barry, D. A. (2011). Hyporheic flow under periodic bed forms influenced by low-density gradients. *Geophysical Research Letters, 38*(22), L22401. https://doi.org/10.1029/2011GL049694

[33] Kelly, L., Kalin, R. M., Bertram, D., Kanjaye, M., Nkhata, M., & Sibande, H. (2019). Quantification of temporal variations in Base Flow Index using sporadic river data: Application to the Bua Catchment, Malawi. *Water, 11*(5), 901. https://doi.org/10.3390/w11050901

[34] Knapp, J. L., & Cirpka, O. A. (2017). Determination of hyporheic travel time distributions and other parameters from concurrent conservative and reactive tracer tests by local-in-global optimization. *Water Resources Research, 53*(6), 4984-5001. https://doi.org/10.1002/2017WR020734

[35] Kollet, S. J., & Maxwell, R. M. (2008). Capturing the influence of groundwater dynamics on land surface processes using an integrated, distributed watershed model. *Water Resources Research, 44*(2). https://doi.org/10.1029/2007WR006004

[36] Krause, S., Abbott, B. W., Baranov, V., Bernal, S., Blaen, P., Datry, T., Drummond, J., Fleckenstein, J. H., Gomez-Vélez, J., Hannah, D. M., Knapp, J. L. A., Kurz, M., Lewandowski, J., Martí, E., Mendoza-Lera, C., Milner, A., Packman, A., Pinay, G., Ward, A. S., & Zarnetzke, J. P. (2024). Organizational principles of hyporheic exchange flow and biogeochemical cycling in river networks across scales. En S. Krause, D. Hannah, & N. Grimm (Eds.), *Ecohydrological Interfaces* (pp. 63-101). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119489702.ch4

[37] Krause, S., Hannah, D., & Grimm, N. (Eds.). (2023). *Ecohydrological Interfaces* (1st ed.). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119489702

[38] Krause, S., Lewandowski, J., Grimm, N. B., Hannah, D. M., Pinay, G., McDonald, K., Martí, E., Argerich, A., Pfister, L., Klaus, J., Battin, T., Larned, S. T., Schelker, J., Fleckenstein, J., Schmidt, C., Rivett, M. O., Watts, G., Sabater, F., Sorolla, A., & Turk, V. (2017). Ecohydrological interfaces as hot spots of ecosystem processes. *Water Resources Research, 53*(8), 6359-6376. https://doi.org/10.1002/2016WR019516

[39] Kuffour, B. N., Engdahl, N. B., Woodward, C. S., Condon, L. E., Kollet, S., & Maxwell, R. M. (2020). Simulating coupled surface–subsurface flows with ParFlow v3.5.0: capabilities, applications, and ongoing development of an open-source, massively parallel, integrated hydrologic model. *Geoscientific Model Development, 13*(3), 1373–1397. https://doi.org/10.5194/gmd-13-1373-2020

[40] Langevin, C. D., Hughes, J. D., Provost, A. M., Russcher, M. J., & Panday, S. (2023). MODFLOW as a Configurable Multi-Model Hydrologic Simulator. *Groundwater, 62*(1), 111-123. https://doi.org/10.1111/gwat.13351

[41] Le Lay, H., Thomas, Z., Rouault, F., Pichelin, P., & Moatar, F. (2019). Characterization of diffuse groundwater inflows into stream water (Part II: Quantifying groundwater inflows by coupling FO-DTS and vertical flow velocities). *Water, 11*(12), 2430. https://doi.org/10.3390/w11122430

[42] Lee, A., Aubeneau, A., & Cardenas, M. (2020). The Sensitivity of Hyporheic Exchange to Fractal Properties of Riverbeds. *Water Resources Research, 56*(5), e2019WR026560. https://doi.org/10.1029/2019WR026560

[43] Lee, A., Aubeneau, A., Liu, X., & Cardenas, M. B. (2021). Hyporheic Exchange in Sand Dunes Under a Freely Deforming River Water Surface. *Water Resources Research, 57*(3), e2020WR028817. https://doi.org/10.1029/2020WR028817

[44] Lewandowski, J., Arnon, S., Banks, E., Batelaan, O., Betterle, A., Broecker, T., Coll, C., Drummond, J. D., Gaona Garcia, J., Galloway, J., Gomez-Velez, J., Grabowski, R. C., Herzog, S. P., Hinkelmann, R., Höhne, A., Hollender, J., Horn, M. A., Jaeger, A., Krause, S., Löchner Prats, A., Magliozzi, C., Meinikmann, K., Mojarrad, B. B., Mueller, B. M., Peralta-Maraver, I., Popp, A. L., Posselt, M., Putschew, A., Radke, M., Raza, M., Riml, J., Robertson, A., Rutere, C., Schaper, J. L., Schirmer, M., Schulz, H., Shanafield, M., Singh, T., Ward, A. S., Wolke, P., Wörman, A., & Wu, L. (2019). Is the Hyporheic Zone Relevant beyond the Scientific Community? *Water, 11*(11), 2230. https://doi.org/10.3390/w11112230

[45] Lewandowski, J., Meinikmann, K., & Krause, S. (2020). Groundwater-Surface Water Interactions: Recent Advances and Interdisciplinary Challenges. *Water, 12*(1), 296. https://doi.org/10.3390/w12010296

[46] Lewandowski, J., Meinikmann, K., Ruhtz, T., Pöschke, F., & Kirillin, G. (2013). Localization of lacustrine groundwater discharge (LGD) by airborne measurement of

thermal infrared radiation. *Remote Sensing of Environment, 138*, 119-125. https://doi.org/10.1016/j.rse.2013.07.005

[47] Lewandowski, J., Rosenberry, D. O., & Meinikmann, K. (2024). Groundwater-Lake interfaces. En S. Krause, D. Hannah, & N. Grimm (Eds.), *Ecohydrological Interfaces* (pp. 103-122). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119489702.ch5

[48] Li, B., Liu, X., Kaufman, M. H., Turetcaia, A., Chen, X., & Cardenas, B. M. (2020). Flexible and Modular Simultaneous Modeling of Flow and Reactive Transport in Rivers and Hyporheic Zones. *Water Resources Research, 56*(2), e2019WR026528. https://doi.org/10.1029/2019WR026528

[49] Li, F., Kurtz, W., Hung, C. P., Vereecken, H., & Hendricks Franssen, H.-J. (2023). Water table depth assimilation in integrated terrestrial system models at the larger catchment scale. *Frontiers in Water, 5*. https://doi.org/10.3389/frwa.2023.1150999

[50] Mahmood, M. N., Schmidt, C., Fleckenstein, J. H., & Trauth, N. (2019). Modeling the impact of stream discharge events on riparian solute dynamics. *Groundwater, 57*(1), 140-152. https://doi.org/10.1111/gwat.12664

[51] Maxwell, R. M., & Condon, L. E. (2016). Connections between groundwater flow and transpiration partitioning. *Science, 353*(6297), 377-380. https://doi.org/10.1126/science.aaf7891

[52] Maxwell, R. M., Condon, L. E., & Kollet, S. J. (2015). A high-resolution simulation of groundwater and surface water over most of the continental US with the integrated hydrologic model ParFlow v3. *Geoscientific Model Development, 8*(3), 923-937. https://doi.org/10.5194/gmd-8-923-2015

[53] Maxwell, R. M., Putti, M., Meyerhoff, S., Delfs, J.-O., Ferguson, I. M., Ivanov, V., Kim, J., Kolditz, O., Kollet, S. J., Kumar, M., Lopez, S., Niu, J., Paniconi, C., Park, Y.-J., Phanikumar, M. S., Shen, C., Sudicky, E. A., & Sulis, M. (2014). Surface-subsurface model intercomparison: A first set of benchmark results to diagnose integrated hydrology and feedbacks. *Water Resources Research*, 50(2), 1531-1549. https://doi.org/10.1002/2013WR013725

[54] McGarr, J. T., Wallace, C. D., Ntarlagiannis, D., Sturmer, D. M., & Soltanian, M. R. (2021). Geophysical mapping of hyporheic processes controlled by sedimentary architecture within compound bar deposits. *Hydrological Processes, 35*(9), e14358. https://doi.org/10.1002/hyp.14358

[55] Meinikmann, K., Hupfer, M., & Lewandowski, J. (2015). Phosphorus in groundwater discharge – A potential source for lake eutrophication. *Journal of Hydrology, 524*, 214-226. https://doi.org/10.1016/j.jhydrol.2015.02.031

[56] Mojarrad, B. B., Betterle, A., Singh, T., Olid, C., & Wörman, A. (2019). The effect of stream discharge on hyporheic exchange. *Water, 11*(7), 1436. https://doi.org/10.3390/w11071436

[57] Morén, I., Worman, A., & Riml, J. (2017). Design of remediation actions for nutrient mitigation in the hyporheic zone. *Water Resources Research, 53*(11), 8872-8899. https://doi.org/10.1002/2016WR020127

[58] Neff, B. P., Rosenberry, D. O., Leibowitz, S. G., Mushet, D. M., Golden, H. E., Rains, M. C., Brooks, J. R., & Lane, C. R. (2020). A hydrologic landscapes perspective on groundwater connectivity of depressional wetlands. *Water, 12*(1), 50. https://doi.org/10.3390/w12010050

[59] Nisbeth, C. S., Jessen, S., Bennike, O., Kidmose, J., & Reitzel, K. (2019). Role of groundwater-borne geogenic phosphorus for the internal P release in shallow lakes. *Water, 11*(9), 1783. https://doi.org/10.3390/w11091783

[60] Nogueira, G. E., Partington, D., Heidbüchel, I., & Fleckenstein, J. H. (2024). Combined effects of geological heterogeneity and discharge events on groundwater and surface water mixing. *Journal of Hydrology, 638*, 131467. https://doi.org/10.1016/j.jhydrol.2024.131467

[61] Nogueira, G. E., Schmidt, C., Partington, D., Brunner, P., & Fleckenstein, J. H. (2022). Spatiotemporal variations in water sources and mixing spots in a riparian zone. *Hydrology and Earth System Sciences, 26*(7), 1883-1905. https://doi.org/10.5194/hess-26-1883-2022

[62] Nogueira, G. E., Schmidt, C., Trauth, N., & Fleckenstein, J. H. (2021a). Seasonal and short-term controls of riparian oxygen dynamics and the implications for redox processes. *Hydrogeological Processes, 35*(2), e14055. https://doi.org/10.1002/hyp.14055

[63] OpenCFD Ltd. (2023, December). *OpenFOAM® v2312 User Guide*. Retrieved December 25, 2024, from https://doc.openfoam.com/2312/

[64] OpenCFD Ltd. (2024, June). OpenFOAM® v2406. OpenFOAM Download [code]. Retrieved from https://dl.openfoam.com/source/v2406/

[65] Oxtoby, O., Heyns, J., & Suliman, R. (2013). A finite-volume solver for two-fluid flow in heterogeneous porous media based on OpenFOAM. Hamburg: 7th Open Source CFD International Conference. https://doi.org/10.13140/2.1.3075.8400

[66] Peel, M., Kipfer, R., Hunkeler, D., & Brunner, P. (2022). Variable 222Rn emanation rates in an alluvial aquifer: Limits on using 222Rn as a tracer of surface water – Groundwater interactions. *Chemical Geology, 599*, 120829. https://doi.org/10.1016/j.chemgeo.2022.120829

[67] Poff, N. L., & Zimmerman, J. K. (2010). Ecological responses to altered flow regimes: a literature review to inform the science and management of environmental flows. *Freshwater Biology, 55*(1), 194-205. https://doi.org/10.1111/j.1365-2427.2009.02272.x

[68] Popp, A. L., Pardo-Álvarez, Á., Schilling, O. S., Scheidegger, A., Musy, S., Peel, M., Brunner, P., Purtschert, R., Hunkeler, D., & Kipfer, R. (2021). A Framework for Untangling Transient Groundwater Mixing and Travel Times. *Water Resources Research, 57*(4), e2020WR028362. https://doi.org/10.1029/2020WR028362

[69] Schilling, O. S., Cook, P. G., & Brunner, P. (2019). Beyond Classical Observations in Hydrogeology: The Advantages of Including Exchange Flux, Temperature, Tracer Concentration, Residence Time, and Soil Moisture Observations in Groundwater Model Calibration. *Reviews of Geophysics, 57*(1), 146-182. https://doi.org/10.1029/2018RG000619

[70] Schilling, O. S., Cook, P. G., Grierson, P. F., Dogramaci, S., & Simmons, C. T. (2021). Controls on Interactions Between Surface Water, Groundwater, and Riverine Vegetation Along Intermittent Rivers and Ephemeral Streams in Arid Regions. *Water Resources Research, 57*(2), e2020WR028429. https://doi.org/10.1029/2020WR028429

[71] Schilling, O. S., Gerber, C., Partington, D. J., Purtschert, R., Brennwald, M. S., Kipfer, R., Hunkeler, D., & Brunner, P. (2017a). Advancing Physically-Based Flow Simulations of Alluvial Systems Through Atmospheric Noble Gases and the Novel 37Ar Tracer Method. *Water Resources Research, 53*(12), 10465-10490. https://doi.org/10.1002/2017WR020754

[72] Schilling, O. S., Partington, D. J., Doherty, J., Kipfer, R., Hunkeler, D., & Brunner, P. (2022). Buried Paleo-Channel Detection With a Groundwater Model, Tracer-Based Observations, and Spatially Varying, Preferred Anisotropy Pilot Point Calibration. *Geophysical Research Letters, 49*(14), e2022GL098944. https://doi.org/10.1029/2022GL098944

[73] Schmidt, C., & Fleckenstein, J. H. (2024). Identifying and Quantifying Water Fluxes at Ecohydrological Interfaces. En S. Krause, D. Hannah, & N. Grimm (Eds.), *Ecohydrological Interfaces* (pp. 149-165). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119489702.ch7

[74] Schulze, L., & Thorenz, C. (2014). The Multiphase Capabilities of the CFD Toolbox OpenFOAM for hydraulic engineering applications. En R. Lehfeldt, & R. Kopmann (Ed.). (pp. 1007-1016). Hamburg: Proceedings of the 11th International Conference on Hydroscience & Engineering. https://doi.org/20.500.11970/99530

[75] Simon, N., Bour, O., Faucheux, M., Lavenant, N., Le Lay, H., Fovet, O., Thomas, Z., & Longuevergne, L. (2022). Combining passive and active distributed temperature

sensing measurements to locate and quantify groundwater discharge variability into a headwater stream. *Hydrology and Earth System Sciences, 26*(5), 1459–1479. https://doi.org/10.5194/hess-26-1459-2022

[76] Sobhi Gollo, V., Broecker, T., Marx, C., Lewandowski, J., Nützmann, G., & Hinkelmann, R. (2022). A comparative study of integral and coupled approaches formodeling hydraulic exchange processes across a rippled streambed. *International Journal on Geomathematics, 13*(16), 1-27. https://doi.org/10.1007/s13137-022-00206-5

[77] Soltani, S. S., Ataie-Ashtiani, B., Al Bitar, A., Simmons, C. T., Younes, A., & Fahs, M. (2024). Assimilating multivariate remote sensing data into a fully coupled subsurface-land surface hydrological model. *Journal of Hydrology, 641*, 131812. https://doi.org/10.1016/j.jhydrol.2024.131812

[78] Soulaine, C., & Tchelepi, H. A. (2016). Micro-continuum Approach for Pore-Scale Simulation of Subsurface Processes. *Transport in Porous Media, 113*, 431–456. https://doi.org/10.1007/s11242-016-0701-3

[79] Steiness, M., Jessen, S., Spitilli, M., van't Veen, S. G., Højberg, A. L., & Engesgaard, P. (2019). The role of management of stream–riparian zones on subsurface–surface flow components. *Water, 11*(9), 1905. https://doi.org/10.3390/w11091905

[80] Tang, Q., Delottier, H., Kurtz, W., Nerger, L., Schilling, O. S., & Brunner, P. (2024). HGS-PDAF (version 1.0): a modular data assimilation framework for an integrated surface and subsurface hydrological model. *Geoscientific Model Development, 17*(8), 3559–3578. https://doi.org/10.5194/gmd-17-3559-2024

[81] Tang, Q., Kurtz, W., Schilling, O. S., Brunner, P., Vereecken, H., & Hendricks Franssen, H. J. (2017). The influence of riverbed heterogeneity patterns on river-aquifer exchange fluxes under different connection regimes. *Journal of Hydrology, 554*, 383-396. https://doi.org/10.1016/j.jhydrol.2017.09.031

[82] Tang, Q., Schilling, O. S., Kurtz, W., Brunner, P., Vereecken, H., & Hendricks Franssen, H.-J. (2018). Simulating Flood-Induced Riverbed Transience Using Unmanned Aerial Vehicles, Physically Based Hydrological Modeling, and the Ensemble Kalman Filter. *Water Resources Research, 54*(11), 9342-9363. https://doi.org/10.1029/2018WR023067

[83] Therrien, R., McLaren, R., Sudicky, E., & Panday, S. (2010). *HydroGeoSphere: A Three-Dimensional Numerical Model Describing Fully-Integrated Subsurface and Surface Flow and Solute Transport.* Groundwater Simulations Group. Waterloo: University of Waterloo.

[84] Tirado-Conde, J., Engesgaard, P., Karan, S., Müller, S., & Duque, C. (2019). Evaluation of temperature profiling and seepage meter methods for quantifying submarine groundwater discharge to coastal lagoons: Impacts of saltwater intrusion

and the associated thermal regime. *Water, 11*(8), 1648. https://doi.org/10.3390/w11081648

[85] Tonina, D., & Buffington, J. M. (2009). Hyporheic Exchange in Mountain Rivers I: Mechanics and Environmental Effects. *Geography Compass, 3*(3), 1063-1086. https://doi.org/10.1111/j.1749-8198.2009.00226.x

[86] Toran, L. (2019). Groundwater-Surface Water Interaction. *Encyclopedia of Water: Science, Technology, and Society*, 1-12. https://doi.org/10.1002/9781119300762.wsts0027

[87] Trauth, N., Musolff, A., Knoller, K., Kaden, U. S., Keller, T., Werban, U., & Fleckenstein, J. H. (2018). River water infiltration enhances denitrification efficiency in riparian groundwater. *Water Research, 130*, 185-199. https://doi.org/10.1016/j.watres.2017.11.058

[88] Trauth, N., Schmidt, C., Vieweg, M., Maier, U., & Fleckenstein, J. H. (2014). Hyporheic transport and biogeochemical reactions in pool-riffle systems under varying ambient groundwater flow conditions. *Journal of Geophysical Research: Biogeosciences, 119*(5), 910-928. https://doi.org/10.1002/2013JG002586

[89] Trauth, N., Schmidt, C., Vieweg, M., Oswald, S. E., & Fleckenstein, J. H. (2015). Hydraulic controls of in-stream gravel bar hyporheic exchange and reactions. *Water Resources Research, 51*(4), 2243-2263. https://doi.org/10.1002/2014WR015857

[90] Versteeg, H. K., & Malalasekera, W. (2007). *An introduction to computational fluid dynamics: The finite volume method* (2nd ed.). Edinburgh, UK: Pearson. ISBN-13: 978-0131274983

[91] Voermans, J. J., Ghisalberti, M., & Ivey, G. N. (2017). The variation of flow and turbulence across the sediment-water interface. *Journal of Fluid Mechanics, 824*, 413-437. https://doi.org/10.1017/jfm.2017.345

[92] Warden, J. G., Izbicki, J. A., Sültenfuß, J., Scheiderich, K., & Fitzpatrick, J. (2023). *Environmental tracers of groundwater source, age, and geochemical evolution.* Hinkley, California: U.S. Geological Survey Professional Paper 1885-F. https://doi.org/10.3133/pp1885F

[93] Watts, G. (2024). Global environmental pressures. En S. Krause, D. Hannah, & N. Grimm (Eds.), *Ecohydrological Interfaces* (pp. 331-353). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119489702.ch15

[94] Weller, H. G., Tabor, G., Jasak, H., & Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics, 12*(6), 620-631. https://doi.org/10.1063/1.168744

[95] White, J. T., Hunt, R. J., Fienen, M. N., & Doherty, J. E. (2020). *Approaches to Highly Parameterized Inversion: PEST++ Version 5, a Software Suite for Parameter*

*Estimation, Uncertainty Analysis, Management Optimization and Sensitivity Analysis.* Techniques and Methods, U.S. Geological Survey,. U.S. Geological Survey, Techniques and Methods. https://doi.org/10.3133/tm7C26

[96] Winter, T. C., Harvey, W. J., Franke, O. L., & Alley, M. W. (1998). *Ground water and surface water: A single resource.* North Dakota Water Science Center. Denver: U.S. Geological Survey. https://doi.org/10.3133/cir1139

# 2

# darcyInterTransportFoam v1.0: an Open-Source, Fully-Coupled 3D Solver for Simulating Surface Water – Saturated Groundwater Processes and Exchanges

A manuscript based on this chapter is currently in submission to the journal *Geoscientific Model Development*.

# 2.1. Introduction

Acknowledging the role that surface water-groundwater (SW-GW) interactions play for the health of aquatic ecosystems has gained importance in recent years (Brunner, et al., 2017; Krause, et al., 2017; Lewandowski, et al., 2020). Surface water bodies, such as streams, lakes or wetlands, and the aquifers they are connected to should be considered as a single, integral entity due to their continuous interaction (Winter, et al., 1998). This interaction controls the fate and transport of contaminants and waterborne substances in the system (Dichgans, et al., 2023; Nisbeth, et al., 2019; Schaper, et al., 2018). Likewise, aerobic and anaerobic reactions as well as the different biological activities produced in the ground are also influenced by this bidirectional exchange (Lewandowski, et al., 2019; Nogueira, et al., 2021b; Trauth, et al., 2018). Consequently, the water quality and the overall ecological health of the SW-GW system is affected by the continuous interaction across the sediment-water interface (Boano, et al., 2014; Toran, 2019).

Multiple investigations on both water and solute exchange have recently been conducted to study the interconnection between SW and GW (Coluccio & Morgan, 2019; Hermans, et al., 2023; Krause, et al., 2023). In this context, numerical models arise as a viable complement to field and laboratory experiments to investigate the dynamic interactions across the sediment-water interface (Barthel & Banzhaf, 2016; Brunner, et al., 2017; Fleckenstein, et al., 2010; Krause, et al., 2014). Compared to physical studies, numerical models can provide high-resolution spatiotemporal information on relevant flow and transport processes at any location in the system (Broecker, et al., 2021). SW-GW interaction cases are commonly solved using an asynchronous linking, or one-way, coupling approach (Maxwell, et al., 2014), with pressure as the coupling parameter (e.g., Chen, et al., 2015; Chen, et al., 2018; Jin, et al., 2011; Lee, et al., 2020; Trauth, et al., 2014; Trauth & Fleckenstein, 2017). Following this technique, a surface flow model is coupled to a groundwater flow model, where the pressure distribution at the bottom boundary of the surface model is imposed as the upper boundary condition (BC) for the groundwater model. No feedback is provided from the subsurface to the surface domain after solving the groundwater fields, meaning that the resulting exchange fluxes do not influence the surface flow dynamics. However, since flow and transport exchanges are bidirectional, considering the interfacial feedback as unidirectional may result in significant errors in the water and mass balances (Li, et al., 2020). In this way, fully-coupled models (FCMs) account for the actual bidirectional interaction between SW and GW by transferring parameter information from one domain to another through an iterative algorithm. This coupling strategy is usually referred to as sequential iterative, or two-way, approach (Maxwell, et al., 2014). FCMs commonly solve the subsurface flow using Richards' equation and Darcy's law, while applying the shallow water equations or the Navier-Stokes equations to simulate the surface flow. Unlike the shallow water

equations, the Navier-Stokes equations account for turbulence in the surface domain. Turbulence has proven to influence the magnitude of SW-GW exchange and, as such, its detailed computation should not be ignored (Tonina & Buffington, 2009; Voermans, et al., 2017). The Navier-Stokes equations are solved by Computational Fluid Dynamics (CFD) models. CFD solvers have the potential to account for the complex three-dimensional (3D) mechanisms of flow dynamics, including turbulence, and therefore, to provide high-resolution information on the flow field characteristics.

Among the different CFD models available to simulate 3D processes, the **OpenFOAM** (**Open**-Source **F**ield **O**peration **A**nd **M**anipulation) software has grown in popularity in recent years (e.g., Broecker, et al., 2017; Broecker, et al., 2021; Lee, et al., 2020; Lee, et al., 2022; Sobhi Gollo, et al., 2022; Trauth, et al., 2014; Trauth, et al., 2015). **OpenFOAM** (Weller, et al., 1998) is a powerful, free and open-source C++ toolbox consisting of a comprehensive set of applications (solvers, utilities, boundary conditions, etc.) designed to simulate diverse two and three-dimensional continuum mechanics problems, most prominently including CFD (OpenCFD Ltd, 2024). The main advantage of **OpenFOAM** over other CFD models is the modular and extensible nature of its code structure. This flexibility allows to customize and extend the software beyond the standard applications to suit specific simulation requirements. As such, **OpenFOAM** provides an adequate modelling framework to implement FCMs for the thorough study of the 3D SW-GW processes. Based on this, Li, et al. (2020) developed **`hyporheicFoam`**. This user-defined model simulates flow and biogeochemical reactions in both surface and subsurface domains as well as the two-way exchange across their interface. The multicomponent reactive transport is modelled in both domains by generic advection-diffusion/dispersion-reaction equations, whereas the standard **`pimpleFoam`** solver is used to simulate the surface flow dynamics and a new-coded **OpenFOAM** solver is applied to solve the saturated groundwater flow. The full coupling between domains is achieved via the mapping of conservative flux BCs at the interface through an iterative scheme. The fact of using a transient, incompressible solver like **`pimpleFoam`** to simulate the surface flow implies that **`hyporheicFoam`** assumes the water surface is flat, i.e., a rigid-lid (e.g., Chen, et al., 2018; Janssen, et al., 2012; Jin, et al., 2011; Lee, et al., 2020). Nevertheless, the water surface in rivers is free to move and can adjust to the topography. In this way, prior studies have shown the impact that a deformable, free surface has on the magnitude of bedform-driven SW-GW exchange, indicating an underestimation of the flux exchanges under the rigid-lid assumption (e.g., Broecker, et al., 2019; Broecker, et al., 2021; Trauth, et al., 2014; Trauth, et al., 2015). To overcome this limitation, a new fully-coupled 3D model, called **`hyporheicScalarInterFoam`**, was implemented by Lee, et al. (2021). This novel solver switches from **`pimpleFoam`** to **`interFoam`** to simulate the surface flow dynamics but keeps the same solver as **`hyporheicFoam`** for the porous media flow. **`interFoam`** (Deshpande, et al., 2012) is a two-phase, incompressible, immiscible solver within the **OpenFOAM** library that uses the Volume of Fluid (VOF)

phase-fraction based interface capturing approach (Hirt & Nichols, 1981) for solving the three-dimensional Navier-Stokes equations. This interface tracking method allows modelling the free surface located between the two simulated fluids (i.e., air and water in this case), so that the deformations of the river's water surface can be considered in the computation of the SW-GW exchange. Furthermore, in contrast to `hyporheicFoam`, `hyporheicScalarInterFoam` only simulates conservative transport at both domains. Model-specific flux BCs are also mapped at the interface through an iterative algorithm for the full coupling of the two domains.

Despite the potential of `hyporheicScalarInterFoam` to simulate surface water-groundwater (SW-GW) flow dynamics and transport processes, its applicability to actual hydrogeological contexts is constrained by various modelling limitations, including: (1) the solver does not explicitly calculate any hydrology-related field, such as the specific discharge or the hydraulic head, but solves the surface and subsurface hydrodynamics based on the so-called modified pressure and the volume-averaged velocity. (2) A water height cannot be set at the outlet patch of the surface domain and, therefore, neither can a hydrostatic pressure BC that would allow the flow to leave the simulation domain freely. (3) The adjustable body force required to make the pressure field compatible with periodic BCs is only considered in the calculation of the specific discharge, while both the hydraulic head and the groundwater flux are determined without this additional pressure gradient. (4) The model only allows the specification of homogeneous and isotropic parameter fields (i.e. hydraulic conductivity, porosity and longitudinal and transversal dispersivities) for the entire subsurface domain, which prevents the definition of further complex underground characterizations (e.g., heterogeneous fields). Moreover, (5) the sediment-water interface has to be fully submerged so that the model can run, which implies that both domains must be hydraulically connected all over the interface boundary. (6) The transport of solutes requires to always be solved in both phases in the surface domain, whereas (7) the heat transfer simulation is not available in any of the domains. (8) The mapping of data between the domains is performed via interpolation at their interface boundary, which may result in a loss of information depending on the resolution of both meshes. (9) The convergence of the surface and subsurface results is evaluated by a user-specific maximum iteration loop that does not check the solution mismatch between both domains after each iteration. (10) The model does not allow defining BCs at the interface other than those used for the two-way coupling mode, which complicates its execution in one-way coupling mode. Furthermore, (11) the reinitialization of the data fields at any time of the simulation is only possible through data-preprocessing. Consequently, further development of the code is required to fill all the above-mentioned simulation gaps as well as to extend the solver functionality, so that the model could be applied to different scale SW-GW interaction scenarios.

Hence, this paper introduces an updated and extended version of the `hyporheicScalarInterFoam` solver, based on the computational fluid dynamics (CFD)

platform OpenFOAM. Analogously to the earlier version, the new code, called darcyInterTransportFoam, is an open-source, fully-coupled 3D model aiming at the simulation of flow and transport processes in both surface and subsurface domains, as well as the interactions between them. The latter is achieved through the bidirectional mapping of flux BCs at the domain's interface via an iterative scheme. Moreover, as intended, darcyInterTransportFoam upgrades the applicability of the original solver by extending its modelling capabilities through a series of solver updates and new simulation add-ons. The latter consist of several pre- and post-processing utilities, along with three novel boundary conditions. As for the solver updates, these include, among others, the possibility to simulate solute transport in a single phase within the surface domain, the definition of non-uniform underground parameter fields, the automatic specification of impervious regions in the subsurface domain and the simulation of heat transfer in both domains. Furthermore, the structure and organization of the code is also updated to make it more user-friendly. Biogeochemical reactions, targeted by other user-defined, fully-coupled OpenFOAM solvers, such as hyporheicFoam (Lee, et al., 2022; Li, et al., 2020), cannot be simulated with the current version of the model, just as they could not in hyporheicScalarInterFoam.

The efficacy of the new solver is additionally demonstrated through a couple of three-dimensional test cases, in which the SW-GW flow, solute transport and heat transfer processes occur at two locations of an actual alluvial river-aquifer system are simulated using darcyInterTransportFoam.

## 2.2. Model Description

darcyInterTransportFoam (Pardo-Álvarez, 2025) is implemented in the free and open-source CFD software OpenFOAM v2412. Similarly to its predecessor, the new fully-coupled model applies a pseudo-transient coupling approach, in which transient surface flow and steady-state saturated subsurface flow are simulated at every time step (Figure 2.1). Accordingly, the subsurface solution is updated at each time step based on the transient BCs computed at the SW-GW interface during the surface flow simulation. This subsurface adjustment is assumed to occur quasi-instantaneously, consistent with the slower response of groundwater compared to surface water. After solving the subsurface flow, relevant quantities are mapped back onto the surface domain within the same time step, reflecting the combined effects of the transient surface forcing and any additional subsurface boundary contributions. A convergence check is then performed to ensure consistency between surface and subsurface exchanges after mapping. A similar procedure is applied to solute transport and heat transfer, although in these processes transience is considered in both domains. Due to the different characteristic response timescales of surface and subsurface flows, a single iteration per time step is generally

sufficient, unless strong, fast two-way feedback across the interface requires additional convergence loops. This approach reflects the physical assumption that, over the short timescales of surface flow variations, the subsurface water table remains stable and does not undergo significant movement. As a result, the model achieves faster and more stable simulations.

The governing equations and coupling modes of `darcyInterTransportFoam` are described in this section, followed by an overview of its newly introduced simulation features.

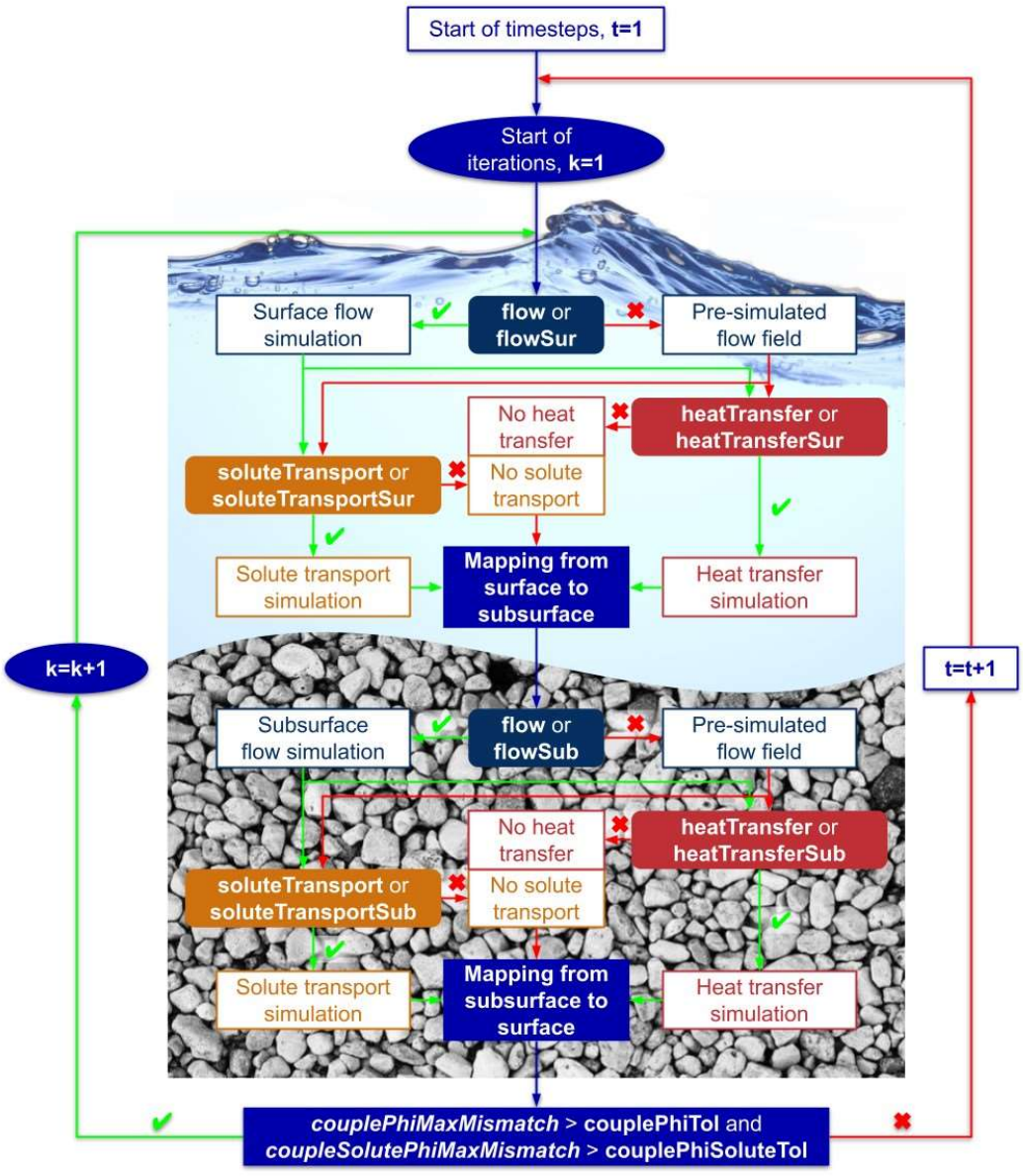



Figure 2.1. Simulation workflow of `darcyInterTransportFoam`. Mass conservation is ensured by the updated iterative scheme, that enforces both water and solute flux consistency across the interface of the surface and subsurface domains.

### 2.2.1. Governing Equations

The new FCM introduces small modifications to some original expressions while expanding the functionality of **hyporheicScalarInterFoam** by incorporating additional equations and simulation options. In this way, the user-defined code for simulating groundwater flow has been updated to improve accessibility and fully account for the influence of surface dynamics on the subsurface solution. Additionally, modified advection-diffusion-dispersion and novel temperature energy equations are implemented in each domain to respectively enhance the solute transport simulation and enable heat transfer modelling.

Both surface and subsurface partial differential equations are discretized using the Finite Volume Method (FVM) in space and the Finite Difference Method (FDM) in time (Schulze & Thorenz, 2014).

#### 2.2.1.1. Surface Domain

**darcyInterTransportFoam** models the surface flow as a transient, immiscible, two-phase flow using the VOF interface capturing approach. This method is extensively applied in environmental studies that require the simulation of free-surface flows, such as the air-water flow in an open-channel (e.g., Broecker, et al., 2017; Sobhi Gollo, et al., 2022). The VOF approach separates the two phases (i.e., air and water) by a continuous, free interface, whose location and shape are determined as a property of the phase fraction field, $\alpha_i$ (-) (Schulze & Thorenz, 2015). Such deformation of the interface allows us to account for the actual flow dynamics in the surface domain, which in turn leads to a correct estimate of the flow, solute and heat exchanges across the sediment-water interface.

##### 2.2.1.1.1. Turbulent Two-Phase Flow

As commonly assumed for free-surface fluids, the surface flow is additionally considered to be incompressible (Young, et al., 2010). Consequently, there is no need for equations of state to relate the thermodynamic variables (Versteeg & Malalasekera, 2007) and the two-phase flow is adequately governed by the continuity equation (**Equation** (**2.1**)), the Navier-Stokes equations (**Equation** (**2.2**)) and the $\alpha_i$ transport equation (**Equation** (**2.8**)). The latter expression is also known as VOF equation, since it is the one used to determine the distribution of both phases through the surface domain.

The VOF method solves the surface flow dynamics considering the two immiscible phases as one fluid. The resulting single-phase is assumed to be Newtonian and incompressible, according to the physical properties of both surface phases. As a result, the continuity equation can be simplified to:

$$\nabla \cdot U_{sur,i} = 0 \tag{2.1}$$

and the momentum equation to:

$$\frac{\partial \rho_{sur,i} U_{sur,i}}{\partial t} + \nabla \cdot \left(\rho_{sur,i} U_{sur,i} U_{sur,i}\right) - \nabla \cdot \underbrace{\left[\mu_{sur,i}\left(\nabla U_{sur,i} + \left(\nabla U_{sur,j}\right)^T\right)\right]}_{\tau_{ij}} = -\nabla p_{sur,i} + \rho_{sur,i} g + \rho_{sur,i} S_{U_{sur,i}} \tag{2.2}$$

where $U_{sur,i}$ and $U_{sur,j}$ respectively denote the single-phase velocity field in the x and y directions (m/s), $t$ is the time (s), $\rho_{sur,i}$ and $\mu_{sur,i}$ are respectively the single-phase density (kg/m$^3$) and dynamic viscosity fields (N/m$^2$), $\tau_{ij}$ is the deviatoric viscous stress tensor field (N/m$^2$), $p_{sur,i}$ is the surface static pressure field (Pa), $g$ is the gravitational acceleration vector (m/s$^2$) and $S_{U_{sur,i}}$ accounts for any source/sink term field (m/s).

Analogously to other multi-phase models, **darcyInterTransportFoam** solves the momentum equations using the so-called modified pressure, $p_{rgh_{sur,i}}$ (Pa). This alternative pressure enables to avoid potential steep pressure gradients caused by hydrostatic effects as well as simplifying the definition of pressure BCs (Rusche, 2003). The $p_{rgh_{sur,i}}$ field is obtained by subtracting the hydrostatic pressure component to $p_{sur,i}$:

$$p_{rgh_{sur,i}} = p_{sur,i} - \rho_{sur,i}(g \cdot x_i) \tag{2.3}$$

where $x_i$ is the cell position vector field (m). However, since $g$ only applies in the vertical direction, $x_i$ actually accounts for the cell height field, $z_{sur,i}$ (m). After the corresponding substitutions, the Navier-Stokes equation becomes:

$$\frac{\partial \rho_{sur,i} U_{sur,i}}{\partial t} + \nabla \cdot \left(\rho_{sur,i} U_{sur,i} U_{sur,i}\right) - \nabla \cdot \underbrace{\left[\mu_{sur,i}\left(\nabla U_{sur,i} + \left(\nabla U_{sur,j}\right)^T\right)\right]}_{\tau_{ij}} = -\nabla p_{rgh_{sur,i}} - g \cdot x_i \nabla \rho_{sur,i} + \rho_{sur,i} S_{U_{sur,i}} \tag{2.4}$$

The single-phase parameters $\rho_{sur,i}$ and $\mu_{sur,i}$ are calculated by weighted averaging the physical properties of both surface fluids based on $\alpha_i$:

$$\rho_{sur,i} = \alpha_i \rho_{w_{sur}} + (1 - \alpha_i)\rho_{a_{sur}} \tag{2.5}$$

$$\mu_{sur,i} = \alpha_i \mu_{w_{sur,i}} + (1 - \alpha_i)\mu_{a_{sur,i}} \tag{2.6}$$

where $\rho_{w_{sur}}$ and $\rho_{a_{sur}}$ are respectively the densities of the surface water and the air (kg/m$^3$) and $\mu_{w_{sur,i}}$ and $\mu_{a_{sur,i}}$ represent the surface water and air dynamic viscosity fields (N/m$^2$). The latter properties result from:

$$\mu_{p_{sur,i}} = \mu_{phys_{p_{sur}}} + \mu_{turb,i} \tag{2.7}$$

where $p$ is the subscript for the respective surface phase, $w$ (water) or $a$ (air), and $\mu_{phys_{p_{sur}}}$ and $\mu_{turb,i}$ are respectively the physical and turbulent dynamic viscosities

(N/m$^2$). The $\mu_{turb,i}$ field is computed by the selected turbulence model. As most **OpenFOAM** solvers, **darcyInterTransportFoam** allows to choose among different Reynolds-Averaged Navier-Stokes (RANS), Large Eddy Simulation (LES) and hybrid RANS-LES Detached Eddy Simulation (DES) models to simulate the turbulent hydrodynamics. The choice of model will ultimately depend on the characteristics of the problem being addressed and the desired level of simulation accuracy. LES models are more advanced than RANS and directly simulate the large-scale eddies while resolving the small-scale eddies via a sub-grid scale algebraic model. This latter fact, along with the need to substantially decrease the mesh resolution (and ideally consider a 3D mesh) to achieve an optimal solution, lead to an increase of the computation times. Such mesh demands are not required by RANS, which allows to solve the turbulent flow with less computational effort and yet an adequate accuracy. Consequently, the widely applied k-ω SST (Shear Stress Transport) RANS model (Menter, 1994) was chosen to resolve the different scales of turbulence in both test cases of the present study. This model splits the turbulent flow velocity into a time-averaged and an instantaneous velocity, leading to a Reynolds $\tau_{ij}$ in the Navier-Stokes equations. This tensor determines the turbulent kinematic viscosity field, $\nu_{turb,i} = k_i/\omega_i$ (m$^2$/s), by solving the turbulent kinetic energy field, $k_i$ (m$^2$/s$^2$), and the specific energy dissipation field, $\omega_i$ (1/s), via a two-equation transport model. More details about the different turbulence models available in **OpenFOAM** can be found in its online User's Guide (OpenCFD Ltd, 2023).

An advanced transport equation is used to accurately determine the distribution of $\alpha_i$ through the surface domain and, therefore, the position of the free surface. In this way, the relative volume fraction of both phases is calculated in each computational cell by introducing an additional convective term, referred to as compression term, into the standard $\alpha_i$ transport equation (Berberović, et al., 2009):

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot \left(\alpha_i U_{sur,i}\right) + \underbrace{\nabla \cdot \left[(1-\alpha_i)\alpha_i U_{r,i}\right]}_{Compression\ term} = 0 \tag{2.8}$$

where $U_{r,i} = U_{w_{sur,i}} - U_{a_{sur,i}}$ is the relative velocity field (m/s) between the two surface phases, known as the compression velocity (Berberović, et al., 2009). The compression term is originated from modelling the advective flow velocity as a weighted average of both phase velocities, assuming that the contribution of each to the evolution of the free surface is proportional to the corresponding $\alpha_i$ (Marquez-Damian, 2012). This additional source provides an artificial contribution to the convection of $\alpha_i$ within the interface region. Outside of this zone, the term vanishes. As a result, the free surface is "compressed", leading to a sharper interface resolution which ultimately enables the proper conservation of $\alpha_i$ throughout the simulation domain (Weller, 2002). The latter is particularly important in the case of high-density fluids, where small errors in the $\alpha_i$ distribution can generate significant errors in the physical properties (Rusche, 2003).

The resultant $\alpha_i$ is strictly bounded between 0 (air) and 1 (water) in each computational cell of the surface domain. Any value in between will imply a mixture of both phases. As for the interface, the VOF method estimates its location as a property of $\alpha_i$ rather than explicitly computing it. Consequently, the free surface is never sharply defined but occupies a volume around the mesh region where a sharp interface is supposed to exist. Such volume is composed by the computational cells where $0 < \alpha_i < 1$, being the interface considered to be around $\alpha_i = 0.5$.

Following the $U_{sur,i}$, $p_{sur,i}$ and $\alpha_i$ solutions, the newly implemented fields of surface water depth, $d_{sur,i}$ (m), piezometric level, $piezo_{sur,i}$ (m) and surface hydraulic head, $h_{sur,i}$ (m), are subsequently calculated according to Bernoulli's equation as:

$$d_{sur,i} = \frac{p_{sur,i}}{\rho_{sur,i} \cdot |g|} \tag{2.9}$$

$$piezo_{sur,i} = d_{sur,i} + z_{sur,i} \tag{2.10}$$

$$h_{sur,i} = \frac{U_{sur,i}^2}{2 \cdot |g|} + piezo_{sur,i} \tag{2.11}$$

#### 2.2.1.1.2. Passive Solute Transport

The new FCM provides two simulation options for modelling the transport of a conservative solute in the surface domain. The first corresponds to the method previously implemented in **hyporheicScalarInterFoam**. This approach, based on the formulation introduced by Teuber (2020), solves a one-fluid advection-diffusion equation within the VOF method, using Henry's law as constant coefficient at the interface:

$$\frac{\partial C_{sur,i}}{\partial t} + \nabla \cdot \left(U_{sur,i} C_{sur,i}\right) - \nabla \cdot \left(D_{sur,i} \nabla C_{sur,i}\right) + \nabla \cdot \left(\phi_i C_{sur,i}\right) = S_{C_{sur,i}} \tag{2.12}$$

where $C_{sur,i}$ is the surface concentration field (kg/m$^3$), $D_{sur,i}$ is the surface diffusion coefficient field (m$^2$/s), $\phi_i$ represents the concentration flux at the interface (m$^2$/s) and $S_{C_{sur,i}}$ accounts for any concentration source/sink field (kg/m$^3$). $D_{sur,i}$ is calculated as the sum of the physical (or molecular) diffusivity, $D_{phys}$ (m$^2$/s), and the turbulent diffusivity, $D_{turb,i}$ (m$^2$/s), following Broecker et al. (2017):

$$D_{sur,i} = D_{phys} + \left\{\frac{\nu_{turb,i}}{Sc_{turb}}\right\} D_{turb,i} \tag{2.13}$$

where $Sc_{turb}$ is the turbulent Schmidt number (-). Both $D_{phys}$ and $Sc_{turb}$ have to be defined by the user. The computation of $D_{sur,i}$ differs from that in **hyporheicScalarInterFoam** in that it also accounts for the contribution of $D_{phys}$ to the total diffusivity across the domain, a factor particularly relevant in low-turbulence zones or for fine-scale processes. Alternatively, $D_{sur,i}$ can be specified via a constant value for

the whole surface domain, offering a simplified yet practical setup not available in the original solver. $\phi_i$ is described by the following equation (Haroun, et al., 2010):

$$\phi_i = D_{sur,i} \frac{1 - H_{cc}}{\alpha_{ij} + H_{cc}(1 - \alpha_i)} \nabla \alpha_i \tag{2.14}$$

where $H_{cc}$ is the user-defined Henry's law solubility constant (-). The specified $H_{cc}$ must be fulfilled at the interface in order to ensure a consistent concentration flux between the two phases. Further details about this approach can be found in Haroun, et al. (2010), Nieves-Remacha, et al. (2015) or Severin (2017).

The second simulation option is newly implemented and designed to reduce computational costs when the focus is limited to a single simulated phase. Based on the two-phase advection-diffusion equation contained in the standard **scalarTransport** function object (OpenCFD Ltd, 2023), this approach uses the computed $\alpha_i$ to delimit the extension of the solute transport simulation within the system. Accordingly, the passive transport of any species is only simulated in those cells whose $\alpha_i$ is equal to 1, skipping the rest in the computation. The described governing equation has the subsequent form:

$$\frac{\partial C_{sur,i}}{\partial t} + \nabla \cdot \left(\alpha_i U_{sur,i} C_{sur,i}\right) - \nabla \cdot \left(D_{sur,i} \nabla C_{sur,i}\right) = \alpha_i S_{C_{sur,i}} \tag{2.15}$$

The latter approach was the one selected to model the conservative solute transport in the two test cases presented in this study.

#### 2.2.1.1.3. Heat Transfer

Analogously to solute transport (**Section 2.2.1.1.2**), **darcyInterTransportFoam** enables to solve the surface heat transfer through two new simulation approaches. One of these is based on Aryal (2019) and simulates the distribution of temperature in both surface phases using the following energy equation:

$$\rho Cp_{sur,i} \frac{\partial T_{sur,i}}{\partial t} + \nabla \cdot \left(\rho Cp_{sur,i} U_{sur,i} T_{sur,i}\right) - \nabla \cdot \left(K_{sur,i} \nabla T_{sur,i}\right) = \rho Cp_{sur,i} S_{T_{sur,i}} \tag{2.16}$$

where $\rho$ is the density (kg/m$^3$), $Cp_{sur,i}$ is the surface specific heat capacity field (m$^2$/s$^2$·°C), $T_{sur,i}$ is the surface temperature field (°C), $K_{sur,i}$ is the surface thermal conductivity field (kg·m/s$^3$·°C) and $S_{T_{sur,i}}$ represents any heat source/sink field (°C). Like it happens to $\rho_{sur,i}$ and $\mu_{sur,i}$, the $\rho Cp_{sur,i}$ field is also computed as a single-phase property by weighted averaging the $\rho$ and $Cp$ of both phases based on $\alpha_i$:

$$\rho Cp_{sur,i} = \alpha_i \rho_{w_{sur}} Cp_{w_{sur}} + (1 - \alpha_i) \rho_{a_{sur}} Cp_{a_{sur}} \tag{2.17}$$

where $Cp_{w_{sur}}$ and $Cp_{a_{sur}}$ account for the surface water and air specific heat capacities (m$^2$/s$^2$·°C). $K_{sur,i}$ is also calculated as a function of $\alpha_i$ as:

$$K_{sur,i} = \alpha_i \underbrace{\frac{\rho_{w_{sur}} Cp_{w_{sur}} v_{phys_{w_{sur}}}}{Pr_{w_{sur}}}}_{K_{w_{sur}}} + (1-\alpha_i) \underbrace{\frac{\rho_{a_{sur}} Cp_{a_{sur}} v_{phys_{a_{sur}}}}{Pr_{a_{sur}}}}_{K_{a_{sur}}} \quad (2.18)$$

where $v_{phys_{w_{sur}}}$ and $v_{phys_{a_{sur}}}$ are respectively the surface water and air physical kinematic viscosities ($m^2/s$), $Pr_{w_{sur}}$ and $Pr_{a_{sur}}$ are the Prandtl numbers of the surface water and air phase (-) and $K_{w_{sur}}$ and $K_{a_{sur}}$ represent the surface water and air thermal conductivities ($kg \cdot m/s^3 \cdot °C$).

The second simulation approach restricts the computation of heat transfer to a single surface phase, thereby reducing computational demands. Similarly to solute transport (Section 2.2.1.1.2), this is done based on the $\alpha_i$ solution. As such, the transfer of heat is skipped in all those computational cells whose $\alpha_i$ differs from 1. The resulting energy equation has the form of:

$$\rho_{w_{sur}} Cp_{w_{sur}} \frac{\partial T_{sur,i}}{\partial t} + \nabla \cdot \left(\rho_{w_{sur}} Cp_{w_{sur}} \alpha_i U_{sur,i} T_{sur,i}\right) - \nabla \cdot \left(K_{sur,i} \nabla T_{sur,i}\right) = \rho_{w_{sur}} Cp_{w_{sur}} \alpha_i S_{T_{sur,i}} \quad (2.19)$$

As occurred with solute transport, this latter simulation option was chosen to solve the temperature dynamics and exchanges produced in the two implemented test cases.

### 2.2.1.2. Subsurface Domain

The introduced model version considers the subsurface domain to always be under saturated conditions regardless of the changes of head with time. Such head changes mainly result from the surface hydrodynamics, which are transferred to the subsurface across the interface of both domains. Surface and subsurface domains are, therefore, hydraulically connected throughout the interface boundary. The subsurface flowpaths as well as the solute and heat processes are driven by the resulting saturated head gradients in the ground.

On the other hand, in contrast to the prior model version, **`darcyInterTransportFoam`** defines the subsurface geological parameters, namely, porosity, $\theta_i$ (-), hydraulic conductivity, $K_{ij}$ (m/s), and longitudinal and transverse dispersivities, $\alpha_{L,i}$ and $\alpha_{T,i}$ (m), as three-dimensional scalar and tensor fields. This novel feature allows to extend the applicability of the model to actual alluvial aquifers commonly characterized by heterogeneous $\theta_i$, $\alpha_{L,i}$ and $\alpha_{T,i}$ scalar fields as well as heterogeneous and anisotropic $K_{ij}$ tensor fields.

#### 2.2.1.2.1. Saturated Groundwater Flow

Since the subsurface domain is assumed to be fully saturated throughout the whole simulation, the saturated thickness of the aquifer does not change with time and, thus,

neither does the position or slope of the water table within it. As such, the subsurface flow is simulated as steady-state, with its final solution depending on the transient surface hydraulics and the selected BCs. Considering all the previous conditions, the 3D groundwater flow is described by the following continuity equation for a heterogeneous, anisotropic aquifer:

$$\nabla \cdot \left(-K_{ij} \nabla h_{sub,i}\right) = 0 \tag{2.20}$$

where $h_{sub,i}$ is the subsurface hydraulic head field (m). Mass conservation is thus solved based on $h_{sub,i}$, in contrast to the modified pressure field, $p_{rgh_{sub,i}}$ (Pa), used in **hyporheicScalarInterFoam**. While both variables are closely related, they are not identical, as $p_{rgh_{sub,i}}$ does not include the velocity head:

$$\frac{p_{rgh_{sub,i}}}{\rho_{w_{sub}} g} = h_{sub,i} - \frac{{q_i}^2}{2|g|{\theta_i}^2} \tag{2.21}$$

where $\rho_{w_{sub}}$ is the groundwater density (kg/m$^3$) and $q_i$ is the specific discharge field (m/s). Consequently, $p_{rg\ _{sub,i}}$ approximates $h_{sub,i}$ in the subsurface, as kinetic energy effects are typically negligible due to the low flow velocities in porous media. However, solving the groundwater flow based on $h_{sub,i}$ allows for the incorporation of the surface water velocity effects at the SW-GW interface, thereby accounting for their potential influence on the overall subsurface dynamics. This contribution was omitted in the original version of the solver. $h_{sub,i}$ is derived from Bernoulli's equation, after neglecting the kinetic component:

$$h_{sub,i} = \frac{\overbrace{p_{sub,i}}^{d_{sub,i}}}{\rho_{w_{sub}} \cdot |g|} + z_{sub,i} \tag{2.22}$$

where $p_{sub,i}$ is the subsurface static pressure field (Pa), $d_{sub,i}$ is the subsurface water depth field (m) and $z_{sub,i}$ is the cell elevation field (m).

Following the head solution, $q_i$ (Haitjema & Anderson, 2016) is sequentially computed using Darcy's law:

$$q_i = -K_{ij} \nabla h_{sub,i} \tag{2.23}$$

Darcy's law is only applicable to very slow-moving groundwater, in which the resistive forces of viscosity predominate. Experimentation has shown that this is fulfilled when the Reynolds number is less than a value from 1 to 10 (Bear, 1972), meaning the groundwater flow is laminar. This situation is satisfied in most natural groundwater environments. However, exceptions may exist, such as regions with big porosities or steep hydraulic gradients, in which the Reynolds number can be so big as to invalidate Darcy's law. Under such circumstances, alternative solutions need to be applied for a correct simulation of the groundwater flow, e.g., the Darcy-Brinkmann equation (Yao &

Zhang, 2023), the Navier-Stokes equation attenuated by the Darcy-Forchheimer term (Dichgans, et al., 2023) or models like `porousInter` (Oxtoby, et al., 2013). The application of Darcy's law in non-Darcy-flow porous media will lead to an overestimation of the groundwater flow rates. `darcyInterTransportFoam` is, therefore, aimed for slow-moving laminar subsurface flows.

#### 2.2.1.2.2. Passive Solute Transport

The conservative transport through the porous media is modelled with the following advection-diffusion-dispersion equation:

$$\frac{\partial C_{sub,i}}{\partial t} + \nabla \cdot \left(U_{sub,i} C_{sub,i}\right) - \nabla \cdot \left(D_{sub,ij} \nabla C_{sub,i}\right) = S_{C_{sub,i}} \tag{2.24}$$

where $C_{sub,i}$ is the subsurface concentration field (kg/m$^3$), $U_{sub,i} = q_i/\theta_i$ is the porewater velocity field (m/s), $D_{sub,ij}$ is the hydrodynamic dispersion coefficient tensor field (m$^2$/s) and $S_{C_{sub,i}}$ represents any concentration source/sink field (kg/m$^3$).

In `darcyInterTransportFoam`, $D_{sub,ij}$ can be either specified as uniform and constant through a single value or calculated as the sum of the effective molecular diffusion coefficient field, $D_{diff,ij}$ (m$^2$/s), and the mechanical dispersion coefficient field, $D_{disp,ij}$ (m$^2$/s). This choice between two alternatives represents an improvement over its predecessor, where only the second option was available, and aims to provide greater flexibility to the user in solute transport simulations. Moreover, unlike in `hyporheicScalarInterFoam`, the $D_{diff,ij}$ tensor corrects for tortuosity the user-defined molecular diffusion coefficient, $D_m$, based on Boudreau (1996), ensuring a more accurate representation of solute transport in complex subsurface environments:

$$D_{diff,ij} = \frac{D_m}{1 - 2\,ln\,\theta_i} J_{ij} \tag{2.25}$$

where $J_{ij}$ is the tensor of ones. The $D_{disp,ij}$ tensor is defined as follows (Scheidegger, 1961):

$$D_{disp,ij} = \alpha_{T,i} |U_{sub}| \delta_{ij} + \left(\alpha_{L,i} - \alpha_{T,i}\right) \frac{U_{sub,i} \times U_{sub,j}}{|U_{sub}|} \tag{2.26}$$

where $U_{sub}$ is the porewater velocity magnitude field (m/s) and $\delta_{ij}$ is the Kronecker delta function (Bear, 1961). The new model sets by default $\alpha_{T,i} = 0.1 \cdot \alpha_{L,i}$ (Cardenas, et al., 2008) if no field is specified for $\alpha_{T,i}$ at the start of the simulation.

#### 2.2.1.2.3. Heat Transfer

The new FCM solves the transport of heat at the subsurface domain assuming local thermodynamic equilibrium in the ground. Thus, both the groundwater and the porous

medium ($T_{w_{sub}}$ and $T_{pm}$, respectively) are considered to be at the same temperature $T_{w_{sub}} = T_{pm} = T_{sub}$. This assumption simplifies the definition of the heat transfer model to a single energy equation (Rubin, 1974), derived by combining the energy balance equations of both subsurface phases (i.e., water and porous medium):

$$\rho Cp_{sub,i}\frac{\partial T_{sub,i}}{\partial t} + \nabla\cdot\left(\rho_{w_{sub}} Cp_{w_{sub}} U_{sub,i} T_{sub,i}\right) - \nabla\cdot\left(K_{sub,i}\nabla T_{sub,i}\right) = \rho Cp_{sub,i} S_{T_{sub,i}} \quad (2.27)$$

where $T_{sub,i}$ is the subsurface temperature field (°C), $Cp_{w_{sub}}$ is the groundwater specific heat capacity (m²/s²·°C), $K_{sub,i}$ is the subsurface thermal conductivity field (kg·m/s³·°C) and $S_{T_{sub,i}}$ represents any heat source/sink field (°C). The overall specific heat capacity field, $\rho Cp_{sub,i}$, is calculated by weighted averaging the $\rho$ and $Cp$ of the water and the porous medium based on $\theta_i$:

$$\rho Cp_{sub,i} = \theta_i \rho_{w_{sub}} Cp_{w_{sub}} + (1-\theta_i)\rho_{pm} Cp_{pm} \quad (2.28)$$

where $\rho_{pm}$ and $Cp_{pm}$ are respectively the density (kg/m³) and specific heat capacity (m²/s²·°C) of the porous medium. $K_{sub,i}$ is also averaged in the same way as follows:

$$K_{sub,i} = \theta_i \underbrace{\frac{\rho_{w_{sub}} Cp_{w_{sub}} v_{phys_{w_{sub}}}}{Pr_{w_{sub}}}}_{K_{w_{sub}}} + (1-\theta_i)K_{pm} \quad (2.29)$$

where $v_{phys_{w_{sub}}}$ and $Pr_{w_{sub}}$ are respectively the physical kinematic viscosity (m²/s) and the Prandtl number of the subsurface water (-) and $K_{w_{sub}}$ and $K_{pm}$ the thermal conductivities of the groundwater and porous medium (kg·m/s³·°C). $\rho_{w_{sub}}$, $Cp_{w_{sub}}$, $v_{phys_{w_{sub}}}$ and $Pr_{w_{sub}}$ take by default the value of its surface counterpart but also allow the assignment of a new value by the user.

### 2.2.2. Coupling Modes Between the Two Domains

Although **darcyInterTransportFoam** has been initially designed to run in sequential iterative mode (i.e., two-way coupling), the model also supports application in asynchronous linking mode (i.e., one-way coupling). Both coupling approaches were previously available in **hyporheicScalarInterFoam**. Nevertheless, minor updates have been made to the sequential iterative coupling scheme.

#### 2.2.2.1. One-Way Coupling

The new solver, like its original version, allows the separate simulation of either of the two coupled domains. This feature enables its application to both single-domain

scenarios and one-way coupled modelling studies. In the latter case, **darcyInterTransportFoam** is first used to solve the flow and transport processes in one domain. The obtained results are then imposed as BCs on another domain, which can also be simulated with **darcyInterTransportFoam** or, more typically, another numerical model (e.g., **HydroGeoSphere**, **FEFLOW** or **ParFlow**). Once the second domain is solved, no feedback is provided to the first.

#### 2.2.2.2. Two-Way Coupling

The fully-coupled simulation of the surface and subsurface domains is achieved via the two-way mapping of conservative flux BCs at the SW-GW interface through an iterative scheme. This method enforces flux consistency across the interface, ensuring mass conservation within the system. Although the coupling strategy remains the same as in the original model, several modifications have been introduced in **darcyInterTransportFoam** to enhance the data transfer between domains. These include the code reorganization to improve readability and comprehension, the reformulation of the data mapping approach and the definition of new options for selecting the processes to simulate. Further details about the updates implemented in the two-way coupling scheme can be found in the Section 2.2.3.1 of this study.

The governing flow equations of both modelling domains are solved, sequentially and iteratively within each time step, as follows:

1. In the first iteration, $k$, the surface hydrodynamic fields, namely $U_{sur,i}^{k}$ and $p_{sur,i}^{k}$, are first solved using the Navier-Stokes equations to next infer from them $h_{sur,i}^{k}$.

2. The estimated $h_{sur,i}^{k}$ at the bottom boundary of the surface domain is transferred to the top boundary of the subsurface domain (Figure 2.2) according to:

$$\begin{cases} \nabla h_{sub,i}^{k} \cdot n_{sub,i} = 0 & U_{sur,i}^{k} \cdot n_{sur,i} < 0 \quad \Rightarrow \quad upwelling\ flux \\ h_{sub,i}^{k} = h_{sur,i}^{k} & U_{sur,i}^{k} \cdot n_{sur,i} > 0 \quad \Rightarrow \quad downwelling\ flux \end{cases} \tag{2.30}$$

   where $n_{sur,i}$ and $n_{sub,i}$ are, respectively, the outward-pointing unit normal vector field of the surface bottom and subsurface top boundaries.

3. The groundwater flow is then simulated using the saturated porous media continuity equation, which results in an estimated $h_{sub,i}^{k}$ from which an estimated $q_i^k$ is ultimately inferred.

4. The resulting $q_i^k$ at the upper boundary of the subsurface domain is subsequently mapped to the bottom boundary of the surface domain (Figure 2.2). Accordingly, in the next iteration, $k+1$, the surface bottom BC for $U_{sur,i}^{k+1}$ will depend on whether the flux is downwelling or upwelling as per:

$$\begin{cases} U_{sur,i}^{k+1} = q_i^k & q_i^k \cdot n_{sub,i} > 0 \quad \Rightarrow \quad upwelling\ flux \\ \nabla U_{sur,i}^{k+1} \cdot n_{sur,i} = 0 & q_i^k \cdot n_{sub,i} < 0 \quad \Rightarrow \quad downwelling\ flux \end{cases} \tag{2.31}$$

$p_{sur,i}^{k+1}$ will be inferred from $U_{sur,i}^{k+1}$ through a fixed flux BC. The iterative process will proceed until convergence (Figure 2.1).

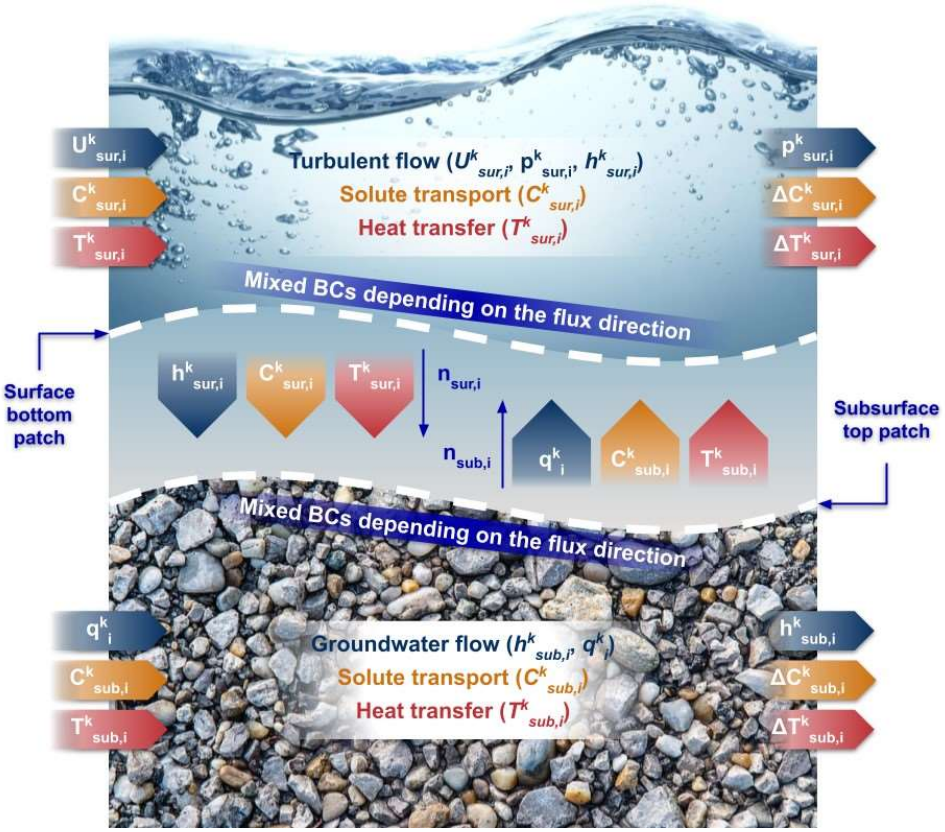



Figure 2.2. Two-way data mapping at the interface patch of the surface and subsurface meshes for the full coupling of both domains.

Once the flow field is solved, a similar iterative scheme is applied to both the solute transport and the heat transfer processes. Depending on the direction of the flux at the interface patch, the BC defined for each domain can be either a fixed value or a fixed flux (Figure 2.2). Thereby, for each iteration within a time step, the potential temperature BCs at the subsurface top boundary will be:

$$\begin{cases} \nabla T_{sub,i}^k \cdot n_{sub,i} = 0 & U_{sur,i}^k \cdot n_{sur,i} < 0 \quad \Rightarrow \quad upwelling\ flux \\ T_{sub,i}^k = T_{sur,i}^k & U_{sur,i}^k \cdot n_{sur,i} > 0 \quad \Rightarrow \quad downwelling\ flux \end{cases} \tag{2.32}$$

whereas at the surface bottom patch:

$$\begin{cases} T_{sur,i}^{k+1} = T_{sub,i}^k & q_i^k \cdot n_{sub,i} > 0 \quad \Rightarrow \quad upwelling\ flux \\ \nabla T_{sur,i}^{k+1} \cdot n_{sur,i} = 0 & q_i^k \cdot n_{sub,i} < 0 \quad \Rightarrow \quad downwelling\ flux \end{cases} \tag{2.33}$$

As happened to the flow solution (Equations (2.30) and (2.31)), the iterations will continue until the convergence is reached within each time step. The same procedure will be employed by solute transport.

### 2.2.3. Model Novelties

A brief description of the model novelties used in the subsequent test cases is provided in the following subsections, while the full set of solver updates and simulation add-ons is detailed in Section 4.2 of Chapter 4.

#### 2.2.3.1. Solver Updates

The updates implemented in the solver comprise modifications to the structure and organization of the code, along with improvements to existing functionalities and the addition of new ones.

The solver updates described below include the names of all the cited parameters in the code (i.e., files, variables, fields, dictionaries, etc.), along with the corresponding point numbers (in brackets) referring to their extended descriptions in Section 4.2.2:

1. Computation of additional hydrology-related fields, such as the specific discharge (`q`), the water depth (`dSur`, `dSub`) or hydraulic head (`hSur`, `hSub`). Accordingly, the groundwater flow is solved based on `hSub`, in contrast to the modified pressure field (`pTildeSub`) used in the original model. (1)
2. Definition of a reference for the free-surface (`hRef`) in the surface domain. This optional feature allows to specify a water height at the surface outlet patch that, combined with the appropriate BCs for pressure and velocity, enables the flow to leave the simulation domain freely. (3)
3. Possibility to set the porosity (`theta`) and the longitudinal and transverse dispersivities (`alpha_L` and `alpha_T`) as heterogeneous via scalar fields, as well as the hydraulic conductivity (`K`) as heterogeneous and anisotropic through a tensor field. This non-uniform field definition provides full flexibility in assigning the previous parameters throughout the subsurface domain, enabling, e.g., the setup of layers with diverse geologic characteristics, a heterogeneous parameter distribution or the definition of impermeable zones. (5)
4. Automatic specification of impermeable regions in the subsurface domain according to the shape and extension of the surface dry areas. The depth (`d_imper`, must be always negative), the geological properties (`K_imper`, `theta_imper`, `alpha_L_imper` and `alpha_T_imper`) as well as the solute concentration (`CSub_imper`) and water temperature (`TSub_imper`) of such

impervious regions have to be specified by the user, who can optionally choose whether or not to apply this new modelling feature (`imperRegion`). Its use is limited to quasi-steady-state scenarios in which the dry areas of the surface bottom patch remain constant throughout the entire simulation. Applying it to transient scenarios would result in unrealistic behaviour, as steep hydraulic gradients would be generated at the newly permeable cells each timestep. (6)

5. Simplification of the scheme used to select the processes to be solved by the model. As a result, the choice of the flow, the solute transport as well as the heat transfer simulation can be made jointly for both surface and subsurface domains (`flow`, `soluteTransport` and `heatTransfer` options in the *simulationProperties* dictionary, respectively). Furthermore, as an alternative to this joint selection, the simulation of each process can be also separately specified for each domain. The different process selection options as well as their corresponding implications can be checked in **Figure 2.1**. (7)
6. Full reformulation of the solute transport module, including both the code organization and the modelling approach. Regarding the later, both the surface diffusion and the subsurface hydrodynamic dispersion fields (`DSur` and `DSub`, respectively) can be either computed through new or partially modified equations or specified as uniform throughout the domain via a single value (`constDSur` or `constDSub`). Moreover, the transport of conservative solutes (`CSur`) can be optionally solved in one or two phases at the surface domain (`solutePhases`). All the solute transport related expressions are gathered within the corresponding header files *CSurEqn.H* and *CSubEqn.H* in the surface and subsurface domains. A thorough description of the novel transport equations is provided in **Sections 2.2.1.1.2** and **2.2.1.2.2**. (8)
7. Addition of a heat transfer module for the simulation of the temperature distribution in both surface and subsurface domains. Both the code organization and the modelling scheme of this new module are based on those implemented in its solute transport counterpart. Consequently, two phase-based simulation options (`heatPhases`) are included in the *TSurEqn.H* file to simulate the transfer of heat in the surface domain (`TSur`), whereas the energy equation required to resolve the temperature distribution in the ground (`TSub`) is contained in *TSubEqn.H*. Further details about the implemented heat transfer equations can be found in **Sections 2.2.1.1.3** and **2.2.1.2.3**. (9)
8. Modification of two-way data mapping (**Section 2.2.2.2**) for a more accurate conveyance of the information between the domains. In this way, the data is no longer interpolated but directly assigned to the matching nodes of both meshes at their respective interface patch. This new method ensures the exact transmission

of the solution from one domain to another (`hSur`, `q`, `CSur`/`CSub` and `TSur`/`TSub`). (10)

9. Reformulation of the convergence check at the SW-GW interface. Rather than relying on a fixed maximum number of correction loops, water and solute flux consistencies are evaluated at each time step by comparing the exchange fluxes predicted from the surface domain with those computed from the groundwater solution. Convergence is considered achieved when both water and solute flux mismatches (`couplePhiMaxMismatch` and `coupleSolutePhiMaxMismatch`) fall below the specified tolerances (`couplePhiTol` and `couplePhiSoluteTol`) (**Figure 2.1**). (11)

#### 2.2.3.2. Pre- and Post-processing Utilities

Four new utilities were developed to be used with **`darcyInterTransportFoam`**. Additionally, some of them are also compatible with other OpenFOAM solvers. The complete new set was used to perform pre- and post-processing actions in the implemented test cases:

- **`checkMeshesMatch`**: checks whether the number of faces of both meshes and their coordinates match at the interface patch. The fulfilment of both conditions ensures the appropriate fitting of both surface and subsurface meshes. Only applicable to **OpenFOAM** models consisting of two adjacent simulation meshes.
- **`changeZRefMesh`**: performs two optional actions: (1) translation of the mesh points according to a user-specified reference mesh elevation; and (2) adjustment of `hRef` to the actual mesh location. Both actions are independent of each other. Applicable to further **OpenFOAM** models.
- **`restoreZRefMesh`**: initially designed to be used in combination with **`changeZRefMesh`** but also applicable on its own. This utility conducts three optional and independent actions: (1) translation of the mesh points according to the `meshShiftZ` value; (2) reestablishment of the original, user-defined `hRef` (if previously applied **`changeZRefMesh`**); and (3) adjustment of the computed hydrological fields (`piezoSur`, `hSur` and `hSub`) based on `meshShiftZ`. `meshShiftZ` can be either previously determined by **`changeZRefMesh`** or provided by the user in the *constant* directory. Applicable beyond **`darcyInterTransportFoam`**.
- **`transferMeshDecomp`**: transfers the domain decomposition from one mesh to another through the interface patch of both meshes. A prior decomposition of both domains (**`decomposePar`** utility) is required to generate the necessary *cellDist* and *cellDecomposition* files for its execution. Furthermore, it optionally overwrites the *cellDecomposition* file of the reference and/or target mesh once the transfer is

done. Only applicable to **OpenFOAM** models consisting of two adjacent modelling meshes.

#### 2.2.3.3. Surface and Subsurface Boundary Conditions

One new boundary condition for $U_{sur,i}$ and another two for $h_{sub,i}$ were implemented along with the new model. Analogously to the novel utilities, some of the new BCs are **darcyInterTransportFoam**-specific while others can also be applied to other **OpenFOAM** solvers. The two $h_{sub,i}$ boundary conditions were used in the test cases:

- **darcyGradHead**: sets at the patch the $\nabla h_{sub,i}$ resulting from the defined $q_i$ BC. It is based on the previously developed **darcyGradPressure** BC (Horgue, et al., 2015). Applicable to **OpenFOAM** models other than **darcyInterTransportFoam**.
- **surfaceHydrostaticHead**: specifies at the inlet, outlet or aquifer (i.e., bottom) patch of the subsurface domain the hydrostatic $h_{sub,i}$ computed at the equivalent surface boundary. Accordingly, single values will be assigned to the subsurface inlet and outlet patches ($max(piezo_{sur,i})$), whereas an entire head field will be prescribed to the bottom boundary of the subsurface domain ($piezo_{sur,i}$). In this early version, its use is restricted to the above-mentioned boundaries. Only applicable to **darcyInterTransportFoam**.

## 2.3. Test Cases

The performance and capabilities of **darcyInterTransportFoam** are demonstrated through two 3D test cases, where the SW-GW flow, solute transport and heat transfer processes are simulated across two topographically distinct segments of an actual, alluvial river-aquifer system. A small description of the simulation domains, the simulated hydraulic scenarios and details of the overall modelling setup are provided below.

### 2.3.1. Simulation Domains

To ensure the necessity of fully-coupled modelling for capturing the full dynamics of the system, a field site characterized by high $K_{ij}$ values was selected to conduct the evaluation test cases. The simulation domains focus on two segments of the Emme River near Aeschau in the Emmental region of Switzerland (**Figure 2.3**). At this location, the river features a coarse gravel bed and several small weirs designed for erosion control. The riverbed is underlain by quaternary alluvium composed of coarse sandy gravel with a variable proportion of silt (Käser & Hunkeler, 2016; Peel, et al., 2022). This geology

creates an extremely conductive unconfined aquifer (Schilling, et al., 2017a). The aquifer is limited underneath by impermeable sediments, forming an aquitard relative to the overlying alluvium (Blau & Muchenberger, 1997). In terms of $K_{ij}$ and $\theta_i$, previous studies have reported depth-averaged $K_{ij}$ values ranging from $2.3 \cdot 10^{-3}$ to $7 \cdot 10^{-3}$ m/s and $\theta_i$ values between 0.1 and 0.3 (Schilling, et al., 2022; Würsten, 1991).

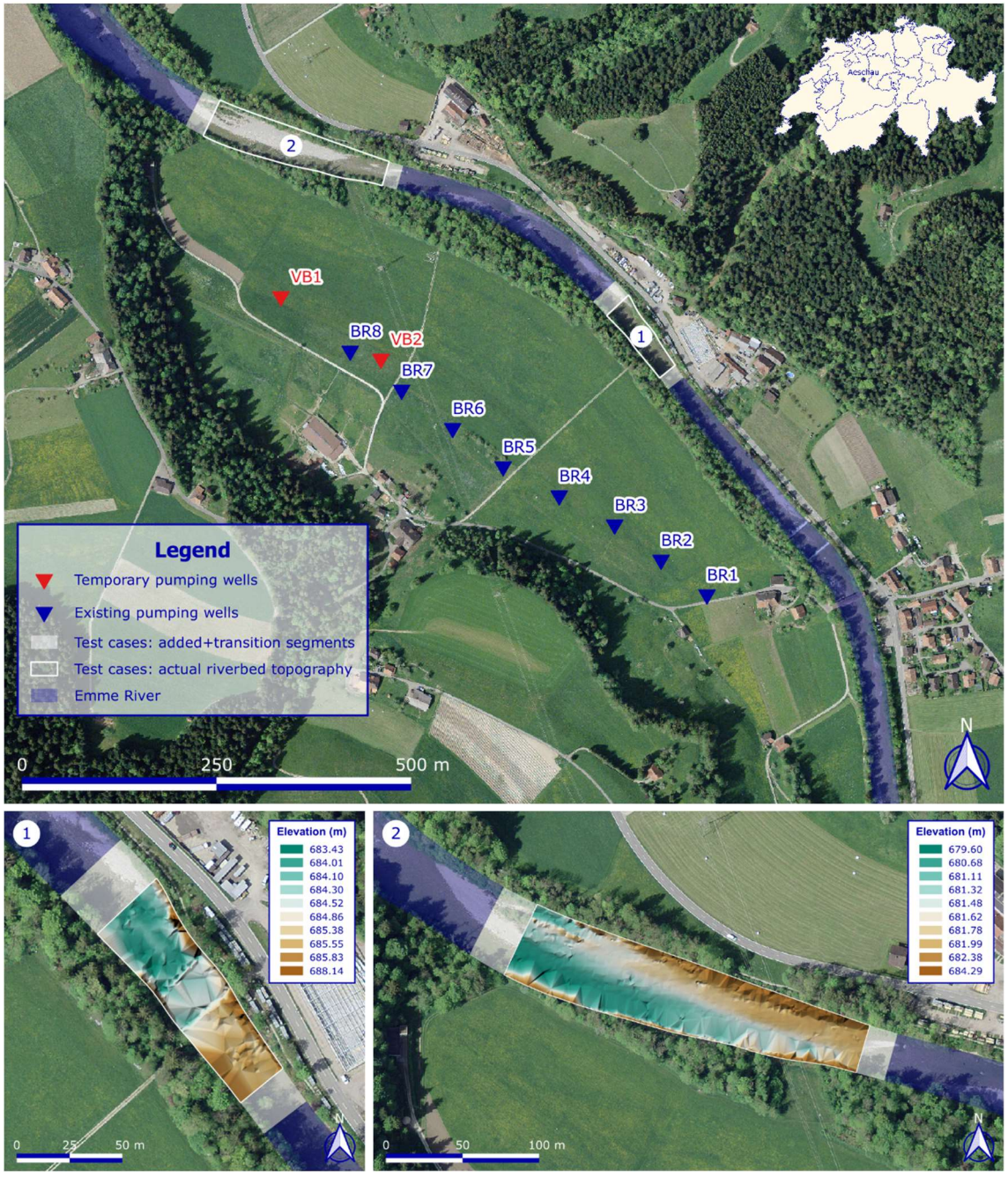



Figure 2.3. The top image shows the location of the model's test cases in the Emme River, while the bottom images display the high-resolution (25 cm) riverbed DEM used to define the 3D meshes of each test case: (1) Test Case 1: River Weirs (the structure of the weirs was later burned on the DEM after the on-site measurement of its dimensions); (2) Test Case 2: Groundwater Injection Jet.

The high permeability of the alluvium results in continuous SW-GW interactions across the riverbed (Tang, et al., 2018). These exchanges significantly impact the groundwater mixing ratios and travel times, particularly during high-flow events, when the river discharge can account for up to 90% of the total aquifer recharge (Poffet, 2011; Popp, et al., 2021). Together, the described geological and hydraulic characteristics create a highly dynamic river-aquifer system, ideal for testing and evaluating the new fully-coupled SW-GW **OpenFOAM** model. Further details on the field site characteristics can be found in **Section B.1** of **Appendix B**.

### 2.3.2. Hydraulic Scenarios

#### 2.3.2.1. Test Case 1: River Weirs

Numerous small weirs designed for erosion control are placed all along the Emme River (Käser & Hunkeler, 2016). Two couples of these hydraulic structures can be found in the vicinity of Aeschau, being one of them chosen to carry out Test Case 1 (TC1) (**Figure 2.4**). This test case aims to demonstrate the capability of `darcyInterTransportFoam` to capture the physical and geological characteristics of real sites, such as weir pillars, and thus assess their impact on the system hydrodynamics. The upstream weir is 30 m long, 0.85 m wide and 3.4 m deep, whereas the downstream weir is 30 m long, 0.5 m wide and 2.1 m deep. Both weirs are followed by a deep hole right after the structure, which helps prevent bed erosion and contributes to the dissipation of energy. Although the upper weir has on-site a ramp in the middle, this feature is not modelled for simplicity.

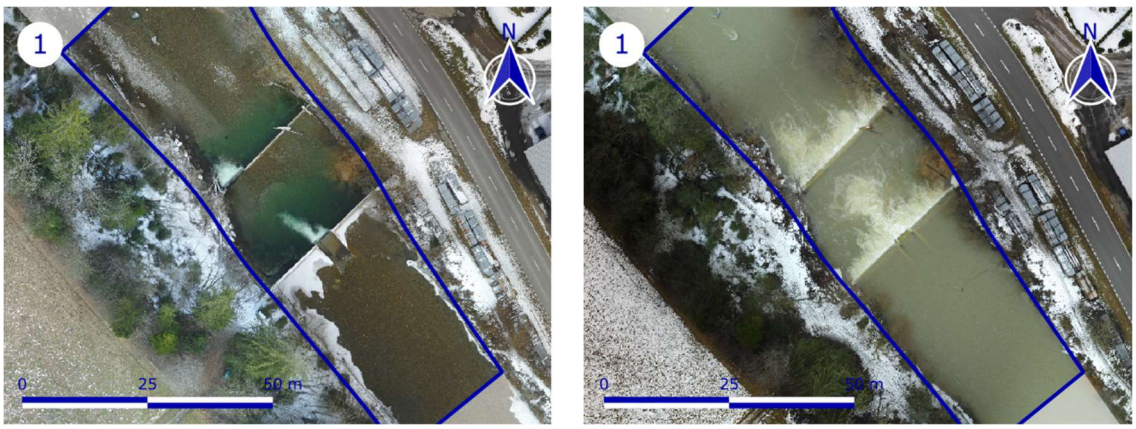



**Figure 2.4. View of the simulated couple of weirs of TC1 under different flow regimes.**

#### 2.3.2.2. Test Case 2: Groundwater Injection Jet

A 8-well abstraction plant (BR1 to BR8) is situated on the extensive floodplain along the river's left bank (Blanc, et al., 2024; Schilling, et al., 2017a; Schilling, et al., 2022)

(Figure 2.3). At the end of 2018, two temporary vertical wells (VB1 and VB2) were drilled in the vicinity of this existing wellfield to conduct a major, 1.5-month pumping test (Peel, et al., 2022; Popp, et al., 2021). Throughout the test, all or part of the water pumped from the two newly installed wells was discharged into the Emme via several pipes (Figure 2.5). The mean river temperature during the experiment was 2 °C, while the temperature of the abstracted groundwater averaged 8 °C, coinciding with the mean annual air temperature at the site (Figura, et al., 2013). Both surface and subsurface temperatures were obtained from point measurements taken on-site. The resulting plume from the temperature contrast, along with the heat exchange results from previous studies at the site, provides an ideal scenario to test the newly implemented heat transfer module of `darcyInterTransportFoam`. Consequently, this scenario is selected for Test Case 2 (TC2). Furthermore, although no solute measurements were taken in the river during the pumping test, the solute exchanges are also simulated in this test case, assuming similar interaction dynamics as heat across the SW-GW interface.

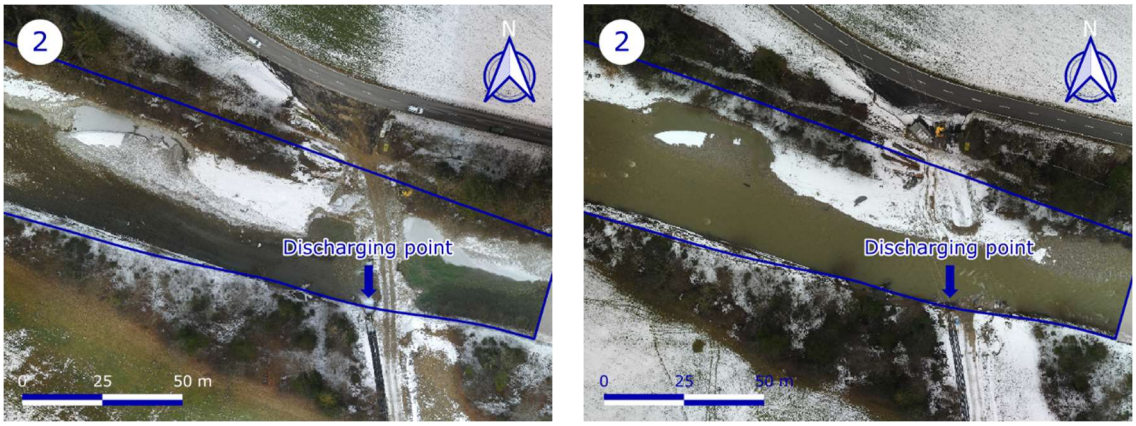



Figure 2.5. Snapshot of the pipes releasing pumped groundwater in the Emme River at two different flow scenarios.

### 2.3.3. General Modelling Setup

#### 2.3.3.1. Mesh Features

The surface and subsurface meshes for both test cases were generated using `blockMesh`, one of the mesh generation utilities of **OpenFOAM**. Accordingly, the resulting meshes consist of 3D, structured, hexahedral blocks defined by either straight or curved edges. To enable full control over the mesh generation process, the *blockMeshDict* file required for running `blockMesh` was generated for each mesh through the combined work of QGIS and **MATLAB** (The MathWorks Inc., 2024). As shown in Figure 2.6, QGIS was first used to define the extension of the computational domain as well as the longitudinal mesh resolution. Next, the defined outline points were imported into a set of user-defined **MATLAB** scripts (Section 4.3), that generated the *blockMeshDict* file based

on the aforementioned geometric features and additional mesh data (e.g., the riverbed DEM, mesh height, inlet/outlet shape, etc.). These scripts were also used to generate other mesh-related dictionaries (e.g., *topoSetDict*) and input data files (e.g., *setFieldsDict*) associated with the resulting structured meshes.

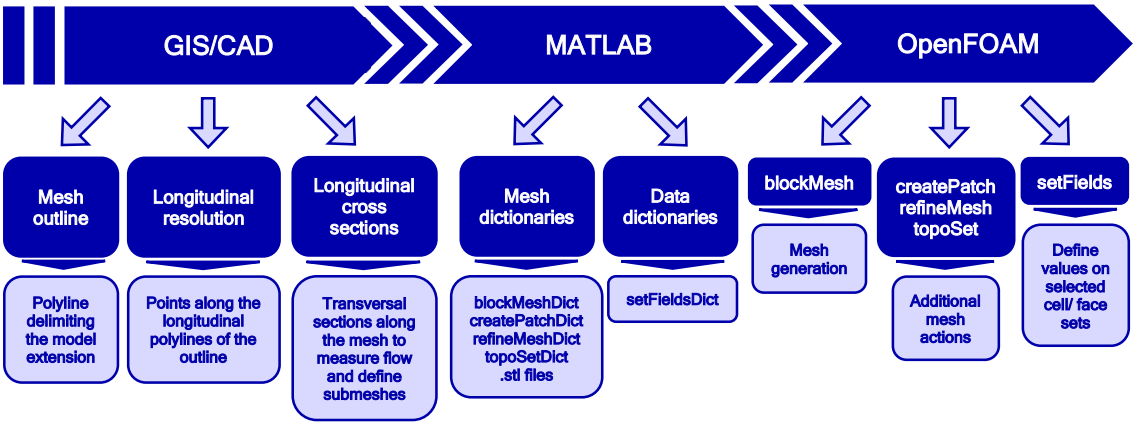



Figure 2.6. Mesh generation workflow.

A high-resolution (0.25 m) Digital Elevation Model (DEM), acquired on 20 March 2015 using drone-based photogrammetry (Schilling, et al., 2017a), was used to define the interface boundary of both domains (i.e., the riverbed patch) (Figure 2.3). The surface and subsurface meshes are approximately 2.5 m and 10 m high, respectively, in the two test cases and both are around 106 m long and 30 m wide in TC1 (Figure 2.3-1) and about 235 m long and 50 m wide in TC2 (Figure 2.3-2). However, rectangular inflow and outflow stretches of around 6.25 m and 12.5 m long, respectively, were added to each mesh to prevent potential perturbations caused by the geometric features of the inlet and outlet boundaries. Moreover, the smooth transition between these regular sections and the actual riverbed topography was achieved through approximately 12.5 m long segments. Consequently, the total length of each mesh was increased by around 43.75 m (Figure 2.3). The other two dimensions remained unaltered.

Table 2.1. Number of mesh cells of both test cases (prior to the double refinement around the weirs in TC1).

| | | Test Case 1 | | Test Case 2 | |
|---|---|---|---|---|---|
| | | Surface | Subsurface | Surface | Subsurface |
| OpenFOAM | Total | 1531557 | 4613819 | 2090610 | 8461230 |
| MATLAB | Longitudinal direction | 600 | | 1115 | |
| | Transversal direction | 125 | | 178 | |
| | Vertical direction | 12 | 41 | 11 | 42 |

As for the mesh discretization, a mesh sensitivity study was conducted to determine the optimal resolution for each mesh, leading to the selection of a surface mesh with approximately $10^6$ cells for TC1 and $2 \cdot 10^6$ for TC2 and a subsurface mesh with about $4 \cdot 10^6$ elements for TC1 and $8 \cdot 10^6$ for TC2 (Table 2.1). Identical cell counts in the longitudinal and transversal directions are imposed for both domains to ensure effective interface node matching and, thus, successful coupling of their solutions.

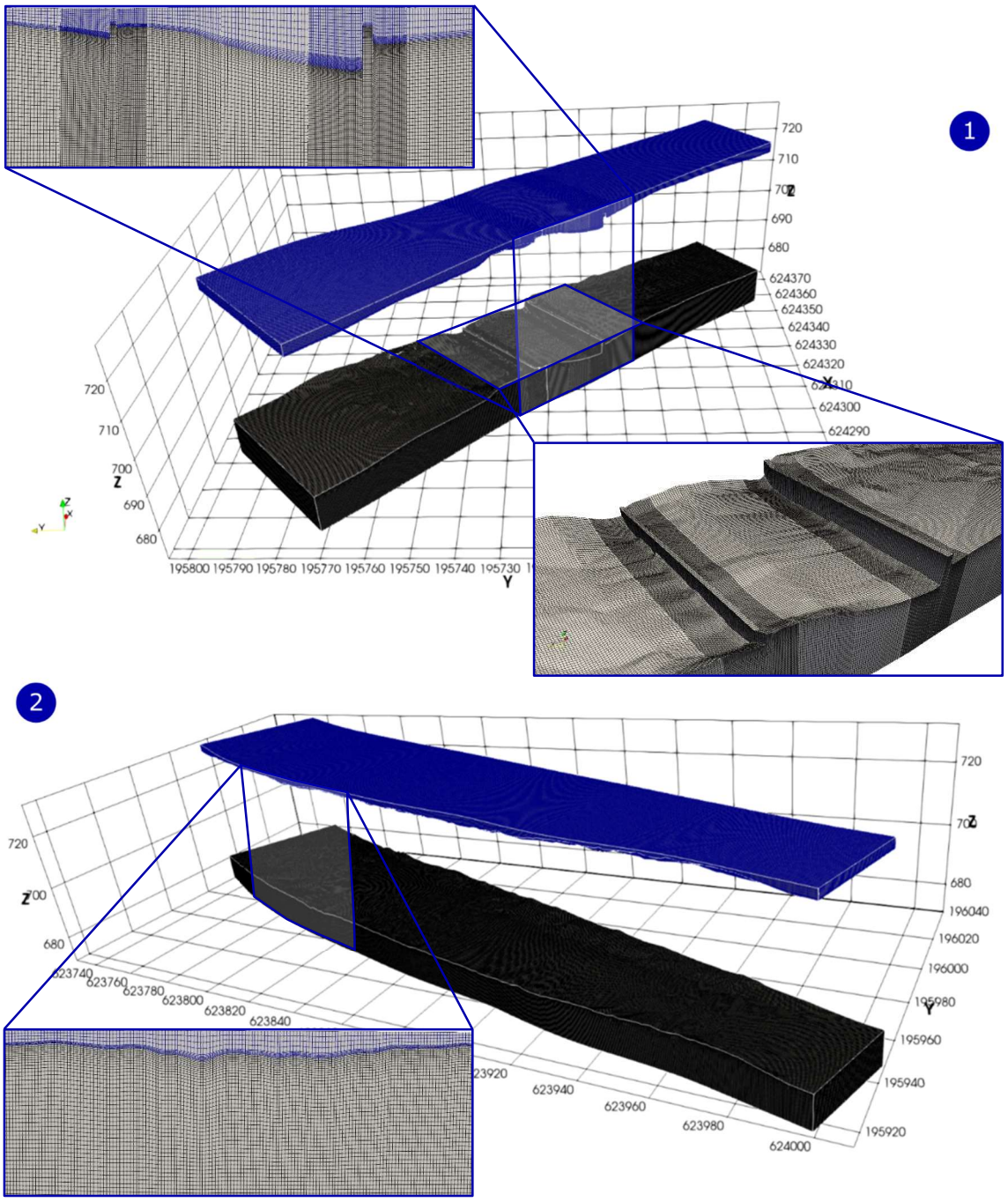



Figure 2.7. Simulation meshes of TC1 and TC2 generated with `blockMesh`.

Moreover, the meshes were gradually refined vertically towards the SW-GW interface to capture the complex dynamics across this boundary and satisfy the y+ (dimensionless wall distance) requirement near the riverbed in the surface domain (**Figure 2.7**). Additionally, the region surrounding the weirs in TC1 was subject of a double refinement in all three directions (**Figure 2.7-1**). In both meshes, the smallest elements are located close to the interface (TC1 = $1.59 \cdot 10^{-7}$ m$^2$, TC2 = 0.001 m$^2$ face area), while the largest cells are situated within the air-phase in the surface domain (TC1 = 0.18 m$^2$, TC2 = 0.24 m$^2$) and at the bottom boundary in the subsurface domain (TC1 = 0.13 m$^2$ , TC2 = 0.1 m$^2$). Especially large element sizes were defined for the surface air-phase, since detailed processes in this region are not of interest for the present study. This phase is only simulated to account for the water level fluctuations.

#### 2.3.3.2. Model Parametrization

The same fluid and geological parametrization was prescribed to both test cases. All fluid parameters were assumed to stay constant over time in agreement with the simulated steady-state flow and transport dynamics. Consequently, the simulated phases (i.e., air and water) were considered Newtonian. Common values of $\nu_{phys_p}$ and $\rho_p$ at ambient temperature were defined for the air and water (**Table 2.2**). Regarding the solute transport parameters, a $D_{phys}$ of $4.2 \cdot 10^{-10}$ m$^2$/s, corresponding to fluorescein sodium (NaFl), and a $Sc_{turb}$ of 1 were applied to both surface fluids. $D_m$ was assumed to be equal to $D_{phys}$. Moreover, heat parameters, such as $Pr_p$, were inferred from the fluid temperatures specified for each domain. Further details on the fluid properties are provided in **Table 2.2**.

Regarding the geological properties, two layers of different homogeneous and isotropic $K_{ij}$ and $\theta_i$ were defined in the subsurface domain (**Figure 2.8**). The upper layer, located at the first 2 m beneath the riverbed, has a $K_{ij}$ of $2.8 \cdot 10^{-3}$ m/s and a $\theta_i$ of 0.41. These values were assigned via a *setFieldsDict* using a triangulated surface geometry. The rest of the aquifer corresponds to the second layer, whose $K_{ij}$ and $\theta_i$ are $6.4 \cdot 10^{-3}$ m/s and 0.1, respectively. Moreover, impermeable regions were defined below the surface dry areas (i.e. the riverbanks and several gravel bars). The depth of such impervious zones as well as the method applied to define them differs between test cases. Whereas the new simulation option `imperRegion` (**Section 2.2.3.1**) was used in TC1 to specify as impermeable all the subsurface cells in the vertical, eight regions of different depths (from 0.25 to 1 m) were defined in the second by means of a triangulated surface geometry using the water table height as reference (**Figure 2.8**). Within these regions, the values of $K_{ij}$ and $\theta_i$ are small enough to ensure the flow cannot pass through them. Thereby, $K_{ij}$ is set to $10^{-20}$ m/s and $\theta_i$ to $10^{-3}$.

Table 2.2. Surface and subsurface (purple) fluid and geological parameters defined in the test cases.

| | Phases | Parameter | Value | | Dimensions |
|---|---|---|---|---|---|
| Fluid parameters | Water | Transport model | Newtonian | | |
| | | Physical kinematic viscosity ($\nu_{phys_w}$) | $10^{-6}$ | | $m^2/s$ |
| | | Density ($\rho_w$) | 1000 | | $kg/m^3$ |
| | | Specific heat capacity ($Cp_w$) | 4211 | 4196 | J/(kg·°C) |
| | | Mean temperature ($T_w$) | 2 | 8 | °C |
| | | Prandtl number ($Pr_w$) | 12.46 | 9.99 | - |
| | Air | Transport model | Newtonian | | |
| | | Physical kinematic viscosity ($\nu_{phys_a}$) | $1.48 \cdot 10^{-5}$ | | $m^2/s$ |
| | | Density ($\rho_a$) | 1 | | $kg/m^3$ |
| | | Specific heat capacity ($Cp_a$) | 1005 | | J/(kg·°C) |
| | | Mean temperature ($T_a$) | -2 | | °C |
| | | Prandtl number ($Pr_a$) | 0.72 | | - |
| | Both | Surface tension ($\sigma$) | 0.07 | | $J/m^2$ |
| | | Molecular diffusion coefficient ($D_{phys}$, $D_m$) | $4.2 \cdot 10^{-10}$ | | $m^2/s$ |
| | | Turbulent Schmidt number ($Sc_{turb}$) | 1 | | - |
| Geological parameters | | Density ($\rho_{pm}$) | 2600 | | $kg/m^3$ |
| | | Specific heat capacity ($Cp_{pm}$) | 782 | | J/(kg·°C) |
| | | Thermal conductivity ($K_{pm}$) | 2.9 | | $(kg \cdot m)/(s^3 \cdot °C)$ |

Additional geological parameters, such as $\alpha_{L,i}$ and $\alpha_{T,i}$, were also prescribed at the aquifer. In this way, a different homogeneous $\alpha_{L,i}$ was assigned to each subsurface region. Particularly, $\alpha_{L,i}$ of 3 and 5 m were set, respectively, to the upper and lower layers, while a field of $10^{-7}$ m was prescribed to the impervious zones. $\alpha_{T,i}$ was approximated by 1/10 of $\alpha_{L,i}$. The impermeable parametrization was also assigned to the weirs in TC1. The rest of the ground parameters can be found in Table 2.2.

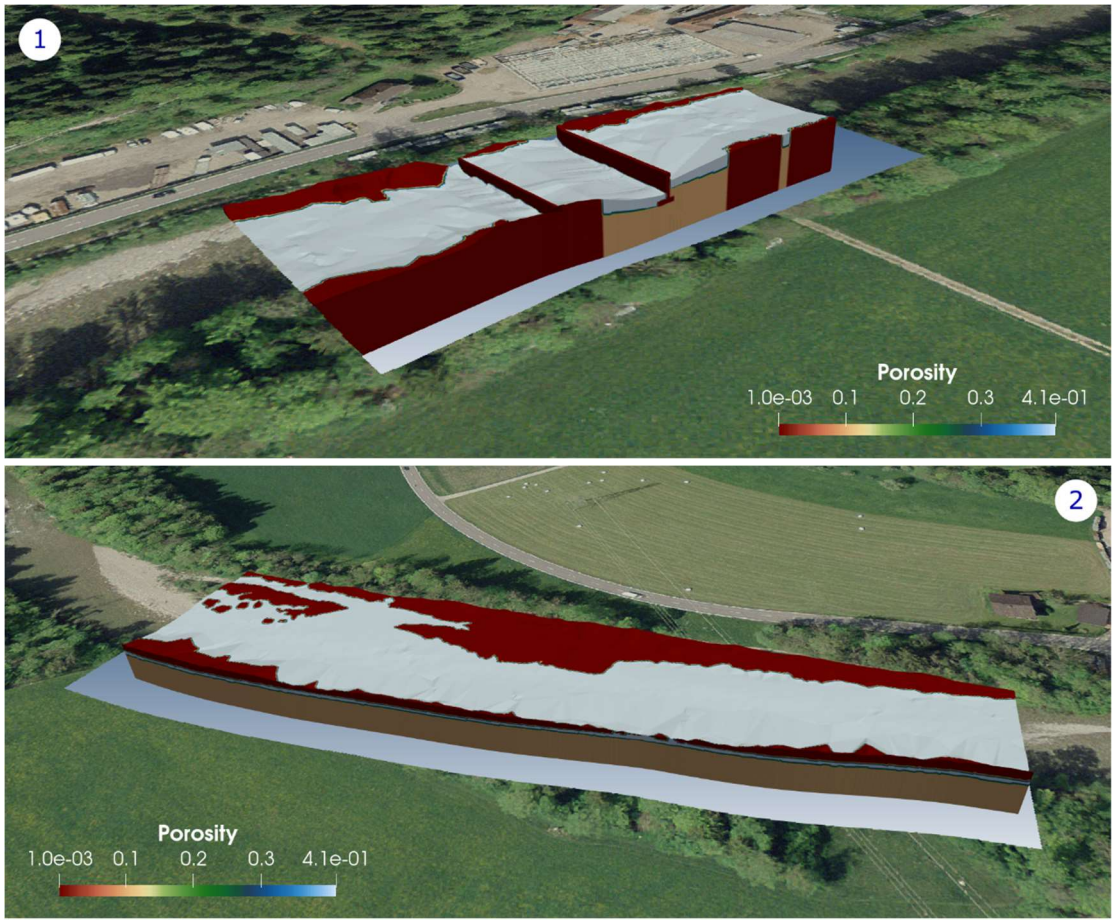



Figure 2.8. Subsurface geological characterization of TC1 and TC2.

#### 2.3.3.3. Initial and Boundary Conditions

All surface and subsurface fields were initialized with values as close as possible to the expected solution to help the models reach the steady state faster. Some values were obtained from a spin-up run, and thus directly specified in the field data files for the entire simulation domain (Table B.1, Table B.2 and Table B.3), while others were assigned to specific mesh locations through different `setFields` actions. The latter was the case of the surface fields alpha.water ($\alpha_i$), U ($U_{sur,i}$), C ($C_{sur,i}$) and T ($T_{sur,i}$).

On the other hand, as it happened with the fluid and geologic properties, all the boundary conditions prescribed to both test cases remained constant during the simulation to represent a steady-state scenario. The inlet of the surface domain is split into two patches: the lower one aimed for the water phase (`inletWater`) and the upper for the air phase (`inletAir`). $\alpha_i$ is fixed accordingly at both patches. As such, water enters the domain through the lower patch via a specified constant discharge (10 $m^3/s$), whereas the upper half is reserved for the air flow by means of a total pressure condition. The latter condition is also applied to the top boundary (`atmosphere`). Additionally, water

also flows into the domain through the `waterJet` patch in TC2 (0.42 $m^3/s$). The water level at the surface `outlet` (0.25 m) is fixed by combining a hydrostatic pressure condition with a reference free-surface elevation (`hRef`). Such water surface elevation was obtained applying Manning's equation to the rectangular section of the outlet patch. The same procedure was used to define the `inletWater` height. At the `riverbed`, $U_{sur,i}$ is initially set to 0 m/s, switching to a heterogeneous field after the first coupling iteration. A zero-velocity value is also assigned to the `steps` boundary in TC1 through a no slip condition. The impact of the riverbed roughness in the flow dynamics is also considered through the application of a rough wall function to $\nu_{turb,i}$ at this patch. As for the subsurface domain, the `inlet` and `outlet` boundaries are prescribed with an upstream $q_i$ of $2 \cdot 10^{-5}$ m/s and a downstream hydrostatic $h_{sub,i}$ condition, respectively. Furthermore, $h_{sub,i}$ is originally fixed to zero at the `riverbed`, pending the first iteration of the coupling scheme to change its value. Regarding the solute transport and heat transfer simulation, constant input values of 5 $kg/m^3$ and 2 °C are set at the `inletWater` patch and 10 $kg/m^3$ and 8 °C at the `waterJet` and subsurface `inlet` boundaries, whereas both `inletAir` and `atmosphere` patches are prescribed with a zero-gradient outflow condition with specified input for the case of reversing flow. Similar mixed BCs are defined at the `riverbed` in the two domains, mapping the values from the coupled domain in case of inflow. Both surface and subsurface `outlet` and `steps` patches are set to a zero-gradient condition. The remaining boundaries, namely `banks` and `aquifer`, are specified as impermeable using a slip condition. As such, the flow can move along the patch but not across it.

The full suite of boundary conditions prescribed for both test cases, together with a small description of each of them, is detailed in **Appendix B**. For more detailed information on their use and implementation, refer to the online **OpenFOAM** User's Guide (OpenCFD Ltd, 2023).

#### 2.3.3.4. Numerical Schemes, Linear Solvers and Solution Algorithms

The terms of the partial differential equations governing the flow, solute transport and heat transfer are transformed into algebraic equations through different spatial and temporal discretization schemes. In **`darcyInterTransportFoam`**, the discretization scheme for each term is specified in the *fvSchemes* dictionary of each domain. The same set of schemes was prescribed for both test cases. In this way, a Euler scheme was used to discretize the temporal terms of the surface and subsurface equations. The advection terms were discretized through different schemes according to the equation to be solved in each domain. These include, e.g., a Gauss linearUpwind scheme for the advective component of the Navier-Stokes equations or a limited version of the Gauss linear scheme for the advection of heat. As for the diffusion terms, these were discretized in

both domains via a non-orthogonality corrected version of the Gauss linear scheme. The rest of defined discretization schemes can be checked in Table 2.3.

Table 2.3. Numerical schemes selected to discretize the surface and subsurface equation terms.

| Scheme types | Equation terms | Numerical schemes |
|---|---|---|
| ddtSchemes | default | Euler |
| gradSchemes | default | Gauss linear |
| divSchemes | default | Gauss linear |
| | div(rhoPhi,U) | Gauss linearUpwind grad(U) |
| | div(rhoCpPhi,T) | Gauss limitedLinear 1 |
| | div(phi,C) | Gauss upwind |
| | div(phi,alpha) | Gauss vanLeer |
| | div(phirb,alpha) | Gauss interfaceCompression |
| | div(phi,k) | Gauss upwind |
| | div(phi,omega) | Gauss upwind |
| | div(((rho*nuEff)*dev2(T(grad(U))))) | Gauss linear |
| laplacianSchemes | default | Gauss linear corrected |
| interpolationSchemes | default | linear |
| snGradSchemes | default | corrected |

The discretized equations are then solved using a range of iterative linear-solvers. The selected linear-solver, along with the preconditioner and tolerance needed to achieve an accurate solution, is defined for each dynamic variable in the *fvSolution* dictionary of each domain. Analogously to the numerical schemes, both test cases applied the same solution techniques to solve the systems of discretized equations. The resulting linear systems for the momentum, turbulence and solute transport equations were solved iteratively using a run-time selectable smoother (`symGaussSeidel`). Moreover, the discretized Poisson equation and groundwater continuity equation were solved with a preconditioned (bi-)conjugate gradient method for symmetric matrices (`PCG`), whereas the linearized heat transfer equation used the same solver for asymmetric matrices (`PBiCG`) in both domains. The entire setup of the solution control is shown in Table 2.4.

In the surface domain, the 3D velocities and pressure are solved iteratively using the PIMPLE pressure-velocity coupling algorithm (OpenCFD Ltd, 2023). This iterative procedure, which combines the PISO (Pressure-Implicit Split-Operator) and the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) algorithms (Issa, 1986; Patankar & Spalding, 1972), is necessary for the VOF method to effectively solve the pressure field while satisfying the continuity equation. The pressure correction is achieved via the

Poisson equation, with convergence attained through 3 inner-correctors (`nCorrectors`) and 1 outer-corrector (`nOuterCorrectors`) in both test cases. As for the subsurface solution, the hydraulic heads are corrected for mesh non-orthogonality using 1 non-orthogonal corrector (`nNonOrthogonalCorrectors`). All correctors are specified in the *fvSolution* dictionary of each domain.

Table 2.4. Numerical solvers selected to solve the surface and subsurface (purple) variables.

| | p_rgh | h | U | alpha.water | k | omega | C | T |
|---|---|---|---|---|---|---|---|---|
| solver | PCG | | smoothSolver | | | | | PBiCG |
| preconditioner/smoother | DIC | | symGaussSeidel | | | | | DILU |
| tolerance | $10^{-7}$ | $10^{-6}$ | $10^{-8}$ | | | | $10^{-6}$ | $10^{-7}$ |
| relative tolerance (relTol) | 0.05 | 0 | 0 | | | | | 0 |

The model setup is completed with the definition of the simulation control options in the *controlDict* dictionary. The initial timestep (`deltaT`) was set to 0.001 seconds, allowing a maximum timestep of 1 second (`maxDeltaT`). Moreover, the timestep size was dynamically adjusted (`adjustTimeStep`) in both test cases according to a specified maximum Courant number of 0.5 (`maxCo`) and a maximum alpha Courant number of 1 (`maxAlphaCo`).

## 2.4. Results

The two test cases are defined as 3D, partially open systems. Consequently, water fluxes enter and exit the modelling domains only through the upstream and downstream boundaries, while the riverbanks and aquifer bottom are considered impermeable. This simplified assumption improves upon the 2D, closed model setups of Li, et al. (2020) and Lee, et al. (2021) by better capturing the dynamic behaviour of a natural river. Furthermore, the defined systems align with the need expressed by Li, et al. (2020) to further study the performance of FCMs in open systems.

Based on the fact that the original model, **`hyporheicScalarInterFoam`**, already proved its efficacy to tackle the hydrodynamics of the hyporheic zone, the reach-scale interfacial exchanges are shown this time, together with their overall impact in the whole system dynamics. The selected scale aims to demonstrate the potential of the new model to also untangle the flow dynamics and transport processes at larger hydrological contexts than the hyporheic region.

Since both the conservative transport and heat transfer do not influence the flow, the steady, pre-simulated flow field was used throughout an intermediate period of the

simulations (area coloured in the graphs) to speed-up the computational times of both processes.

A selection of the most representative results from both test cases is presented in this section. The simulated flow exchanges across the riverbed and associated groundwater flowpaths, along with their impact on the solute concentration and water temperature, are shown and discussed below.

### 2.4.1. Exchange Water Fluxes Across the Riverbed

Figure 2.9 and Figure 2.10 illustrate, respectively, the groundwater flowpaths of TC1 and TC2. As observed in the images, both direction and length of the paths differ between cases. Whereas in TC1 the water flows prominently along the longitudinal direction, dodging both the physical riverbanks and the weirs, in TC2 the flowpaths cross diagonally the domain, moving from the right to the left bank as they travel through the aquifer. Moreover, in the latter case, the paths do not avoid the natural riverbanks, but flow underneath as a consequence of the subsurface characterization (Section 2.3.3.2).

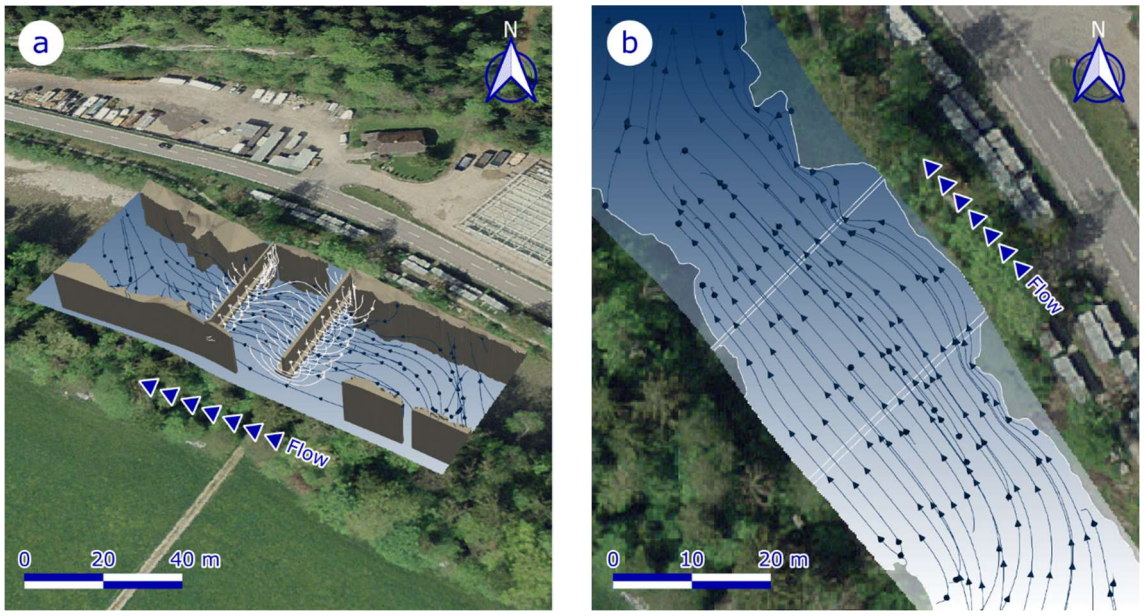



Figure 2.9. TC1 – Simulated groundwater flowpaths underneath the riverbed (blue) and the river steps (white): (a) 3D view, (b) 2D top-view.

The length of the resulting flowpaths is directly related to the described directions in each system. In this way, maximum lengths of 110 m and minimum of 10 m can be observed in TC1. As expected, the longest flowpaths align with the length of the simulation domain (Section 2.3.2.1) in agreement with their main direction within the aquifer. The shortest paths surround the weirs as a result of the steep gradient created by such structures. In TC2 the lengths are more diverse, being possible to find values ranging from 5 to 170 m. Both physical and geological features of the system explain this

fact. In contrast to TC1, the topography of TC2 presents a significant height variation along its cross-section. As a result of this, a shallow channel is formed on the left side of the riverbed (Figure 2.3-2). Such topographical difference causes a head gradient that drives most of the groundwater flowpaths of the domain. Rather than flowing straight, the paths change their direction towards the left bank of the river. This fact, along with the permeable riverbanks, generally makes the flow travel longer distances than if moved longitudinally under the course of the river. Furthermore, the same system characteristics favour the existence of a broad set of smaller path lengths generated by the multiple gradients produced across the riverbed.

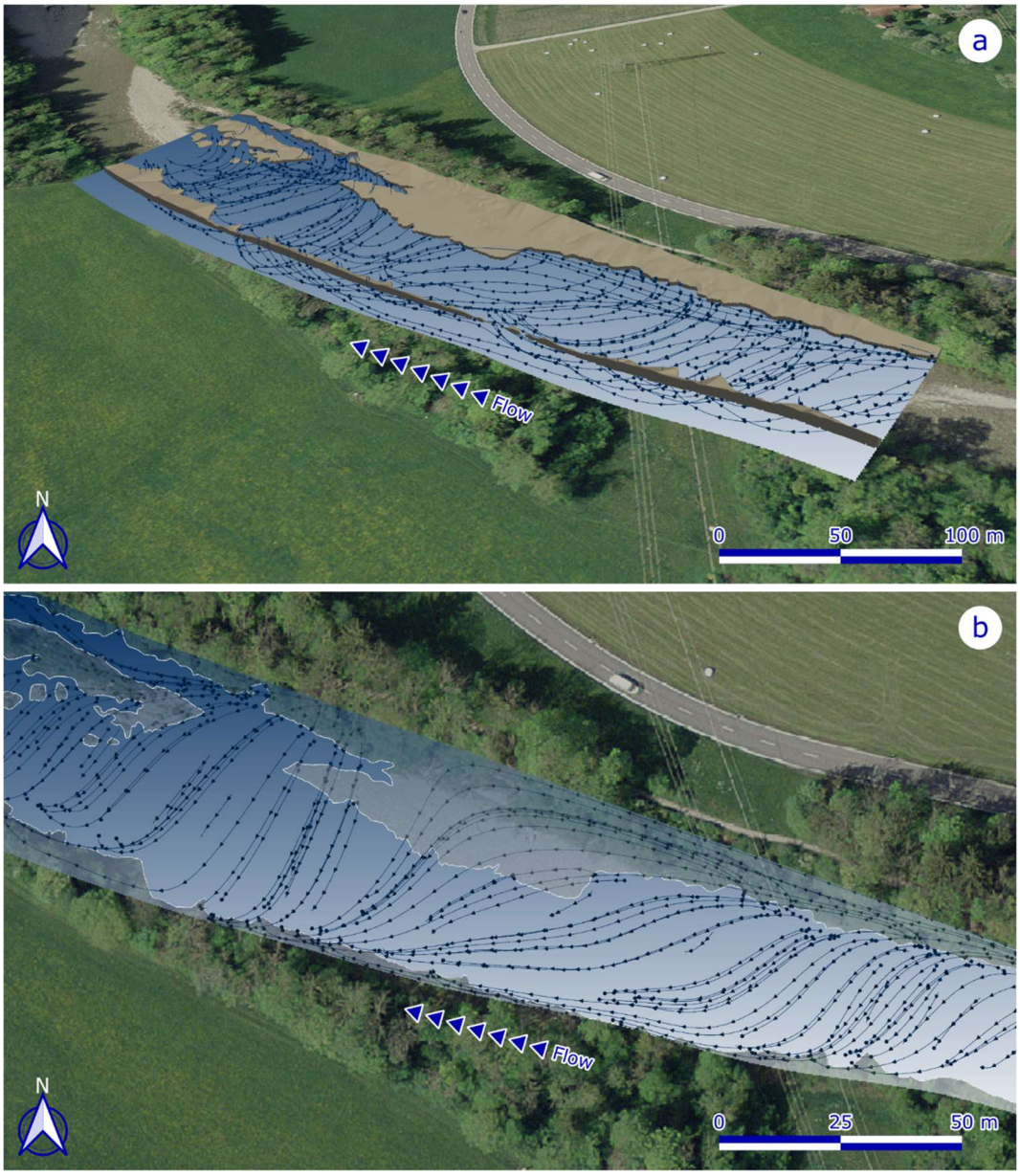



Figure 2.10. TC2 – Simulated groundwater flowpaths underneath the riverbed: (a) 3D view, (b) 2D top-view.

As required, the water balance is conserved within the system in both test cases. This fact is verified by the surface and subsurface effluxes (Figure 2.11a), that equalled the water influxes once the steady-state was reached. Such conservation of mass confirms the good functioning of the coupling iterative scheme of darcyInterTransportFoam. On the other hand, the exchange fluxes across the riverbed show higher upwelling rates than downwelling (Figure 2.11b), indicating that the exfiltration towards the river prevails over the infiltration into the ground. This pattern is confirmed by the conservative solute and temperature results presented in the following sections.

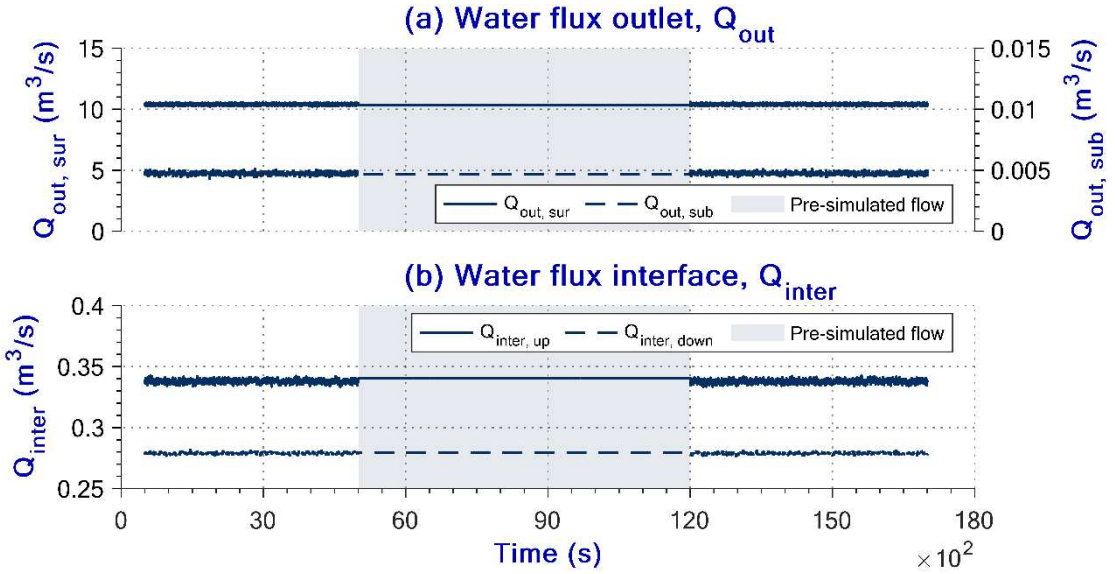



Figure 2.11. TC1 - Water flux across the (a) outlet and (b) interface boundaries. The subscripts *sur* and *sub* refer to surface and subsurface domains, respectively, while *up* and *down* indicate upwelling and downwelling fluxes. The coloured area corresponds to the simulation period in which the pre-simulated flow field was used to only run solute transport and heat transfer.

### 2.4.2. Water Temperature Field

Both the SW-GW heat exchanges and the groundwater plume generated in the river during the pumping test were modelled in TC2 to evaluate the efficiency of the novel module.

The specified initial temperatures correspond to the mean values measured in-situ during the pumping experiment. Accordingly, the air and water phases in the surface domain were initially set to -2 and 2 °C, respectively, while the groundwater temperature was initialized to 8 °C.

The simulated plume was compared to that measured on site using a drone-mounted thermal camera. As shown in Figure 2.12, both drone-based (a) and model-based (b) outcomes depict the temperature contrast generated by the warm abstracted groundwater. However, while the trajectory and location of the plume look similar in both

images, the shape and length differ between them. The reasons behind these differences can be diverse and go from the lack of accurate field observations to a wrong selection of the simulation parameters, or even a combination of both. In this case, the high-flow conditions presented in the river the day the thermal flight was carried out made the acquisition of additional field data especially difficult. This fact reduced the availability of in-situ temperature measurements and, consequently, the capability to later validate the values observed with the drone.

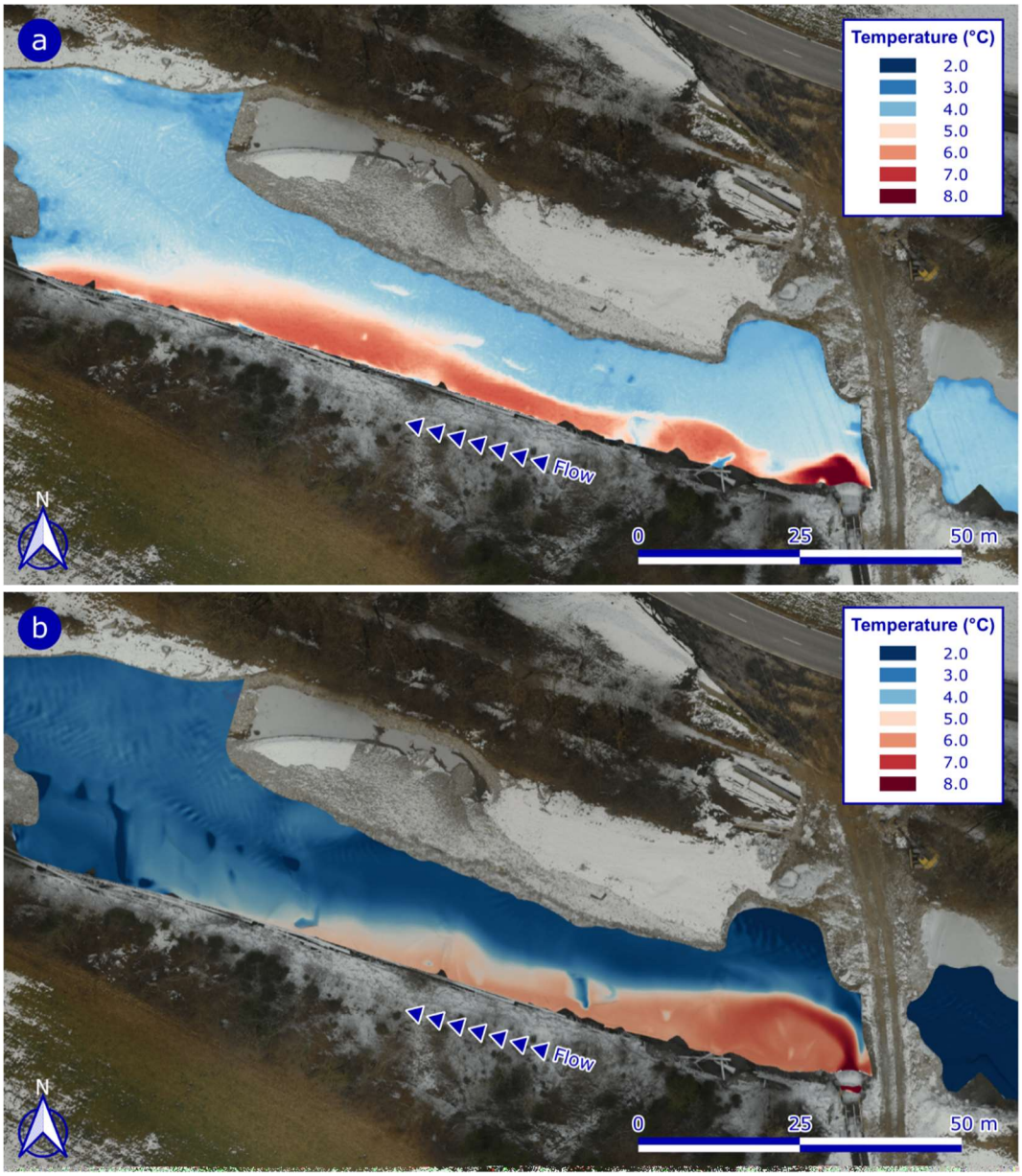



Figure 2.12. TC2 – (a) Drone-measured and (b) simulated temperature plumes generated by the groundwater jet at the water surface.

Moreover, access to detailed information on the discharged groundwater was challenging due to the engineering approach of the data source. In most cases, the

companies responsible for the pumping test provided data at hourly or even daily intervals. These temporal resolutions—far from those required to thoroughly study highly dynamic processes like temperature—combined with the described on-site data limitations, complicated the definition of a suitable model setup for reproducing the temperature field captured by the drone. However, despite these constraints, the overall pattern of heat redistribution is closely matched, indicating an adequate functioning of the new model's heat transfer module. The mismatch with the observed values can be attributed to a suboptimal selection of the simulation parameters, suggesting that an improved choice would lead to a better match.

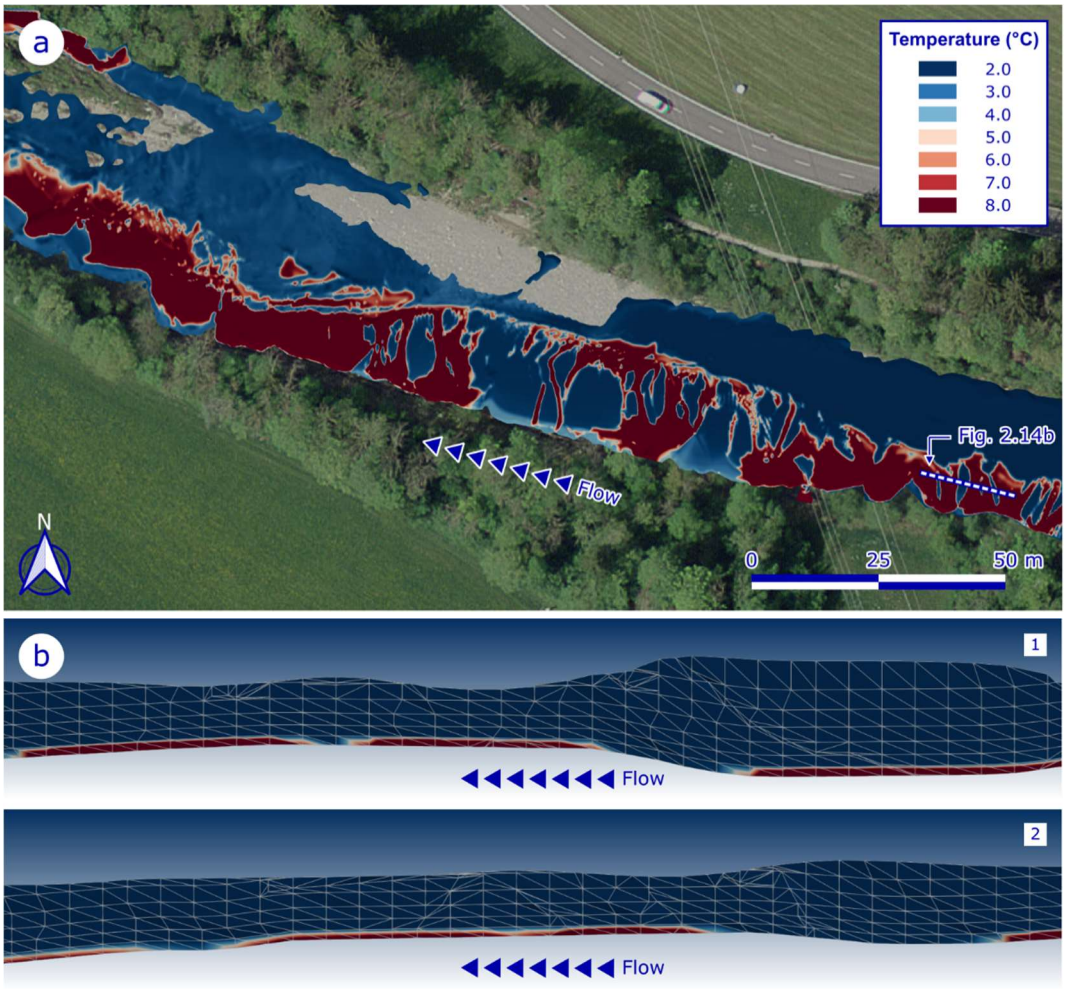



Figure 2.13. TC2 – (a) Modelled water temperature pattern at the riverbed and (b) cross-sectional view in the x-z plane along the dashed line in Figure 2.13a.

As for the heat exchanges across the riverbed, the results obtained show that the two-way exchange mostly occurs across the main course of the river, namely, the shallow channel located on the left side of the bed (Figure 2.13a). Within this region, areas of warm groundwater are combined with areas of cold surface water. No groundwater contribution is produced outside of this zone. The resulting temperature pattern aligns

with that described in previous studies at the site. One of these, Käser & Hunkeler (2016), identified an alternate sequence of upwelling and downwelling locations along the Emme which proved the existence of a complex pattern of river-aquifer interactions. The outcomes achieved with the new model reproduce such pattern, corroborating in this way the capability of `darcyInterTransportFoam` to solve actual heat transfer processes.

Furthermore, the heat outputs also indicate that, despite the multiple interactions produced across the riverbed, their impact in the surface and subsurface water temperatures is relatively small, as shown in **Figure 2.14**. This figure illustrates how the average temperatures stay nearly constant over time at the outlet of both domains. Such steady behaviour can be explained by the small magnitude of the riverbed exchanges (**Figure 2.11b**). As a result of this, the tiny gaining fluxes infiltrate in the ending domain just a few millimetres (first row of cells) before mixing with the main flow of the system (**Figure 2.13b**). The big difference in volume between the two flows makes the temperature of the former to diffuse within the second, consequently minimizing its effect in the mean temperature of the domain. These little heat infiltrations would also explain the measuring approach followed by Käser & Hunkeler (2016), that required from near-bed measurements to properly detect the temperature changes resulting from the subsurface upwelling.

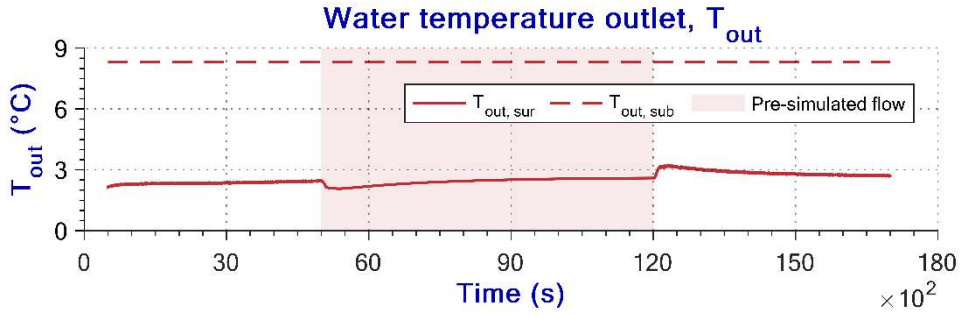



**Figure 2.14. Average water temperature across the surface and subsurface outlet boundaries. The subscripts *sur* and *sub* refer to surface and subsurface domains, respectively. The coloured area corresponds to the simulation period in which the pre-simulated flow field was used to only run heat transfer.**

### 2.4.3. Conservative Solute Field

The solute analysis skipped the plume caused by the released groundwater and only simulated the mass exchanges that occurred across the river-aquifer interface in TC2.

Initial $C_{sur,i}$ of 0 and 5 kg/m$^3$ were respectively prescribed for the air and water phases, while $C_{sub,i}$ was initialized to 10 kg/m$^3$. Although unrealistic, these high concentration values were chosen to ensure a clear representation of both the interfacial exchanges and the solute mixture in each domain.

The solute concentrations attained at the riverbed are shown in Figure 2.15. Analogously to temperature, the bidirectional exchange pattern is mainly located along the shallow channel situated on the left side of the bed. Outside of this zone, no groundwater exfiltration is produced and, thus, the concentrations remain around 5 kg/m$^3$. However, in contrast to heat, that diffuses within the surface flow, the solute mass concentrates over time towards the end of the simulation domain. Such accumulation is behind the different extension of subsurface upwelling depicted in Figure 2.13a and Figure 2.15. Yet, the locations where the groundwater actually exfiltrates can be still identified, as they correspond to the darkest spots of the resultant solute field. The pattern described by such spots matches that exhibited by the warm upwelling groundwater, as expected (Figure 2.13a).

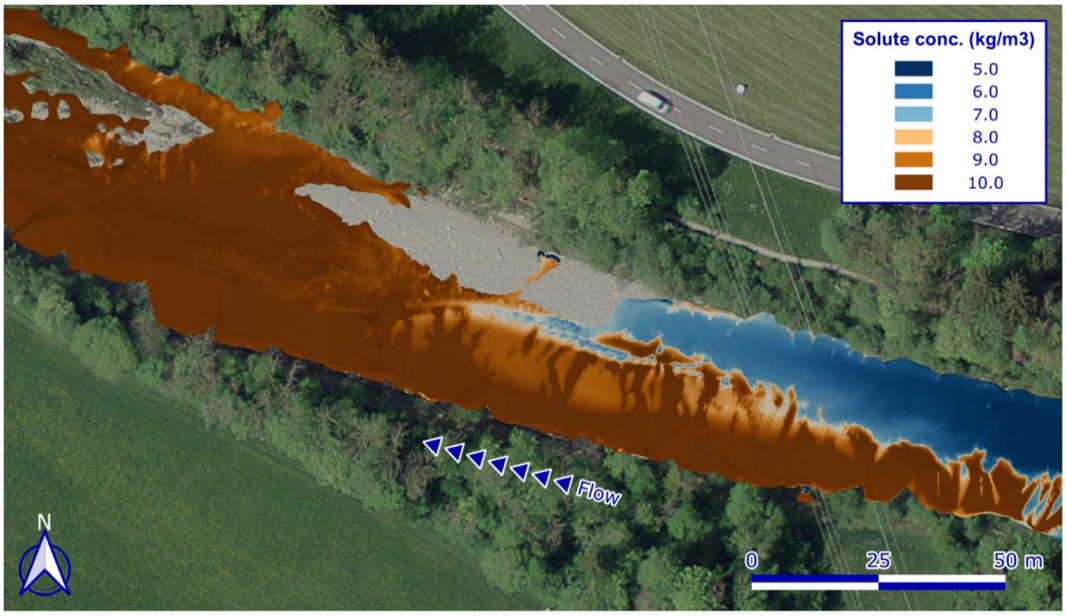



Figure 2.15. TC2 - Modelled solute concentration pattern at the riverbed.

The aforementioned downstream concentrations are supported by the solute efflux of the surface domain. As plotted in Figure 2.16a, the concentration increases over time at the surface outlet while remains essentially stable at its subsurface counterpart. The followed trends confirm the solute accumulation at the surface and also point out the prevalence of subsurface upwelling over surface downwelling. The latter is corroborated by the solute fluxes measured across the riverbed (Figure 2.16b), which aligned with the previously described water exchanges (Figure 2.11b).

The solute mass increases in the surface domain and decreases in the subsurface as a result of the described interfacial exchanges (Figure 2.16c). The steady-state is reached in both domains once the solute is fully mixed within the system. However, as expected, the total mass is conserved and stays constant throughout the whole simulation. This fact corroborates the system mass balance and, thus, the good functioning of the coupling iterative algorithm at the interface of both domains.

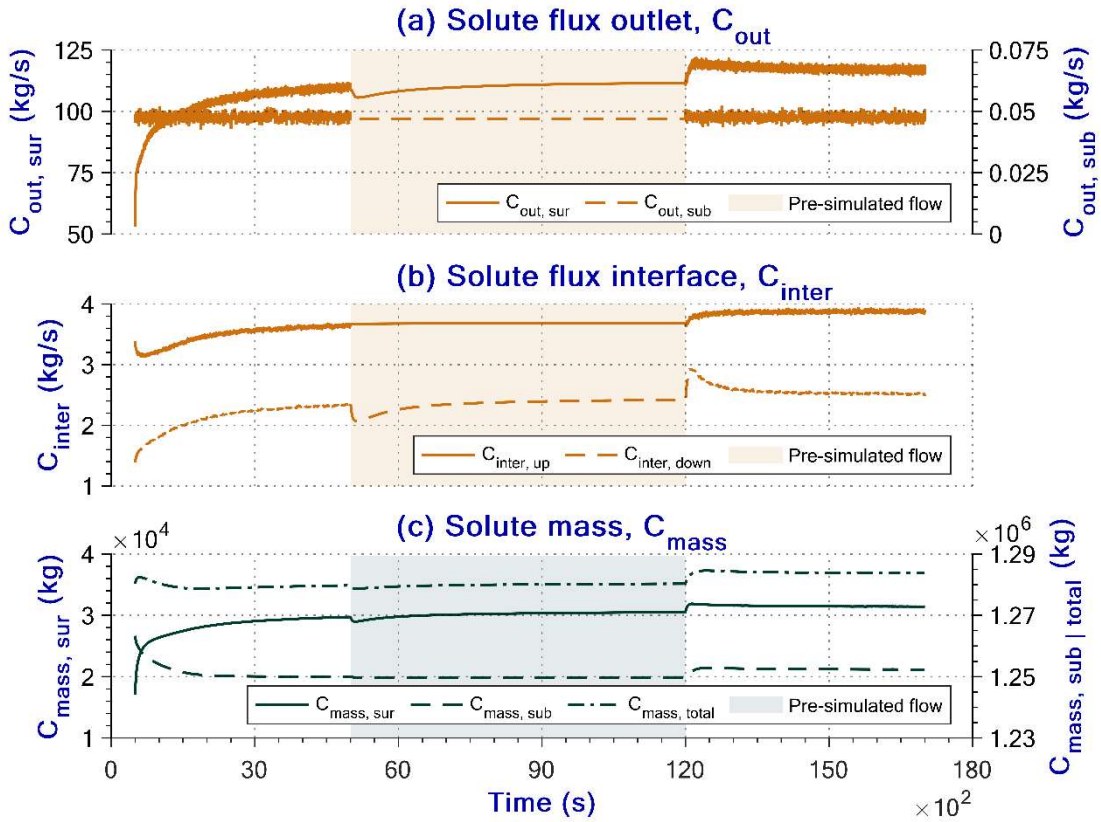



Figure 2.16. TC2 - Solute flux across the (a) outlet and (b) interface boundaries. (c) Solute mass in each simulated domain and total mass in the system. The subscripts *sur* and *sub* refer to surface and subsurface domains, respectively, while *up* and *down* indicate upwelling and downwelling fluxes. The coloured area corresponds to the simulation period in which the pre-simulated flow field was used to only run solute transport.

## 2.5. Limitations and Implications

Analogously to its predecessor, the current version of `darcyInterTransportFoam` simulates steady-state, fully saturated subsurface flow assuming a static groundwater table. This modelling framework reflects the physical assumption that surface water dynamics occur on timescales much shorter than those governing subsurface flow, thereby justifying the steady-state treatment of groundwater. However, the inability to simulate unsaturated conditions introduces limitations regarding the applicability of the solver. For example, if the groundwater table drops under a clogged streambed, unsaturated zones develop. The model cannot reproduce this disconnection between the surface and the subsurface. Similarly, rapid declines in surface water levels can lead to unsaturated conditions along the riverbanks, which also cannot be represented within the current framework. These limitations are not unique to `darcyInterTransportFoam`, but were already inherent to `hyporheicScalarInterFoam`. Nonetheless, scenarios

involving localized and well-defined unsaturated regions, such as fine-grained riverbank deposits or seasonally disconnected floodplain margins, can still be approximated with the new code by prescribing impermeable zones and conducting fully-coupled simulations only in the hydraulically connected sections.

Meanwhile, the assumption of full saturation remains valid in other hydrological settings, including gaining river reaches supported by shallow, unconfined aquifers; alluvial valleys with consistently high groundwater tables due to regional discharge; and perennial rivers where sustained baseflow maintains saturation across the riverbed and adjacent banks. However, while the current modelling approach may be appropriate for the aforementioned cases, further development of the solver will be required to account for the groundwater table fluctuations and the effects of unsaturated flow on the overall system dynamics. One possible path forward would be to adopt a strategy similar to that implemented in **groundwaterFoam** (Horgue, et al. 2022), which solves unsaturated flow by linearizing the pressure head form of Richards' equation using either Picard's or Newton's method. Additional work will also be needed to develop a reactive solute transport module for the simulation of key biogeochemical reactions (e.g., aerobic respiration, denitrification, etc.). Although previous custom fully-coupled solvers in OpenFOAM, such as **hyporheicFoam**, were already capable of simulating reactive processes, integrating them into the current framework will require reformulating the module to operate within a two-phase flow context, which adds complexity to its implementation. These enhancements will enable more comprehensive simulations of the dynamic and biogeochemical processes occurring at the river-aquifer interface, even in regions where SW and GW are hydraulically disconnected.

On the other hand, the accessibility to sufficient computational resources to run big CFD models, such as those described in this paper, may represent a limitation to achieving an efficient computation with **darcyInterTransportFoam**. In this way, the OpenFOAM software allows users to take full advantage of the computer hardware by enabling parallel computations on a theoretically unlimited number of processors. Four mesh decomposition methods are supported to that end. In the present study, the manual decomposition method was applied to both cases to ensure an equal division of the surface and subsurface domains. Moreover, 100 and 124 CPUs were used, respectively, to run in parallel TC1 and TC2 on 4 nodes of a local HPC (High-Performance Computing) cluster (2.9 GHz, Ubuntu 22.04.1 LTS operating system). However, access to such amount of computing resources can sometimes be challenging, since most computers are normally equipped with just 8 to 16 cores (with hyperthreading), rarely more. For this reason, it is highly encouraged to use **darcyInterTransportFoam** on massively parallel machines (e.g. HPC supercomputers or multi-core processors) provided with the necessary computational resources to run any model regardless of its scale and resolution. However, there are some caviates concerning parallelisation. A common recommendation for domain decomposition is that the number of cells hosted per core

should lie between 30000 and 100000 to guarantee an optimal parallelisation. Whenever this condition is not met and, therefore, the number of cells per core exceeds the recommended cap, the computational times increase dramatically, consequently breaking the compromise between computational efficiency and accuracy in the numerical solution.

Regarding the limitations specific to the simulated test cases, the difficulty of acquiring high-resolution in-situ data on the simulated flow, transport and heat processes reduced the availability of field observations for comparison against the models' results. The small number of on-site observations (drone-based temperature maps, point temperature measurements and flow data) could initially be considered insufficient to guarantee the quality and accuracy of the simulations. However, the simulated results are consistent with findings from previously conducted field and modelling studies at the site. This agreement, together with the conservation of flow and mass observed in the simulations, provides additional evidence supporting the computational consistency and efficiency of the new solver.

## 2.6. Summary and Conclusions

An upgraded version of the fully-coupled, three-dimensional SW-GW solver `hyporheicScalarInterFoam`, named `darcyInterTransportFoam`, is presented in this chapter. Its novel simulation features are thoroughly described, along with an evaluation of its performance in two test cases conducted at 235 m- and 106 m-long river segments of an actual alluvial aquifer.

The results achieved in both test cases confirm the good functioning of the new model and its add-ons. Such a statement is supported by the facts presented below. The number of the involved solver modification (Section 2.2.3.1) is included between square brackets in each point:

- The water flow exits the surface domain freely at a subcritical water height without requiring an obstacle (e.g., a step, ramp or gate) to stabilize the water level before reaching the outlet patch (Bayon-Barrachina & Lopez-Jimenez, 2015) (2).
- The parametrization of the real aquifer is captured in the simulated subsurface domain (including anthropogenic structures), enabling a realistic estimation of groundwater flowpaths, subsurface velocities and hydraulic heads (1, 3).
- Unsaturated regions in the subsurface domain, formed beneath surface dry areas, are no longer a limitation for the computation, as they are defined as impermeable. The simulation is still carried out at the hydraulically connected river-aquifer zones. Accordingly, both direction and length of the flowpaths adjust to the physical and geological characteristics of the system (1, 3, 4).

- Solute transport and heat transfer are effectively simulated in the surface and subsurface domains. Additionally, both processes are solved in a single phase (water) in the surface domain. The agreement between the modelled outcomes and field observations further validates this model's performance (5, 6, 7, 9).
- Both water and solute mass balances are conserved within the system, which ensures the flux consistency across the interface of both domains (1, 6, 8, 9).
- Water, solute and temperature results exhibit interconnected behaviour. Similarly, the flux exchanges across the interface and the mass balance of each domain display interdependent dynamics (1, 5, 6, 7, 8, 9).

The above points illustrate that, despite the modelling limitations of the present version of **darcyInterTransportFoam**, the new solver fills most of the simulation gaps of its predecessor, **hyporheicScalarInterFoam**. In this way, the model is successfully applied to solve the flow, solute and heat exchanges across the riverbed of the selected Emme River segments. Moreover, its utility to study the SW-GW interactions and dynamics beyond the hyporheic zone is also demonstrated through its application to two topographically different, reach-scale scenarios.

The successful performance of **darcyInterTransportFoam** confirms its potential for simulating multiple hydrogeological scenarios, underpinned by its capability to represent the intricate characteristics of natural systems and the diverse dynamic processes they drive. Among its new functionalities, the ability to prescribe heterogeneity and anisotropy in the subsurface, as well as to define layers or anthropogenic structures (e.g., weirs), represents a significant step toward more physically realistic riverbed-aquifer interactions. This feature, combined with the solute transport and heat transfer modelling, enables the simulation of important feedback mechanisms between SW and GW, whose detailed dynamics could not be accounted for in the previous version of the solver. For instance, the migration of a contaminant plume from GW into a river under geologically complex conditions can now be simulated, with spatial variability in $K_{ij}$ and $\theta_i$ governing both the transport pathways and the timing of pollutant discharge. Similarly, the inclusion of stratified layering and anisotropic properties allows for a more accurate representation of gaining-losing conditions and hyporheic exchanges, both of which are highly sensitive to subsurface characterization and drive key biogeochemical processes. In such contexts, temperature dynamics act not only as tracers but also as active drivers of biochemical reactions. Moreover, in river-aquifer systems affected by anthropogenic modifications, such as dams, levees or bank reinforcements, the improved modelling framework enables the explicit definition of these structures to assess their influence on hydraulic connectivity, and consequently on processes like solute retention and thermal buffering.

Further code development is still needed to address the remaining limitations of the current model version, such as the simulation of unsaturated flow and multicomponent

reactive transport. Closing these gaps will enable a more comprehensive representation of the physical and biogeochemical processes occurring at the river-aquifer interface.

# References


[1] Aryal, P. (2019). *Modeling free surface thermal flow with relative motion of heat source and drop injector with respect to a liquid pool.* Technical Report in Proceedings of the course CFD with OpenSource Software, Chalmers University of Technology, Göteborg, Sweden. https://doi.org/10.17196/OS_CFD#YEAR_2019

[2] Barthel, R., & Banzhaf, S. (2016). Groundwater and surface water interaction at the regional-scale – A review with focus on regional integrated models. *Water Resources Management, 30*(1), 1-32. https://doi.org/10.1007/s11269-015-1163-z

[3] Bayon-Barrachina, A., & Lopez-Jimenez, P. A. (2015). Numerical analysis of hydraulic jumps using OpenFOAM. *Journal of Hydroinformatics, 17*(4), 662-678. https://doi.org/10.2166/hydro.2015.041

[4] Bear, J. (1961). On the tensor form of dispersion in porous media. *Journal of Geophysical Research, 66*(4), 1185-1197. https://doi.org/10.1029/JZ066i004p01185

[5] Bear, J. (Ed.). (1972). *Dynamics of Fluids in Porous Media* (1st ed.). New York, USA: American Elsevier Publishing Company. ISBN-13: 978-0444001146

[6] Berberović, E., van Hinsberg, N. P., Jakirlić, S., Roisman, I. V., & Tropea, C. (2009). Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution. *Physical Review E, 79*(3), 036306. https://doi.org/10.1103/PhysRevE.79.036306

[7] Blanc, T., Peel, M., Brennwald, M. S., Kipfer, R., & Brunner, P. (2024). Efficient injection of gas tracers into rivers: A tool to study Surface water-Groundwater interactions. *Water Research, 254*, 121375. https://doi.org/10.1016/j.watres.2024.121375

[8] Blau, R. V., & Muchenberger, F. (1997). *Grundlagen für Schutz und Bewirtschaftung der Grundwasser des Kantons Bern: Nutzungs-, Schutz- und Überwachungskonzept für die Grundwasserleiter des obersten Emmentals, zwischen Emmenmatt, Langnau und Eggiwil.* Technical Report, Wasser- und Energiewirtschaft des Kantons Bern, Bern, Switzerland.

[9] Boano, F., Harvey, J. W., Marion, A., Packman, A. I., Revelli, R., Ridolfi, L., & Wörman, A. (2014). Hyporheic flow and transport processes: Mechanisms, models, and biogeochemical implications. *Reviews of Geophysics, 41*, 603-679. https://doi.org/10.1002/2012RG000417

[10] Boudreau, B. P. (1996). The diffusive tortuosity of fine-grained unlithified sediments. *Geochimica et Cosmochimica Acta, 60*(16), 3139-3142. https://doi.org/10.1016/0016-7037(96)00158-5

[11] Broecker, T., Elsesser, W., Teuber, K., Özgen, I., Nützmann, G., & Hinkelmann, R. (2017). High-resolution simulation of free-surface flow and tracer retention over streambeds with ripples. *Limnologica, 68*, 46-58. https://doi.org/10.1016/j.limno.2017.06.005

[12] Broecker, T., Gollo, V. S., Fox, A., Lewandowski, J., Nützmann, G., Arnon, S., & Hinkelmann, R. (2021). High-Resolution Integrated Transport Model for Studying Surface Water-Groundwater Interaction. *Groundwater, 59*(4), 488-502. https://doi.org/10.1111/gwat.13071

[13] Broecker, T., Teuber, K., Sobhi Gollo, V., Nützmann, G., Lewandowski, J., & Hinkelmann, R. (2019). Integral Flow Modelling Approach for Surface Water-Groundwater Interactions along a Rippled Streambed. *Water, 11*(7), 1517. https://doi.org/10.3390/w11071517

[14] Brunner, P., Therrien, R., Renard, P., Simmons, C. T., & Hendricks Franssen, H.-J. (2017). Advances in understanding river-groundwater interactions. *Reviews of Geophysics, 55*(3), 818-854. https://doi.org/10.1002/2017RG000556

[15] Cardenas, M. B., Cook, P. L., Jiang, H., & Traykovski, P. (2008). Constraining denitrification in permeable wave-influenced marine sediment using linked hydrodynamic and biogeochemical modeling. *Earth and Planetary Science Letters, 275*(1-2), 127-137. https://doi.org/10.1016/j.epsl.2008.08.016

[16] Chen, X., M. Bayani, C., & Chen, L. (2015). Three-dimensional versus two-dimensional bed form-induced hyporheic exchange. *Water Resources Research, 51*(4), 2923-2936. https://doi.org/10.1002/2014WR016848

[17] Chen, X., M. Bayani, C., & Chen, L. (2018). Hyporheic Exchange Driven by Three-Dimensional Sandy Bed Forms: Sensitivity to and Prediction from Bed Form Geometry. *Water Resources Research, 54*(6), 4131-4149. https://doi.org/10.1029/2018WR022663

[18] Coluccio, K., & Morgan, L. K. (2019). A review of methods for measuring groundwater-surface water exchange in braided rivers. *Hydrology and Earth System Sciences, 23*(10), 4397-4417. https://doi.org/10.5194/hess-23-4397-2019

[19] Deshpande, S. S., Anumolu, L., & Trujillo, M. F. (2012). Evaluating the performance of the two-phase flow solver interFoam. *Computational Science & Discovery, 5*(1), 014016. https://doi.org/10.1088/1749-4699/5/1/014016

[20] Dichgans, F., Boos, J.-P., Ahmadi, P., Frei, S., & Fleckenstein, J. H. (2023). Integrated numerical modeling to quantify transport and fate of microplastics in the

hyporheic zone. *Water Research, 243*, 120349. https://doi.org/10.1016/j.watres.2023.120349

[21] Figura, S., Livingstone, D. M., & Kipfer, R. (2013). Competing controls on groundwater oxygen concentrations revealed in multidecadal time series from riverbank filtration sites. *Water Resources Research, 49*(11), 7411-7426. https://doi.org/10.1002/2013WR013750

[22] Fleckenstein, J. H., Krause, S., Hannah, D. M., & Boano, F. (2010). Groundwater-surface water interactions: New methods and models to improve understanding of processes and dynamics. *Advances in Water Resources, 33*, 1291-1295. https://doi.org/10.1016/j.advwatres.2010.09.011

[23] Haitjema, H. M., & Anderson, M. P. (2016). Darcy Velocity Is Not a Velocity. *Groundwater, 54*(1), 1-1. https://doi.org/10.1111/gwat.12386

[24] Haroun, Y., Legendre, D., & Raynal, L. (2010). Volume of fluid method for interfacial reactive mass transfer: Application to stable liquid film. *Chemical Engineering Science, 65*(10), 2896-2909. https://doi.org/10.1016/j.ces.2010.01.012

[25] Hermans, T., Goderniaux, P., Jougnot, D., Fleckenstein, J. H., Brunner, P., Nguyen, F., Linde, N., Huisman, J. A., Bour, O., Lopez Alvis, J., Hoffmann, R., Palacios, A., Cooke, A.-K., Pardo-Álvarez, Á., Blazevic, L., Pouladi, B., Haruzi, P., Fernandez Visentini, A., Nogueira, G. E. H., Tirado-Conde, J., Looms, M. C., Kenshilikova, M., Davy, P., & Le Borgne, T. (2023). Advancing measurements and representations of subsurface heterogeneity and dynamic processes: towards 4D hydrogeology. *Hydrology and Earth System Sciences, 27*(1), 255-287. https://doi.org/10.5194/hess-27-255-2023

[26] Hirt, C. W., & Nichols, B. D. (1981). Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics, 39*(1), 201-225. https://doi.org/10.1016/0021-9991(81)90145-5

[27] Horgue, P., Soulaine, C., Franc, J., Guibert, R., & Debenest, G. (2015). An open-source toolbox for multiphase flow in porous media. *Computer Physics Communications, 187*, 217-226. https://doi.org/10.1016/j.cpc.2014.10.005

[28] Horgue, P., Renard, F., Gerlero, G., Guibert, R., & Debenest, G. (2022). porousMultiphaseFoam v2107: An open-source tool for modeling saturated/unsaturated water flows and solute transfers at watershed scale. *Computer Physics Communications, 273*, 108278. https://doi.org/10.1016/j.cpc.2021.108278

[29] Issa, R. I. (1986). Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics, 62*(1), 40-65. https://doi.org/10.1016/0021-9991(86)90099-9

[30] Janssen, F., Cardenas, M. B., Sawyer, A. H., Dammrich, T., Krietsch, J., & de Beer, D. (2012). A comparative experimental and multiphysics computational fluid dynamics study of coupled surface–subsurface flow in bed forms. *Water Resources Research, 48*(8), W08514. https://doi.org/10.1029/2012WR011982

[31] Jin, G., Tang, H., Li, L., & Barry, D. A. (2011). Hyporheic flow under periodic bed forms influenced by low-density gradients. *Geophysical Research Letters, 38*(22), L22401. https://doi.org/10.1029/2011GL049694

[32] Käser, D., & Hunkeler, D. (2016). Contribution of alluvial groundwater to the outflow of mountainous catchments. *Water Resources Research, 52*, 680-697. https://doi.org/10.1002/2014WR016730

[33] Krause, S., Boano, F., Cuthbert, M. O., Fleckenstein, J. H., & Lewandowski, J. (2014). Understanding process dynamics at aquifer-surface water interfaces: An introduction to the special section on new modeling approaches and novel experimental technologies. *Water Resources Research, 50*(2), 1847-1855. https://doi.org/10.1002/2013WR014755

[34] Krause, S., Hannah, D., & Grimm, N. (Eds.). (2023). *Ecohydrological Interfaces* (1st ed.). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119489702

[35] Krause, S., Lewandowski, J., Grimm, N. B., Hannah, D. M., Pinay, G., McDonald, K., Martí, E., Argerich, A., Pfister, L., Klaus, J., Battin, T., Larned, S. T., Schelker, J., Fleckenstein, J., Schmidt, C., Rivett, M. O., Watts, G., Sabater, F., Sorolla, A., & Turk, V. (2017). Ecohydrological interfaces as hot spots of ecosystem processes. *Water Resources Research, 53*(8), 6359-6376. https://doi.org/10.1002/2016WR019516

[36] Lee, A., Aubeneau, A., & Cardenas, M. (2020). The Sensitivity of Hyporheic Exchange to Fractal Properties of Riverbeds. *Water Resources Research, 56*(5), e2019WR026560. https://doi.org/10.1029/2019WR026560

[37] Lee, A., Aubeneau, A., Cardenas, M., & Liu, X. (2022). Hyporheic Exchange Due to Cobbles on Sandy Beds. *Water Resources Research, 58*(1), e2021WR030164. https://doi.org/10.1029/2021WR030164

[38] Lee, A., Aubeneau, A., Liu, X., & Cardenas, M. B. (2021). Hyporheic Exchange in Sand Dunes Under a Freely Deforming River Water Surface. *Water Resources Research, 57*(3), e2020WR028817. https://doi.org/10.1029/2020WR028817

[39] Lewandowski, J., Arnon, S., Banks, E., Batelaan, O., Betterle, A., Broecker, T., Coll, C., Drummond, J. D., Gaona Garcia, J., Galloway, J., Gomez-Velez, J., Grabowski, R. C., Herzog, S. P., Hinkelmann, R., Höhne, A., Hollender, J., Horn, M. A., Jaeger, A., Krause, S., Löchner Prats, A., Magliozzi, C., Meinikmann, K., Mojarrad, B. B., Mueller, B. M., Peralta-Maraver, I., Popp, A. L., Posselt, M., Putschew, A., Radke, M., Raza, M., Riml, J., Robertson, A., Rutere, C., Schaper, J. L., Schirmer, M.,

Schulz, H., Shanafield, M., Singh, T., Ward, A. S., Wolke, P., Wörman, A., & Wu, L. (2019). Is the Hyporheic Zone Relevant beyond the Scientific Community? *Water, 11*(11), 2230. https://doi.org/10.3390/w11112230

[40] Lewandowski, J., Meinikmann, K., & Krause, S. (2020). Groundwater-Surface Water Interactions: Recent Advances and Interdisciplinary Challenges. *Water, 12*(1), 296. https://doi.org/10.3390/w12010296

[41] Li, B., Liu, X., Kaufman, M. H., Turetcaia, A., Chen, X., & Cardenas, B. M. (2020). Flexible and Modular Simultaneous Modeling of Flow and Reactive Transport in Rivers and Hyporheic Zones. *Water Resources Research, 56*(2), e2019WR026528. https://doi.org/10.1029/2019WR026528

[42] Marquez-Damian, S. (2012). *Description and utilization of interFoam multiphase solver.* Buenos Aires: CIMEC-INTEC. Retrieved from https://www.cfdyna.com/Home/OpenFOAM/of_Tut_Web/of_Suites_Description.pdf

[43] Maxwell, R. M., Putti, M., Meyerhoff, S., Delfs, J.-O., Ferguson, I. M., Ivanov, V., Kim, J., Kolditz, O., Kollet, S. J., Kumar, M., Lopez, S., Niu, J., Paniconi, C., Park, Y.-J., Phanikumar, M. S., Shen, C., Sudicky, E. A., & Sulis, M. (2014). Surface-subsurface model intercomparison: A first set of benchmark results to diagnose integrated hydrology and feedbacks. *Water Resources Research*, 50(2), 1531-1549. https://doi.org/10.1002/2013WR013725

[44] Menter, F. R. (1994). Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal, 32*(8), 1598-1605. https://doi.org/10.2514/3.12149

[45] Nieves-Remacha, M. J., Yang, L., & Jensen, K. F. (2015). OpenFOAM Computational Fluid Dynamic Simulations of Two-Phase Flow and Mass Transfer in an Advanced-Flow Reactor. *Industrial & Engineering Chemistry Research, 54*(26), 6649-6659. https://doi.org/10.1021/acs.iecr.5b00480

[46] Nisbeth, C. S., Kidmose, J., Weckström, K., Reitzel, K., Odgaard, B. V., Bennike, O., Thorling, L., McGowan, S., Schomacker, A., Kristensen, D. L. J., & Jessen, S. (2019). Dissolved inorganic geogenic phosphorus load to a groundwater-fed lake: Implications of terrestrial phosphorus cycling by groundwater. *Water, 11*(11), 2213. https://doi.org/10.3390/w11112213

[47] Nogueira, G. E., Schmidt, C., Brunner, P., Graeber, D., & Fleckenstein, J. H. (2021b). Transit-Time and Temperature Control the Spatial Patterns of Aerobic Respiration and Denitrification in the Riparian Zone. *Water Resources Research, 57*(12), e2021WR030117. https://doi.org/10.1029/2021WR030117

[48] OpenCFD Ltd. (2023, December). *OpenFOAM® v2312 User Guide*. Retrieved December 25, 2024, from https://doc.openfoam.com/2312/

[49] OpenCFD Ltd. (2024, June). OpenFOAM® v2406. OpenFOAM Download [code]. Retrieved from https://dl.openfoam.com/source/v2406/

[50] Oxtoby, O., Heyns, J., & Suliman, R. (2013). A finite-volume solver for two-fluid flow in heterogeneous porous media based on OpenFOAM. Hamburg: 7th Open Source CFD International Conference. https://doi.org/10.13140/2.1.3075.8400

[51] Pardo-Álvarez, Á. (2025). *darcyInterTransportFoam v1.0.* [Software], Zenodo. https://doi.org/10.5281/zenodo.15857142

[52] Patankar, S. V., & Spalding, D. B. (1972). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer, 15*(10), 1787-1806. https://doi.org/10.1016/0017-9310(72)90054-3

[53] Peel, M., Kipfer, R., Hunkeler, D., & Brunner, P. (2022). Variable 222Rn emanation rates in an alluvial aquifer: Limits on using 222Rn as a tracer of surface water – Groundwater interactions. *Chemical Geology, 599*, 120829. https://doi.org/10.1016/j.chemgeo.2022.120829

[54] Poffet, D. (2011). *Interactions nappe-rivière et stockage dans l'aquifère de la Haute-Emme: Approche par la modélisation numérique.* MSc Thesis, University of Neuchâtel, Centre for Hydrogeology and Geothermics, Neuchâtel, Switzerland.

[55] Popp, A. L., Pardo-Álvarez, Á., Schilling, O. S., Scheidegger, A., Musy, S., Peel, M., Brunner, P., Purtschert, R., Hunkeler, D., & Kipfer, R. (2021). A Framework for Untangling Transient Groundwater Mixing and Travel Times. *Water Resources Research, 57*(4), e2020WR028362. https://doi.org/10.1029/2020WR028362

[56] Rubin, H. (1974). Heat dispersion effect on thermal convection in a porous medium layer. *Journal of Hydrology, 21*(2), 173-185. https://doi.org/10.1016/0022-1694(74)90035-3

[57] Rusche, H. (2003). *Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions.* PhD Thesis, Imperial College of Science, Technology and Medicine, London, UK.

[58] Schaper, J. L., Posselt, M., McCallum, J. L., Banks, E. W., Hoehne, A., Meinikmann, K., Shanafield, M. A., Batelaan, O., & Lewandowski, J. (2018). Hyporheic Exchange Controls Fate of Trace Organic Compounds in an Urban Stream. *Environmental Science Technology, 52*(21), 12285–12294. https://doi.org/10.1021/acs.est.8b03117

[59] Scheidegger, A. E. (1961). General theory of dispersion in porous media. *Journal of Geophysical Research, 66*(10), 3273–3278. https://doi.org/10.1029/JZ066i010p03273

[60] Schilling, O. S., Gerber, C., Partington, D. J., Purtschert, R., Brennwald, M. S., Kipfer, R., Hunkeler, D., & Brunner, P. (2017a). Advancing Physically-Based Flow

Simulations of Alluvial Systems Through Atmospheric Noble Gases and the Novel 37Ar Tracer Method. *Water Resources Research, 53*(12), 10465-10490. https://doi.org/10.1002/2017WR020754

[61] Schilling, O. S., Partington, D. J., Doherty, J., Kipfer, R., Hunkeler, D., & Brunner, P. (2022). Buried Paleo-Channel Detection With a Groundwater Model, Tracer-Based Observations, and Spatially Varying, Preferred Anisotropy Pilot Point Calibration. *Geophysical Research Letters, 49*(14), e2022GL098944. https://doi.org/10.1029/2022GL098944

[62] Schulze, L., & Thorenz, C. (2014). The Multiphase Capabilities of the CFD Toolbox OpenFOAM for hydraulic engineering applications. En R. Lehfeldt, & R. Kopmann (Ed.). (pp. 1007-1016). Hamburg: Proceedings of the 11th International Conference on Hydroscience & Engineering. https://doi.org/20.500.11970/99530

[63] Schulze, L., & Thorenz, C. (2015). *Mehrphasenmodellierung im Wasserbau.* Karlsruhe: Bundesanstalt für Wasserbau (Hg.): Wasserbauwerke - Vom hydraulischen Entwurf bis zum Betrieb. https://doi.org/20.500.11970/102274

[64] Severin, T. (2017). *Computational fluid dynamics assisted design of thin-layer cascade photobioreactor components.* PhD Thesis, Technical University of Munich, Chair of Biochemical Engineering, Munich, Germany. Retrieved from https://mediatum.ub.tum.de/doc/1363208/1363208.pdf

[65] Sobhi Gollo, V., Broecker, T., Marx, C., Lewandowski, J., Nützmann, G., & Hinkelmann, R. (2022). A comparative study of integral and coupled approaches formodeling hydraulic exchange processes across a rippled streambed. *International Journal on Geomathematics, 13*(16), 1-27. https://doi.org/10.1007/s13137-022-00206-5

[66] Tang, Q., Schilling, O. S., Kurtz, W., Brunner, P., Vereecken, H., & Hendricks Franssen, H.-J. (2018). Simulating Flood-Induced Riverbed Transience Using Unmanned Aerial Vehicles, Physically Based Hydrological Modeling, and the Ensemble Kalman Filter. *Water Resources Research, 54*(11), 9342-9363. https://doi.org/10.1029/2018WR023067

[67] Teuber, K. (2020). *A three-dimensional two-phase model for flow, transport and mass transfer processes in sewers.* PhD Thesis, Technischen Universität Berlin, Water Resources Management and Modeling of Hydrosystems, Berlin, Germany. https://doi.org/10.14279/depositonce-9576

[68] The MathWorks Inc. (2024). *MATLAB R2024a*. Retrieved December 25, 2024, from https://www.mathworks.com/products/matlab.html

[69] Tonina, D., & Buffington, J. M. (2009). Hyporheic Exchange in Mountain Rivers I: Mechanics and Environmental Effects. *Geography Compass, 3*(3), 1063-1086. https://doi.org/10.1111/j.1749-8198.2009.00226.x

[70] Toran, L. (2019). Groundwater-Surface Water Interaction. *Encyclopedia of Water: Science, Technology, and Society*, 1-12. https://doi.org/10.1002/9781119300762.wsts0027

[71] Trauth, N., & Fleckenstein, J. H. (2017). Single discharge events increase reactive efficiency of the hyporheic zone. *Water Resources Research, 53*(1), 779-798. https://doi.org/10.1002/2016WR019488

[72] Trauth, N., Musolff, A., Knoller, K., Kaden, U. S., Keller, T., Werban, U., & Fleckenstein, J. H. (2018). River water infiltration enhances denitrification efficiency in riparian groundwater. *Water Research, 130*, 185-199. https://doi.org/10.1016/j.watres.2017.11.058

[73] Trauth, N., Schmidt, C., Vieweg, M., Maier, U., & Fleckenstein, J. H. (2014). Hyporheic transport and biogeochemical reactions in pool-riffle systems under varying ambient groundwater flow conditions. *Journal of Geophysical Research: Biogeosciences, 119*(5), 910-928. https://doi.org/10.1002/2013JG002586

[74] Trauth, N., Schmidt, C., Vieweg, M., Oswald, S. E., & Fleckenstein, J. H. (2015). Hydraulic controls of in-stream gravel bar hyporheic exchange and reactions. *Water Resources Research, 51*(4), 2243-2263. https://doi.org/10.1002/2014WR015857

[75] Versteeg, H. K., & Malalasekera, W. (2007). *An introduction to computational fluid dynamics: The finite volume method* (2nd ed.). Edinburgh, UK: Pearson. ISBN-13: 978-0131274983

[76] Voermans, J. J., Ghisalberti, M., & Ivey, G. N. (2017). The variation of flow and turbulence across the sediment-water interface. *Journal of Fluid Mechanics, 824*, 413-437. https://doi.org/10.1017/jfm.2017.345

[77] Weller, H. G. (2002). *Derivation, modelling and solution of the conditionally averaged two-phase flow equations.* Boston, USA: Nabla Ltd.

[78] Weller, H. G., Tabor, G., Jasak, H., & Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics, 12*(6), 620-631. https://doi.org/10.1063/1.168744

[79] Winter, T. C., Harvey, W. J., Franke, O. L., & Alley, M. W. (1998). *Ground water and surface water: A single resource.* North Dakota Water Science Center. Denver: U.S. Geological Survey. https://doi.org/10.3133/cir1139

[80] Würsten, M. (1991). *GWB–Hydrogeologische Untersuchungen Aeschau: Schlussbericht.* Technical Report, Geotechnisches Institut, Zürich, Switzerland.

[81] Yao, J., & Zhang, H. (2023). Comparing Darcy's Law and the Brinkman Equation for Numerical Simulations of Saltwater Intrusion. *Sustainability, 15*(18), 13287. https://doi.org/10.3390/su151813287

[82] Young, D. F., Munson, B. R., Okiishi, T. H., & Huebsch, W. W. (Eds.). (2010). *A brief introduction to fluid mechanics* (5th ed.). New York, USA: John Wiley & Sons. ISBN-13: 978-0470596791

# 3

# A Comparison Study of a Fully-Coupled vs. a Fully-Integrated OpenFOAM Solver for Simulating Surface Water - Groundwater Interactions

A manuscript based on this chapter is currently in preparation for publication.

# 3.1. Introduction

Understanding the interactions between surface water (SW) and groundwater (GW) is critical for achieving effective water resource management, environmental conservation and ecological sustainability (Boano, et al., 2014; Krause, et al., 2023; Lewandowski, et al., 2019). The evolution of numerical modelling tools has played a pivotal role in enhancing this understanding, providing a robust complement to traditional field observation techniques (Barthel & Banzhaf, 2016; Brunner, et al., 2017; Fleckenstein, et al., 2010). Compared to the latter, numerical models can generate detailed datasets over extensive areas and extended periods, facilitating the comprehensive investigation of hydrological processes under various scenarios (Broecker, et al., 2021). Early hydrological models simulated SW and GW systems separately, often using simplified representations of the other system or ignoring it altogether. These models were frequently constrained by computational limitations and a lack of detailed data, leading to significant uncertainties in water balance estimates and interaction dynamics (Paniconi & Putti, 2015; Schilling, et al., 2019). Advances in computational power and hydrological science have led to the development of more sophisticated modelling techniques capable of accounting for the continuous interactions between SW and GW (Brunner, et al., 2017; Lewandowski, et al., 2020). The asynchronous linking, or one-way, coupling approach marked a significant advancement in this regard, allowing outputs from one model to serve as inputs for the other (Maxwell, et al., 2014). However, while extensively applied in SW-GW interaction cases (e.g., Chen, et al., 2015; Chen, et al., 2018; Janssen, et al., 2012; Lee, et al., 2020; Trauth, et al., 2015; Trauth & Fleckenstein, 2017), this technique is limited by its capacity to account for the interfacial feedback in only one direction, which can lead to significant errors in the water balance and solute mass conservation (Li, et al., 2020). This limitation is overcome by further advanced approaches, namely, fully-coupled models (FCMs; e.g., `ParFlow` (Kuffour, et al., 2020), `HydroGeoSphere` (Brunner & Simmons, 2012; Therrien, et al., 2010), `MODFLOW` (Brunner, et al., 2010; Langevin, et al., 2023)) and fully-integrated models (FIMs; e.g., `porousInter` (Oxtoby, et al., 2013)). These two modelling techniques allow the simulation of the actual bidirectional interactions between SW and GW, capturing the full range of hydrodynamics and chemical exchanges across their interface. FCMs involve the coupling of distinct equations for surface and subsurface flow through a sequential iterative, or two-way, scheme (Maxwell, et al., 2014), characterized by successive exchanges of boundary conditions (BCs) across the SW-GW interface. In contrast, FIMs solve a unified set of equations that describe both surface and subsurface flow within a single computational framework. This strategy is commonly referred to as global implicit coupling (Maxwell, et al., 2014), offering a more seamless representation of hydrological processes. Both modelling techniques represent a significant leap forward from traditional approaches, providing a more realistic representation of the natural functioning of a hydrosystem.

According to the paradigm shift towards studying SW-GW as a single resource, fully-integrated modelling should always be the preferred choice. Nevertheless, the use of this approach has been largely limited to local-scale investigations due to its high computational burden (e.g., Broecker, et al., 2019; Dichgans, et al., 2023). On the other hand, FCMs can be particularly beneficial when the distinct processes of each domain must be precisely modelled. In such cases, these techniques may not only enhance the accuracy of the simulations but also ensure the efficient use of computational resources by applying different control parameters and modelling strategies to each domain (Kollet & Maxwell, 2008). Selecting the most suitable technique to determine the continuous SW-GW interaction can therefore be challenging and, at times, confusing. Despite the extensive application of both modelling approaches to SW-GW studies in recent years (e.g., Broecker, et al., 2021; Chow, et al., 2019; Chow, et al., 2020; Dichgans, et al., 2023; Lee, et al., 2021; Lee, et al., 2022; Li, et al., 2020), very few systematic comparisons have been conducted to test whether these different approaches can achieve the same level of process understanding (e.g., Sobhi Gollo, et al., 2022). Consequently, there is a need for comprehensive comparison studies of hydrologic models and the development of a robust methodology based on their findings.

Among the various modelling tools available, open-source software like **OpenFOAM** (**Open**-Source **F**ield **O**peration **A**nd **M**anipulation) has greatly facilitated the development of FCMs and FIMs. An example of the former is `darcyInterTransportFoam` (**Chapter 2**), a recently developed user-defined, fully-coupled three-dimensional (3D) model that uses the Navier-Stokes equations to solve the surface flow dynamics and Darcy's law to drive the groundwater flow. This model additionally simulates solute transport and heat transfer processes using separate transport and energy equations for each domain. The coupling between domains is achieved through the bidirectional mapping of conservative flux BCs at the SW-GW interface using an iterative scheme. As regards FIMs, **OpenFOAM** offers a wide range to choose from (OpenCFD Ltd, 2024). One of the most common for studying the continuous interaction between SW and GW is `interFoam` (Deshpande, et al., 2012). `interFoam` is a standard 3D transient solver designed for incompressible, immiscible, two-phase flows. This model simulates both surface and subsurface flow dynamics employing the Navier-Stokes equations, with an additional sink term in the subsurface to account for the energy losses caused by the porous medium. Furthermore, although `interFoam` does not allow the direct simulation of solute transport, this is still possible by coupling the standard `scalarTransport` function object. Both `darcyInterTransportFoam` and `interFoam` use the Volume of Fluid (VOF) phase-fraction based interface capturing approach (Hirt & Nichols, 1981) for solving the 3D Navier-Stokes equations.

In the present study, `darcyInterTransportFoam` (FCM) and `interFoam` (FIM) were used to simulate the SW-GW exchange across a three-dimensional, synthetic rippled streambed. The computational efficiency and performance of both solvers were

compared to assess their respective strengths and limitations in simulating complex SW-GW interactions. Specifically, the comparison focused on the differences and similarities in flow dynamics, solute transport fields and computational demands. This comparison provides a solid basis for which type of applications the models are suited. Additionally, this study further tests the efficacy of `darcyInterTransportFoam` for capturing high spatiotemporal SW-GW exchanges and provides insights into the potential of using `interFoam` to address these processes beyond the local-scale. Ultimately, the detailed examination of these models aims to advance the use of numerical simulations as robust complimentary tools to interpret field observations, thereby improving the sustainable and responsive management of water resources.

## 3.2. Flow and Transport Governing Equations

Both `darcyInterTransportFoam` and `interFoam` are implemented in the free and open-source Computational Fluid Dynamics (CFD) software **OpenFOAM** (Weller, et al., 1998). This powerful computational platform discretizes the solver-specific governing equations using the Finite Volume Method (FVM) in space and the Finite Difference Method (FDM) in time (Schulze & Thorenz, 2014). A description of the partial differential equations governing the flow and the transport of solutes in both numerical models is provided below.

### 3.2.1. darcyInterTransportFoam (Fully-Coupled Solver)

The sequential iterative coupling implemented in `darcyInterTransportFoam` allows to accurately capture the surface-subsurface feedback (**Figure 3.1a**), in contrast to the asynchronous linking approach commonly applied to study SW-GW interactions (e.g., Lee, et al., 2020; Trauth, et al., 2014). Such bidirectional exchange prevents potential errors in the mass balance, especially when the seepage across the sediment-water interface is significant. As a result, mass conversation is ensured in the system.

#### 3.2.1.1. Surface Water Modelling

The surface flow is modelled as a transient, incompressible, immiscible, two-phase flow using the VOF interface capturing approach. Accordingly, the two phases are separated by a continuous, free interface, whose location and shape arise from the phase fraction field, $\alpha_i$ (Schulze & Thorenz, 2015).

The two-phase flow is governed by five partial differential equations: the continuity equation (**Equation (3.1)**), the three Navier-Stokes equations (**Equation (3.2)**) and the $\alpha_i$ transport equation (**Equation (3.7)**) (Ubbink, 1997). Since the VOF method considers the

two immiscible fluids as one fluid, the resulting single-phase continuity and momentum equations are:

$$\nabla \cdot U_{sur,i} = 0 \tag{3.1}$$

$$\frac{\partial \rho_{sur,i} U_{sur,i}}{\partial t} + \nabla \cdot \left(\rho_{sur,i} U_{sur,i} U_{sur,i}\right) - \nabla \cdot \underbrace{\left[\mu_{sur,i}\left(\nabla U_{sur,i} + \left(\nabla U_{sur,j}\right)^T\right)\right]}_{\tau_{ij}} = -\nabla p_{rgh,i} - g \cdot x_i \nabla \rho_{sur,i} + \rho_{sur,i} S_{U_{sur,i}} \tag{3.2}$$

where $U_{sur,i}$ and $U_{sur,j}$ respectively denote the single-phase velocity field in the x and y directions (m/s), $t$ is the time (s), $\rho_{sur,i}$ and $\mu_{sur,i}$ are respectively the single-phase density (kg/m$^3$) and dynamic viscosity fields (N/m$^2$), $\tau_{ij}$ is the deviatoric viscous stress tensor field (N/m$^2$), $p_{rgh,i}$ is the modified pressure field (Pa), $g$ is the gravitational acceleration vector (m/s$^2$), $x_i$ is the cell position vector field (m) and $S_{U_{sur,i}}$ denotes any source/sink term field (m/s). $p_{rgh,i}$ substitutes the surface static pressure field, $p_{sur,i}$, in the Navier-Stokes equations. This alternative pressure avoids the occurrence of steep pressure gradients and facilitates the definition of pressure BCs by getting rid of the hydrostatic component of $p_{sur,i}$ (Rusche, 2003):

$$p_{rgh,i} = p_{sur,i} - \underbrace{\rho_{sur,i}(g \cdot x_i)}_{\substack{Hydrostatic \\ pressure}} \tag{3.3}$$

since $g$ only applies in the vertical direction, $x_i$ turns into the cell height field, $z_{sur,i}$ (m). In line with the one-fluid VOF approach, both $\rho_{sur,i}$ and $\mu_{sur,i}$ are determined as weighted averages based on the distribution of the phase fraction field, $\alpha_i$ (-), throughout the surface domain (Berberović, et al., 2009). Accordingly, the resulting single-phase parameters will be equal to the physical properties of each surface fluid in their corresponding occupied regions, only varying across the interface as follows:

$$\rho_{sur,i} = \alpha_i \rho_{w_{sur}} + (1 - \alpha_i)\rho_{a_{sur}} \tag{3.4}$$

$$\mu_{sur,i} = \alpha_i \mu_{w_{sur,i}} + (1 - \alpha_i)\mu_{a_{sur,i}} \tag{3.5}$$

where $\rho_{w_{sur}}$ and $\rho_{a_{sur}}$ are respectively the densities of the surface water and the air (kg/m$^3$) and $\mu_{w_{sur,i}}$ and $\mu_{a_{sur,i}}$ denote the surface water and air dynamic viscosity fields (N/m$^2$). The latter are obtained as the sum of the physical dynamic viscosity, $\mu_{phys_{p_{sur}}}$, and the turbulent dynamic viscosity field, $\mu_{turb,i}$:

$$\mu_{p_{sur,i}} = \mu_{phys_{p_{sur}}} + \mu_{turb,i} \tag{3.6}$$

where $p$ is the subscript of the respective surface phase, either $w$ (water) or $a$ (air). While $\mu_{phys_{p_{sur}}}$ has to be specified by the user, $\mu_{turb,i}$ is calculated by the selected turbulence model. **OpenFOAM** contains a wide suite of turbulence models, including Reynolds-Averaged Navier-Stokes (RANS), Large Eddy Simulation (LES) and hybrid

RANS-LES Detached Eddy Simulation (DES) models. Each turbulent approach will require from a different level of mesh resolution and computational effort to achieve an optimal solution. Consequently, the model choice will ultimately rely on the features of the problem to be addressed as well as the desired accuracy for the simulation. In this regard, the k-ω SST (Shear Stress Transport) RANS model was used in the present study (Menter, 1994). The mesh demands of RANS models are less restrictive than those of more advanced turbulence models, such as LES. This fact allows to decrease the computational time of the simulation while still achieving an accurate estimation of the different scales of turbulence. Such savings in computational effort makes RANS models a common choice in hydrogeological studies to solve the turbulent flow (e.g., Broecker, et al., 2021; Chen, et al., 2015; Janssen, et al., 2012).

The distribution of $\alpha_i$ throughout the surface domain and, thus, the position of the free surface, is determined via a transport equation extended by an additional convective term (Berberović, et al., 2009):

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot \left(\alpha_i U_{sur,i}\right) + \underbrace{\nabla \cdot \left[(1-\alpha_i)\alpha_i U_{r,i}\right]}_{Compression\ term} = 0 \tag{3.7}$$

where $U_{r,i} = U_{w_{sur,i}} - U_{a_{sur,i}}$ is the relative velocity field (m/s) between the two surface phases, designated as the compression velocity (Berberović, et al., 2009). The added term, referred to as compression term, provides an artificial convection to $\alpha_i$ within the interface region. This extra energy source derives in the compression of the free surface, which ultimately allows the adequate conservation of $\alpha_i$ within the simulation domain (Weller, 2002). A detailed derivation of the described approach can be found in Marquez-Damian (2012).

The resulting $\alpha_i$ in each computational cell of the surface domain is interpreted as follows:

$$\alpha_i = \begin{cases} 0 & \Rightarrow \ Air \\ 0 < \alpha_i < 1 & \Rightarrow \ Transition\ zone,\ both\ air\ and\ water \\ 1 & \Rightarrow \ Water \end{cases} \tag{3.8}$$

The free surface is considered to be located within the transition zone, specifically around the mesh region where $\alpha_i = 0.5$.

The surface hydraulic head field, $h_{sur,i}$ (m), is subsequently calculated using Bernoulli's equation based on the $U_{sur,i}$ and $p_{sur,i}$ results:

$$h_{sur,i} = \frac{U_{sur,i}^2}{2 \cdot |g|} + \frac{p_{sur,i}}{\rho_{sur,i} \cdot |g|} + z_{sur,i} \tag{3.9}$$

Regarding the conservative solute transport, this process is solved using the two-phase simulation option provided by the model (**Section 2.2.1.1.2**). The selected approach limits the simulation of the surface solute transport to a single phase, using the

$\alpha_i$ distribution to define the extension of the computation. Accordingly, the conservative transport is only solved at those cells whose $\alpha_i = 1$. The applied advection-diffusion equation has the following form:

$$\frac{\partial C_{sur,i}}{\partial t} + \nabla \cdot \left(\alpha_i U_{sur,i} C_{sur,i}\right) - \nabla \cdot \left(D_{sur,i} \nabla C_{sur,i}\right) = \alpha_i S_{C_{sur,i}} \tag{3.10}$$

where $C_{sur,i}$ is the surface concentration field (kg/m$^3$), $D_{sur,i}$ is the surface diffusion coefficient field (m$^2$/s) and $S_{C_{sur,i}}$ accounts for any concentration source/sink field (kg/m$^3$). In the present study, the $D_{sur,i}$ was specified by a constant value for the entire surface domain (Section 3.3.2).

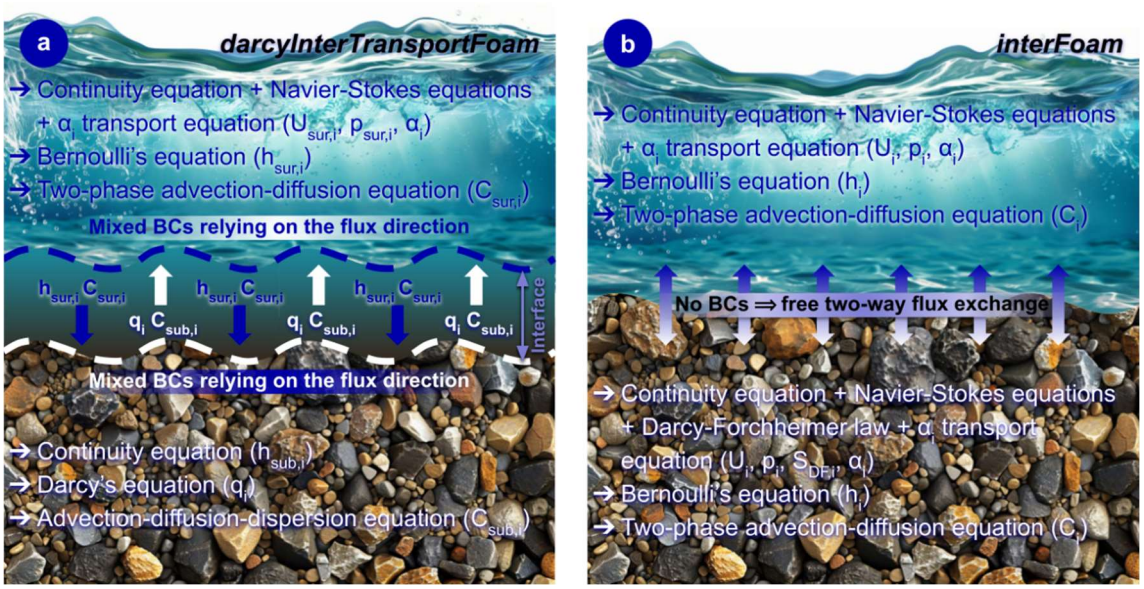



**Figure 3.1. Sketch of the two numerical SW-GW modelling approaches compared in this study: (a) sequential iterative coupling applied in `darcyInterTransportFoam` and (b) global implicit coupling implemented in `interFoam`.**

### 3.2.1.2. Groundwater Modelling

**`darcyInterTransportFoam`** assumes the subsurface domain to be fully saturated regardless the changes of head over time. As a result, the water table location does not vary throughout the simulation and, thus, the 3D groundwater flow can be simulated as steady-state by the following continuity equation:

$$\nabla \cdot \left(-K_{ij} \nabla h_{sub,i}\right) = 0 \tag{3.11}$$

where $K_{ij}$ is the hydraulic conductivity tensor field (m/s) and $h_{sub,i}$ the subsurface hydraulic head field (m). While $K_{ij}$ can be defined as heterogeneous and anisotropic, $h_{sub,i}$ is obtained from Bernoulli's equation after cancelling the kinetic term as a result of the small flow velocities in the ground. The steady-state condition makes the $h_{sub,i}$

solution to rely on the transient surface hydrodynamics and the assigned BCs at the subsurface domain.

The specific discharge field (Haitjema & Anderson, 2016), $q_i$ (m/s), is sequentially computed using Darcy's law based on the resulting $h_{sub,i}$:

$$q_i = -K_{ij} \nabla h_{sub,i} \tag{3.12}$$

Darcy's law only applies to very slow moving groundwaters, which means that the simulated subsurface flow is considered to be laminar (Reynolds number smaller than a value from 1 to 10).

As for the transport of conservative solutes in the underground domain, the model uses the subsequent advection-diffusion-dispersion equation to conduct their simulation through the porous medium:

$$\frac{\partial C_{sub,i}}{\partial t} + \nabla \cdot \left( U_{sub,i} C_{sub,i} \right) - \nabla \cdot \left( D_{sub,ij} \nabla C_{sub,i} \right) = S_{C_{sub,i}} \tag{3.13}$$

where $C_{sub,i}$ is the subsurface concentration field (kg/m$^3$), $U_{sub,i} = q_i / \theta_i$ is the porewater velocity field (m/s), where $\theta_i$ is the porosity field (-), $D_{sub,ij}$ is the hydrodynamic dispersion coefficient tensor field (m$^2$/s) and $S_{C_{sub,i}}$ denotes any concentration source/sink field (kg/m$^3$). Analogously to the surface domain, the $D_{sub,ij}$ was defined in this study as uniform and constant through a single value (Section 3.3.2).

Further information on the equations used by `darcyInterTransportFoam` can be found in Section 2.2.1 of Chapter 2.

### 3.2.2. interFoam (Fully-Integrated Solver)

`interFoam` is commonly applied to simulate the free-surface turbulent flow in SW-GW studies (e.g., Broecker, et al., 2017; Sobhi Gollo, et al., 2022). Besides solving the two-phase dynamics in the surface, this solver also allows to model the flow through porous media (e.g., Dichgans, et al., 2023). This fact makes `interFoam` suitable to investigate SW-GW systems in a fully-integrated way by setting separated surface and subsurface regions in the simulation mesh. Accordingly, flow and transport processes are modelled as a continuum throughout the domain, with no need for coupling BCs at the interface between mesh regions (i.e., global implicit coupling; Figure 3.1b).

#### 3.2.2.1. Surface Water Modelling

Analogously to `darcyInterTransportFoam`, `interFoam` uses the VOF phase-fraction based interface capturing method to solve the governing equations of the surface flow, namely, the continuity equation, the 3D Navier-Stokes equations and the phase fraction,

$\alpha_i$, transport equation. Therefore, no further description of the flow equations is included in this section, being the reader suggested to check Section 3.2.1.1 of this study for additional details.

As regards the transport of solutes, since `interFoam` does not provide an application for its simulation, this was done through the two-phase advection-diffusion equation contained in the standard `scalarTransport` function object. This transport equation matches that used by `darcyInterTransportFoam`, so its form can be checked in Equation (3.10). Moreover, the function object was slightly modified to include the solute flux computation.

#### 3.2.2.2. Groundwater Modelling

`interFoam` simulates the defined underground zones as a porous continuum, where a sink term, $S_{DF,i}$, is added to the Navier-Stokes equations to account for the energy losses caused by the porous medium. This resistance source term is obtained from the Darcy-Forchheimer equation (Whitaker, 1996):

$$S_{DF,i} = \nabla p_i = -\left( \underbrace{\mu_w D_i}_{\substack{Viscous \\ loss\ (1)}} + \underbrace{\frac{\rho_w |U_i|}{2} F_i}_{\substack{Inertial \\ loss\ (2)}} \right) U_i \tag{3.14}$$

where $\mu_w$ is the dynamic viscosity of the water (kg/m·s), $D_i$ is the Darcy coefficient vector field (1/m$^2$), $\rho_w$ is the water density (kg/m$^3$), $U_i$ is the velocity field (m/s) and $F_i$ is the Forchheimer coefficient vector field (1/m). Both terms of the equation create a pressure drop ($\nabla p_i$) proportional to the velocity (1) and velocity squared (2). Since the flow conditions in the subsurface region of the model are assumed to be laminar, only the Darcy contribution has been considered in this study ($F_i = 0$). As such, the inertial loss is cancelled from Equation (3.14) and, thus, $S_{DF,i}$ only accounts for the linear behaviour between $U_i$ and $\nabla p_i$:

$$S_{DF,i} = -\mu_w D_i U_i = -\frac{\mu_w}{k_i} U_i \tag{3.15}$$

where $k_i$ is the intrinsic permeability vector (m$^2$) of the porous media and the inverse of $D_i$. $S_{DF,i}$ is, therefore, inferred from Darcy's law considering the relationship between $k_i$ and the hydraulic conductivity, $K_i$:

$$K_i = k_i \left( \frac{\rho_w g}{\mu_w} \right) \tag{3.16}$$

Note that $D_i$ is defined as a vector field and, thus, so does $K_i$. This definition opposes that of `darcyInterTransportFoam`, where the hydraulic conductivity, $K_{ij}$, is specified as a tensor field.

Both $D_i$ and $F_i$ coefficients must be specified within an `explicitPorositySource` dictionary in the *fvOptions* file (*constant* directory). Additionally, a selection of the model zone where to apply the Darcy-Forchheimer source term (subsurface mesh region) must be also provided within the same dictionary.

In line with the fully-integrated computation of the water dynamics, the conservative solute transport in the subsurface region of the mesh was simulated by the same modified **`scalarTransport`** function object as that used in the surface zone.

Additional information about **`interFoam`** and the **`scalarTransport`** function object is provided in the online **OpenFOAM** User's Guide (OpenCFD Ltd, 2023).

## 3.3. Methodology

The capability to simulate SW-GW interaction processes of both integral and coupled models was evaluated and compared across a 3D synthetic streambed (Figure 3.2). A description of all the mesh features, as well as the overall setup of both models (i.e., model parameters, initial and boundary conditions, simulation controls, etc.), is provided below. Additionally, the mesh locations, variables and data types used for their comparison are also detailed in this section.

### 3.3.1. Mesh Features

Three-dimensional surface and subsurface meshes, formed by structured, hexahedral blocks, were defined for both model cases using the **OpenFOAM** mesh generation utility **`blockMesh`**. A similar methodology to that applied in Chapter 2 was used this time to generate all the mesh-related dictionaries. Therefore, both surface and subsurface *blockMeshDict* files were created through a suite of **MATLAB** scripts (Section 4.3) based on the imported riverbank point data and additional mesh information (e.g., vertical dimension and resolution, riverbed geometry or grid refinement). The interface patch is described by a symmetrical, sinusoidal rippled shape consisting of steady longitudinal ripples all along the boundary and evolving transversal geometries from both longitudinal ends towards the centre of the domain (Figure 3.2). This geometry follows the typical riffle-pool sequence commonly used in SW-GW interaction studies. Flat inlet and outlet boundaries were established to prevent potential stability issues in the simulation of the flow dynamics. Additional **MATLAB**-generated dictionaries include *topoSetDict*, *refineMeshDict* and *createPatchDict*, all used along with *blockMeshDict* to build the desired meshes, and *setFieldsDict*, to assign input data.

The application of all the aforementioned actions resulted in two rectangular meshes of 9.8 m long by 2.5 m wide and respective heights of 1.4 m and 2 m for the surface and

subsurface domains (Figure 3.2). Both meshes were subsequently merged in the `interFoam` case to conduct the fully-integrated simulation. Moreover, the riverbed ripples were defined using a continuous pattern, with no distance in between, and length and height dimensions of ~1 m (in both horizontal directions) and ~0.2 m, respectively.

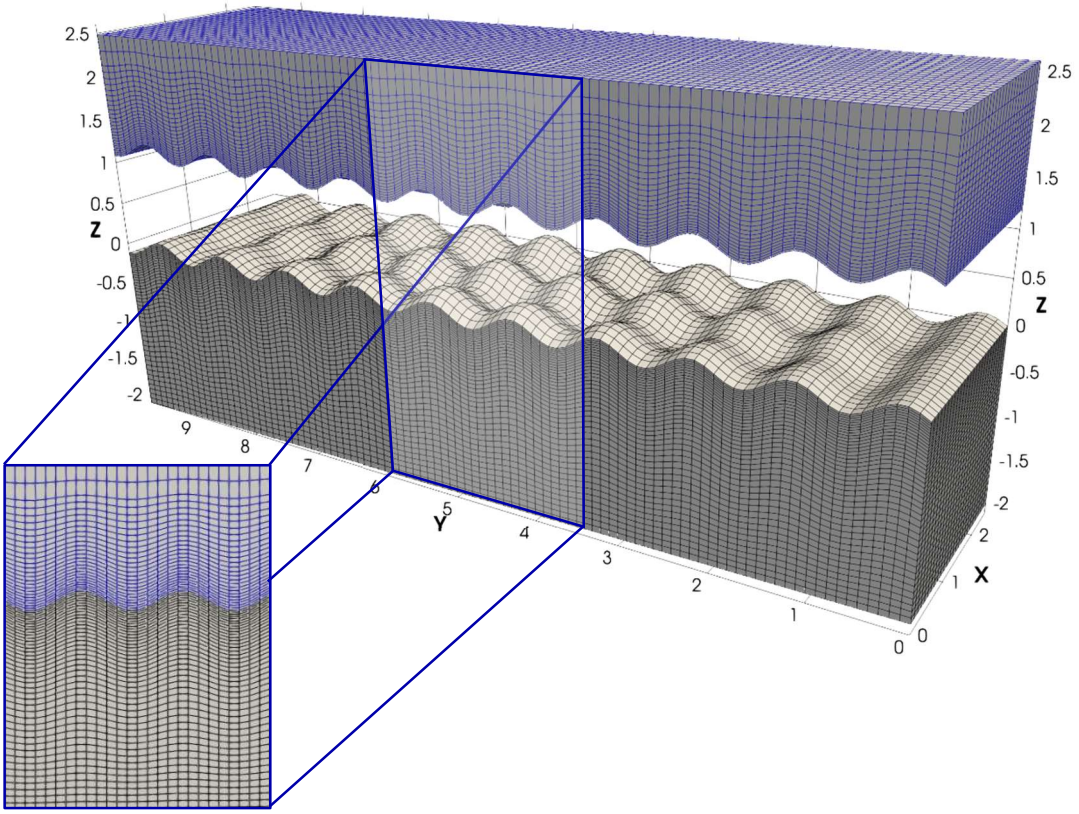



Figure 3.2. Simulation mesh generated with `blockMesh`.

The resolution of each mesh was set based on a prior mesh sensitivity analysis. As a result, two gradually refined meshes with an average cell size of 0.1x0.1x0.1 m were selected for the surface and subsurface domains. The gradual refinement was vertically implemented towards the interface of both domains through different multi-grading configurations (Figure 3.2). Additionally, the vertical dimension was later split in two using a different `refineMesh` setup for each domain. In this way, the whole subsurface mesh was vertically divided in half, whereas only those cells potentially contained within the water-phase fraction were subject to a double refinement in the surface domain. The no-refined surface elements correspond to the air-phase, whose detailed process simulation is not required in the present study. This phase was only simulated to capture the surface water level fluctuations, which have proved to have a major impact on the pressure field (e.g., Broecker, et al., 2021; Sobhi Gollo, et al., 2022; Trauth, et al., 2015). The maximum face area resulting from all the aforementioned refinements reaches 0.02 $m^2$ within the surface air-phase and 0.01 $m^2$ at the bottom of the subsurface domain, whereas a minimum face area of 0.002 $m^2$ is achieved right at the interface in both domains. Such

small element sizes in the vicinity of the water-sediment interface allow to accurately model the complex exchange dynamics and satisfy the y+ (dimensionless wall distance) close to the bed in the surface domain. The resultant mesh resolution leads to a surface mesh formed by 63700 cells and a subsurface mesh of 102900 cells. Furthermore, the number of elements in the longitudinal and transversal directions are the same in both meshes (i.e., 98 and 25, respectively), which facilitates the effective coupling of the nodes located at the interface of the `darcyInterTransportFoam` case. The cell number of the `interFoam` model will result from summing the cells of the two above-mentioned meshes (166600 elements).

### 3.3.2. Model Parametrization

The same constant fluid and geological parameters were defined for both cases in line with the steady-state scenario simulated by each model. Accordingly, the two simulated fluids (i.e. air and water) were assumed to be Newtonian, as previously stated in Section 3.2. Moreover, standard values of physical kinematic viscosity ($\nu_{phys_p}$) and density ($\rho_p$) at ambient temperature were set for both phases (Table 3.1). As for the solute transport parametrization, a $D_{sur,i}$ and $D_{sub,ij}$ of $4{\cdot}10^{-4}$ m$^2$/s were respectively prescribed to the surface and subsurface fluids. The definition of the latter value differs between models. Whereas in the `darcyInterTransportFoam` model, $D_{sur,i}$ and $D_{sub,ij}$ were specified in the corresponding *transportProperties* dictionary of each domain, a single $D$ value was defined within the `scalarTransport` function object dictionary in the `interFoam` case. The entire set of surface fluid properties is shown in Table 3.1.

Table 3.1. Surface fluid parameters of both model cases.

| Fluid phases | Parameter | Value | Dimensions |
|---|---|---|---|
| Water | Transport model | Newtonian | |
| | Physical kinematic viscosity ($\nu_{phs_w}$) | $10^{-6}$ | m$^2$/s |
| | Density ($\rho_w$) | 1000 | kg/m$^3$ |
| Air | Transport model | Newtonian | |
| | Physical kinematic viscosity ($\nu_{phys_a}$) | $1.48{\cdot}10^{-5}$ | m$^2$/s |
| | Density ($\rho_a$) | 1 | kg/m$^3$ |
| Both | Surface tension ($\sigma$) | 0.07 | J/m$^2$ |

Regarding the underground characterization, the subsurface domain was divided into two layers of different homogeneous and isotropic $K_{ij}$ and $D_i$ (Figure 3.3). Values of $K_{ij}$= $3{\cdot}10^{-3}$ m/s and $D_i$= $3.27{\cdot}10^{9}$ 1/m$^2$ were respectively defined for the 0.5 m wide top layer

of the **darcyInterTransportFoam** and **interFoam** cases, whereas a $K_{ij}$ of $5 \cdot 10^{-3}$ m/s and a $D_i$ of $1.96 \cdot 10^{9}$ 1/m$^2$ were prescribed to the lower part of the domain. The upper layer definition was done via the **setFields**/**topoSet** action `rotatedBoxToCell`, similarly to some initial field assignments (**Section 3.3.3**). Moreover, $\theta_i$ was assumed to be 1 throughout the whole domain. This is so because the two-phase advection-diffusion equation (**Equation (3.10)**) implemented in the **scalarTransport** function object does not consider $\theta_i$ to compute the transport of species. Consequently, the default unitary porosity field was applied to the fully-coupled case in order to adapt the transport simulation to that of **interFoam** and, thus, be able to adequately compare the solute outcomes from both models.

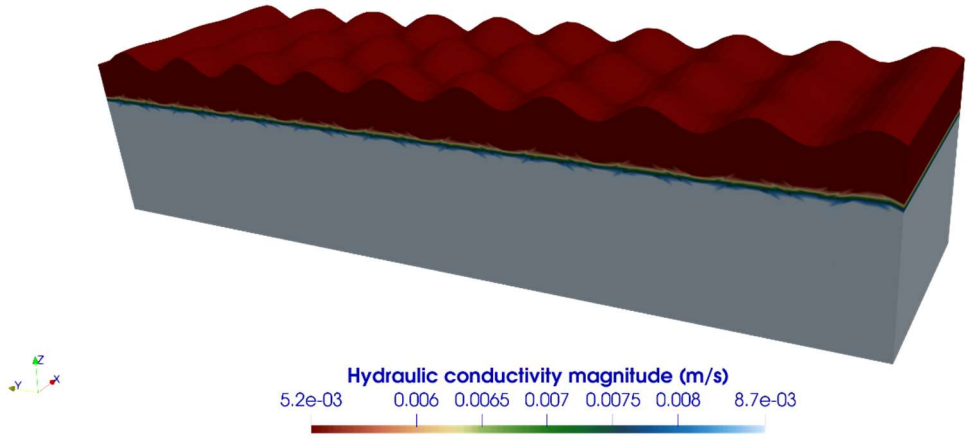



**Figure 3.3. Subsurface geological characterization of both models.**

### 3.3.3. Initial and Boundary Conditions

Uniform and nonuniform fields were initially prescribed to all surface and subsurface variables in order to reach the steady-state faster in both model cases. Whereas the uniform fields were directly specified via a single value in the corresponding variable files (e.g., k=0.2 m$^2$/s$^2$ or omega=4.5 s$^{-1}$), the nonuniform fields were established through different **setFields** actions (e.g., alpha.water and C). These include `rotatedBoxToCell` and `fieldToCell`, which allow to assign user-defined field values to mesh regions delimited by a specified rotated box and field range, respectively.

As regards the boundary conditions, they stay constant over time according to the steady-state flow simulated with both **darcyInterTransportFoam** and **interFoam**. The same BCs were applied to the two simulated cases except at the `riverbed`, namely, the surface-subsurface interface. At this patch, a periodic BC was assigned to all variables in the **interFoam** model, whereas a different BC was specified for each variable in the

**darcyInterTransportFoam** case. Although the FIM does not require coupling BCs at the domains' interface, these were still included for the posterior analysis of the water and solute exchanges. The collection of interfacial BCs of the fully-coupled case, as well as the rest of surface conditions prescribed to both model cases, match those of Chapter 2 (Section 2.3.3.3). Consequently, the surface domain was set up as an open system, where a constant water inflow of 2 $m^3/s$ with a solute concentration of 2 $kg/m^3$ enter the domain through the lower patch (`inletWater`) of a two-fraction inlet, interact with the groundwater through the rough interface patch (`riverbed`) and freely leave the system through the `outlet` patch. A water height of 0.95 m was fixed at this latter boundary via `hRef` along with a hydrostatic pressure condition. The top patch (`atmosphere`) as well as the upper fraction of the inlet (`inletAir`) were defined as open boundaries under atmospheric pressure conditions. Contrary to the aforementioned setup, the subsurface domain was conceptualized as a closed system, where water and solutes can only flow in and out of the domain through the `riverbed` patch. Accordingly, the remaining boundaries (i.e., `inlet`, `outlet`, `banks` and `aquifer`) contain slip conditions. The surface `banks` were also set as slip. The full set of boundary conditions applied to both model cases is included in Appendix C of this document (Table C.1-Table C.5) along with a brief description of each of them. Additional details about each above-mentioned BC can be acquired in the **OpenFOAM** User's Guide (OpenCFD Ltd, 2023).

### 3.3.4. Numerical Schemes, Linear Solvers and Solution Algorithms

The spatial and temporal discretization schemes applied to the different terms of the flow and the solute transport governing equations were specified in the corresponding *fvSchemes* dictionaries of each model. The suite of numerical schemes assigned to the surface domain of the **darcyInterTransportFoam** case matches that prescribed to the **interFoam** case. Nonetheless, the subsurface scheme collection for the former case slightly differs from its surface set. In this way, a unique and stable Gauss upwind scheme was specified for all the advection terms of the equations, while the rest of schemes remain the same as those of the surface domain. Overall, the choice of the discretization scheme will be a compromise between solution precision and stability of the computation. Thereby, some schemes will enhance the former in detriment of the latter, whereas others will boost stability over precision. In the present study, the temporal terms of the surface and subsurface equations were discretized with a Euler scheme. Furthermore, diverse numerical schemes were defined for the advection terms of the different surface equations. These include, e.g., a Gauss upwind scheme for the advective components of the Navier-Stokes equations and the two-equation turbulence model or a Gauss vanLeer scheme for the advection of $\alpha_i$. As regards the diffusion terms, a non-orthogonality corrected version of the Gauss linear scheme was used in both domains of the models. The full set of defined discretization schemes is shown in Table 3.2.

Table 3.2. Numerical schemes selected to discretize the surface and subsurface (purple) equation terms of both `darcyInterTransportFoam` and `interFoam` models.

| Scheme types | Equation terms | Numerical schemes |
|---|---|---|
| ddtSchemes | default | Euler |
| gradSchemes | default | Gauss linear |
| divSchemes | default | Gauss linear / Gauss upwind |
| | div(rhoPhi,U) | Gauss upwind |
| | div(phi,C) | Gauss upwind |
| | div(phi,alpha) | Gauss vanLeer |
| | div(phirb,alpha) | Gauss linear |
| | div(phi,k) | Gauss upwind |
| | div(phi,omega) | Gauss upwind |
| | div(((rho*nuEff)*dev2(T(grad(U))))) | Gauss linear |
| laplacianSchemes | default | Gauss linear corrected |
| interpolationSchemes | default | linear |
| snGradSchemes | default | corrected |

Concerning the linear-solvers use to solve the discretized equations and their related preconditioners and tolerances, these were specified in the *fvSolution* dictionaries of each model case. Similarly to the discretization schemes, the same setup is defined for the **`interFoam`** case and the surface domain of the **`darcyInterTransportFoam`** case. Furthermore, the solution suite used for the subsurface domain of the latter case will be different from that applied to the surface, as will be the system of linear equations to be solved. Thereby, the discretized equations of the momentum, the turbulence model and the solute transport were solved through an iterative solver using a run-time selectable smoother (`symGaussSeidel`). Moreover, a preconditioned conjugate gradient method for symmetric matrices (`PCG`) was used to solve the resulting linear systems of the Poisson and the groundwater continuity equations. The entire solution suite can be checked in Table 3.3.

Table 3.3. Numerical solvers selected to solve the surface and subsurface (purple) variables of both `darcyInterTransportFoam` and `interFoam` models.

| | p_rgh | h | U | alpha.water | k | omega | C |
|---|---|---|---|---|---|---|---|
| solver | PCG | | smoothSolver | | | | |
| preconditioner/smoother | DIC | | symGaussSeidel | | | | |
| tolerance | $10^{-7}$ | $10^{-6}$ | $10^{-8}$ | | | | $10^{-6}$ |
| relative tolerance (relTol) | 0.05 | 0 | 0 | | | | |

The iterative PIMPLE (PISO-SIMPLE) algorithm (OpenCFD Ltd, 2023) is used by both models to effectively solve the pressure-velocity coupling in the surface domain. This procedure first evaluates the initial solutions to then iteratively correct them according to the number of correctors specified in the corresponding *fvSolution* dictionaries of each case. In the present study, 3 inner-correctors (`nCorrectors`) and 1 outer-correction (`nOuterCorrectors`) were prescribed for PIMPLE. Moreover, the subsurface solution was corrected for mesh non-orthogonality using 1 `nNonOrthogonalCorrectors` in the **`darcyInterTransportFoam`** case.

The rest of the simulation controls were defined in the *controlDict* dictionary of each model's case. Initial and maximum timesteps of 0.001 and 1 seconds, respectively, as well as maximum Courant and alpha Courant numbers of 0.5 and 1 were specified in both cases. Furthermore, the timestep size was set to dynamically adjust according to the latter Courant parameters.

### 3.3.5. Comparison Setup

Several slices were defined for the comparison of both model outcomes (**Figure 3.4**). These include four vertical sections –two longitudinal and two transverse– along with six horizontal slices. All vertical sections comprise the surface water fraction and the subsurface sediment, excluding the surface air fraction due to its irrelevance to the present study. The longitudinal slices intersect the crest of the central ripples of the riverbed (x=1.25 m) and their adjacent left trough (x=0.75 m). The transverse slices are placed at the upstream (y=4.95 m) and downstream (y=5.45 m) sides of the 6$^{th}$ series of longitudinal ripples. As for the horizontal sections, one is located halfway between the water surface and the interface (z=0.5 m), another midway between the interface and the subsurface bottom (z=-1 m) and the remaining four in the vicinity of the interface, in both the surface water (z=0.2 m and z=0.1 m) and groundwater (z=-0.1 m and z=-0.2 m). All the specified slices are assumed to be representative of the results achieved with both **`darcyInterTransportFoam`** and **`interFoam`**.

In addition to the aforementioned 2D outputs, 1D results were also extracted at the intersections of the vertical slices with the different horizontal slices and the riverbed (**Figure 3.4**). Among the different variables computed by each model, the flow velocity, the hydraulic head and the solute concentration fields were chosen to assess the similarities and differences between the integral and coupled approaches. The analysis was completed with a 3D comparison of the flowpaths across the domains and the evolution of the solute concentration throughout the simulation. Furthermore, the computational efficiency of each modelling approach was also evaluated as part of the investigation. The above-mentioned outcomes and their corresponding discussion can be consulted in **Section 3.4**.

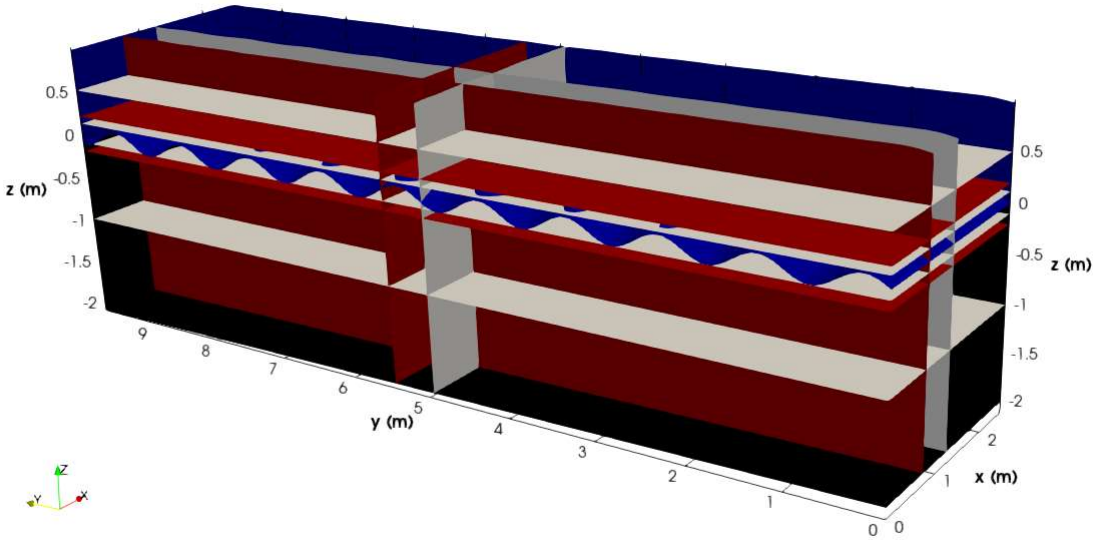



Figure 3.4. Vertical and horizontal slices (red and grey colour) defined to evaluate both model outcomes.

## 3.4. Results and Discussion

The flow velocity, hydraulic head and solute concentration fields computed by `darcyInterTransportFoam` and `interFoam` are evaluated and compared in this section, along with their respective computational demands. Both models ensure complete water and solute balances and are thus mass-conservative.

Initial solute concentrations of 0 and 2 kg/m$^3$ were respectively prescribed for the air and water phases in the surface domain of each model's case, while the groundwater concentration was initialized to 4 kg/m$^3$.

Among all the results obtained, the analysis presented below focuses on the longitudinal slice at x=1.25 m and its intersections with the specified transverse and horizontal slices, as well as the riverbed. Furthermore, the evaluation is complemented by the transversal slice at y=4.95 m and its intersection with the riverbed. The results at these two slices are considered representative of the overall performance of both modelling approaches. Additional results from all the defined slices and their multiple intersections are provided in Section C.2 of Appendix C.

Based on the obtained outcomes, the description and discussion of the results are divided into two subsections, each focusing on a different zone of the studied domain: the region far from the interface (Section 3.4.1) and the region close to the interface (Section 3.4.2). After that, the computational requirements of each model are evaluated in Section 3.4.3.

### 3.4.1. Water Dynamics and Solute Concentrations Far from the Interface

Figure 3.5 and Figure 3.6 respectively show the flow velocity magnitude of both models through the longitudinal slice at x=1.25 m and the transversal slice at y=4.95 m. In both cross-sections, the main differences in flow velocities between the two models are observed in the area near the interface, especially in the surface domain. This observation is supported by the results extracted along the intersections of the longitudinal slice at x=1.25 m with the two specified transversal slices (y=4.95 m and y=5.45 m). As illustrated in Figure 3.7, flow velocities in the x (A and B), y (C and D) and z (E and F) directions align closely along the vertical intersections until reaching the interface (around z=0 m), where the differences between both models emerge.

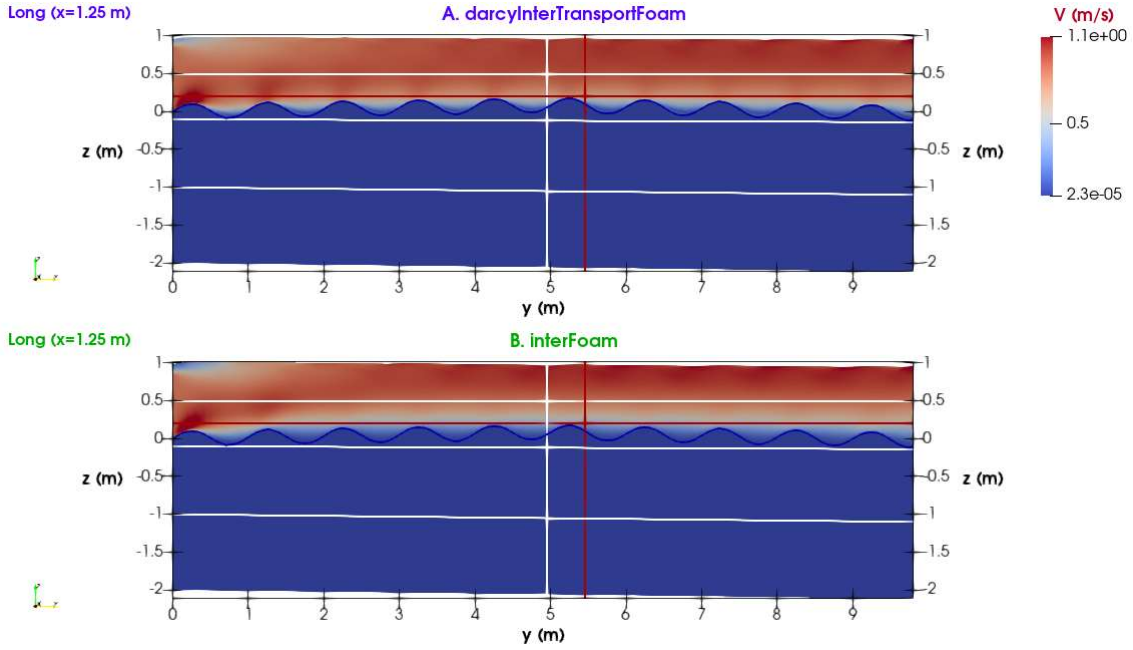



Figure 3.5. Flow velocity field (V) across the longitudinal slice at x=1.25 m. The red and white lines indicate the transversal and horizontal slices that intersect it (Figure 3.4).

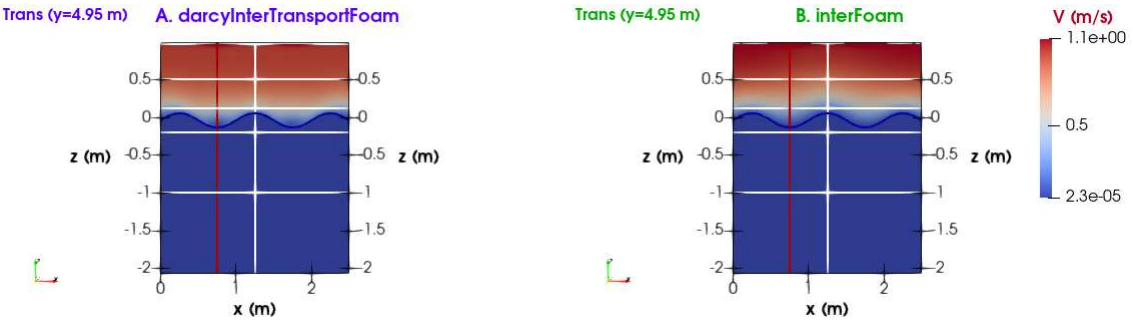



Figure 3.6. Flow velocity field (V) across the traversal slice at y=4.95 m. The red and white lines indicate the longitudinal and horizontal slices that intersect it (Figure 3.4).

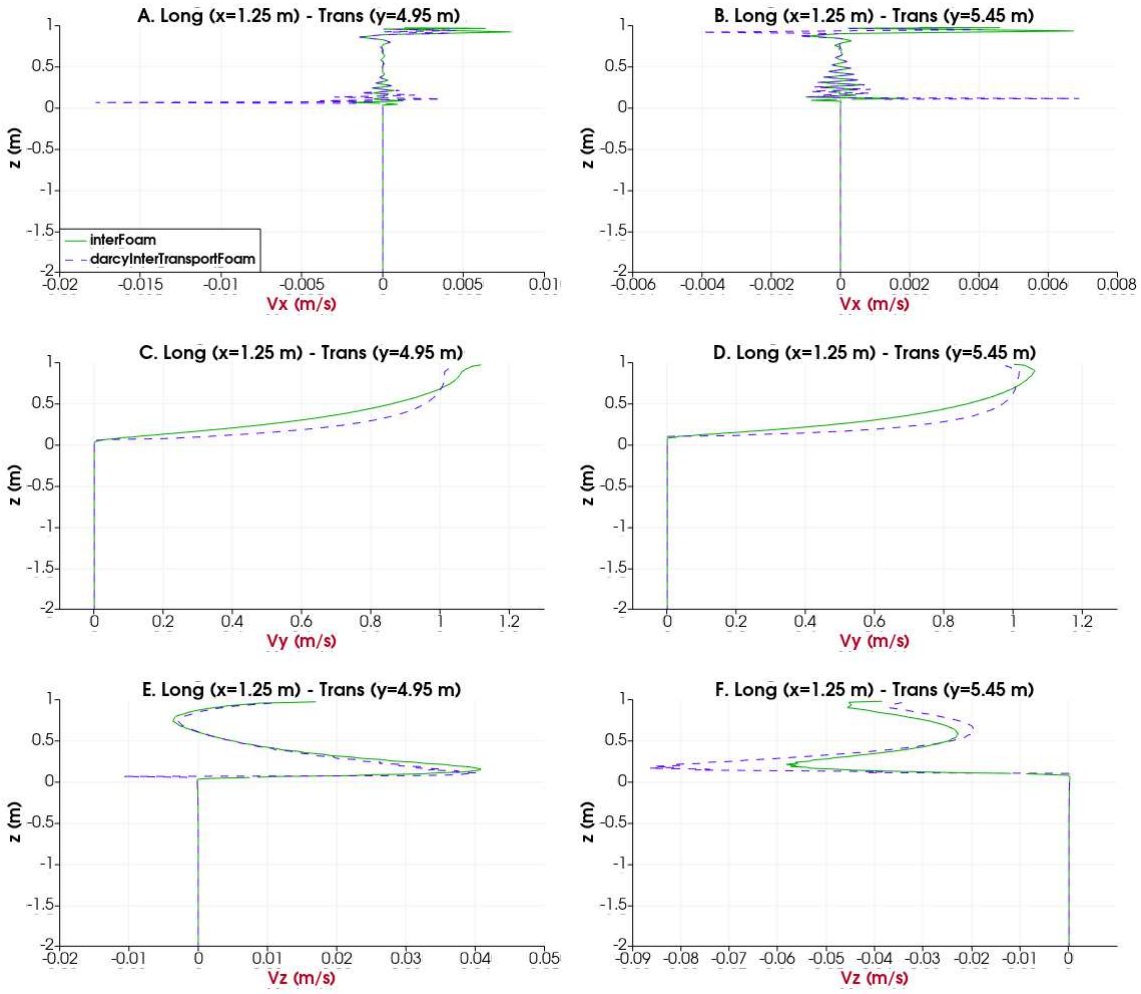



**Figure 3.7. Transversal (A, B), longitudinal (C, D) and vertical (E, F) flow velocities ($V_x$, $V_y$ and $V_z$, respectively) along the intersections of the longitudinal slice at x=1.25 m and the transversal slices at y=4.95 m and y=5.45 m.**

Longitudinal and vertical flow velocities ($V_y$ and $V_z$) along the intersections of the longitudinal slice with the various horizontal slices further confirm that, a few decimetres away from the interface (in both the SW and GW domains), the differences in flow velocities diminish and approach zero. In this way, Figure 3.8 shows maximum flow velocity differences in the y direction of ~0.1 m/s halfway between the water surface and the interface (A) and ~$10^{-5}$ m/s midway between the interface and the subsurface bottom (C). In contrast, near the interface, the maximum differences increase up to ~0.4 m/s in the surface domain (B) and ~$5{\cdot}10^{-5}$ m/s in the subsurface domain (D). The velocity discrepancies between the two models along the horizontal intersections are even smaller in the z direction, becoming negligible at those far from the interface (Figure C.6).

As expected, flow velocities are higher in the SW than in the GW. Furthermore, `darcyInterTransportFoam` generally provides higher velocity fields than `interFoam` in the surface domain (Figure 3.8-A and -B). However, this pattern is often reversed near

the interface (Figure 3.7) and in the subsurface domain (Figure 3.8-C and -D), where higher values are obtained with the FIM.

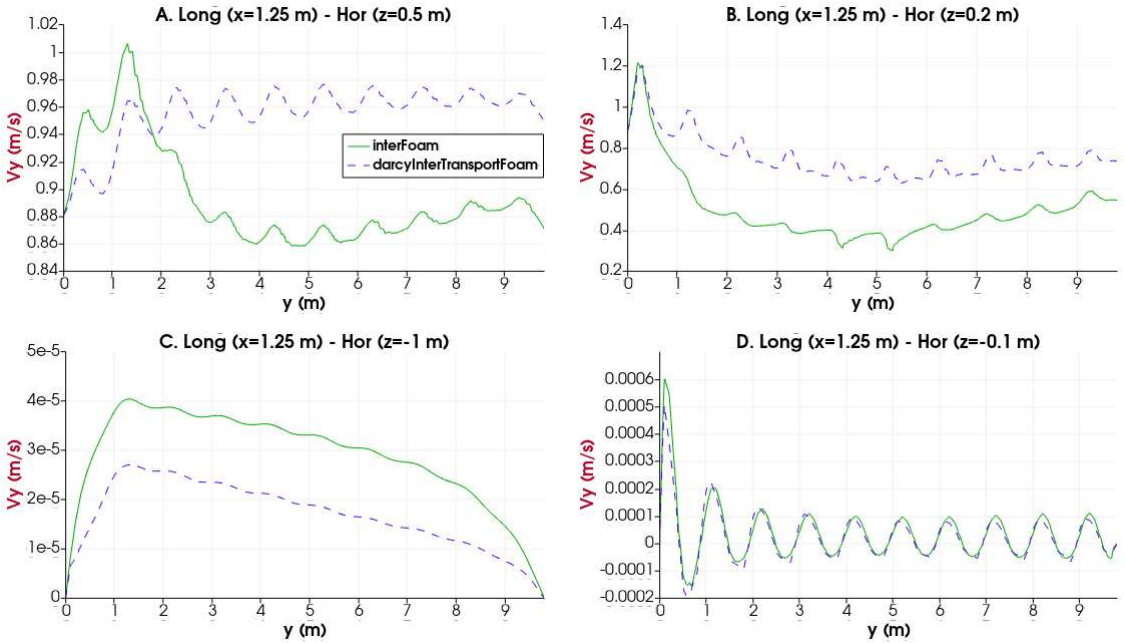



Figure 3.8. Longitudinal flow velocities ($V_y$) along the intersections of the various horizontal slices with the longitudinal slice at x=1.25 m.

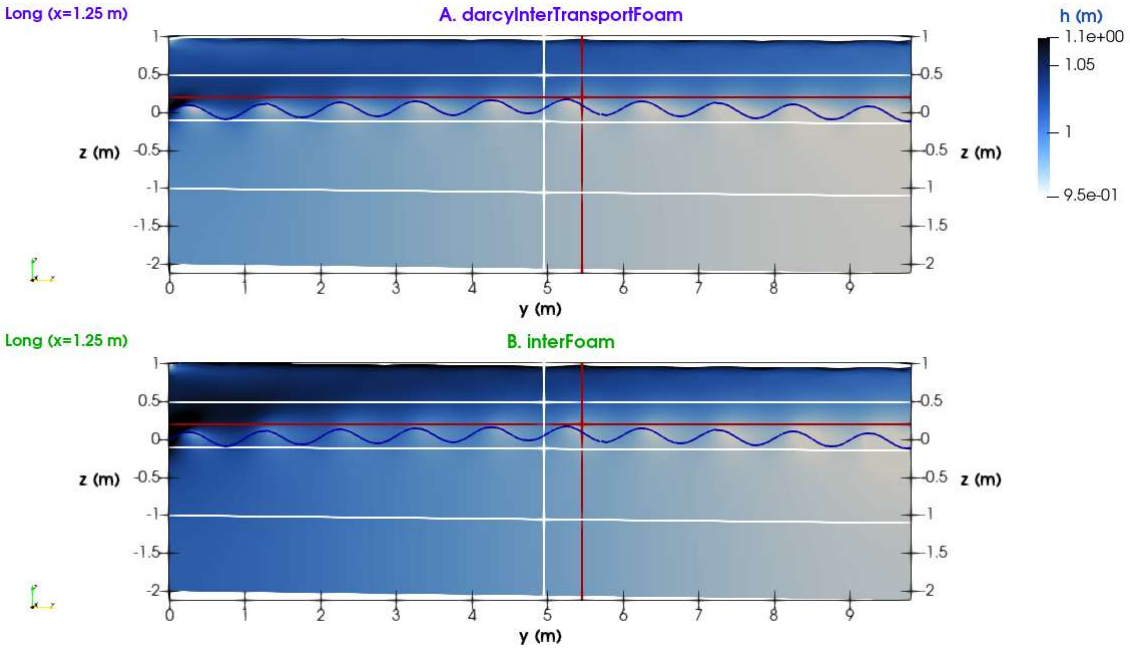



Figure 3.9. Hydraulic head field (h) across the longitudinal slice at x=1.25 m. The red and white lines indicate the transversal and horizontal slices that intersect it (Figure 3.4).

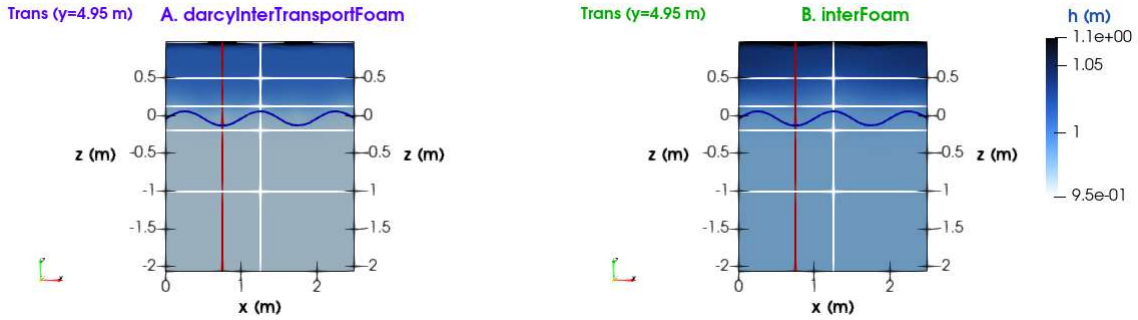



**Figure 3.10.** Hydraulic head field (h) across the traversal slice at y=4.95 m. The red and white lines indicate the longitudinal and horizontal slices that intersect it (Figure 3.4).

The velocity distribution described above, together with the static and hydrostatic pressure components, defines the resulting hydraulic head field across both the surface and subsurface domains (Figure 3.9 and Figure 3.10). The hydrostatic pressure remains approximately constant in both models throughout the entire simulation, maintained by a stable water level of around 0.95 m in the surface domain. As a result, a similar head pattern is observed in the surface domain, whereas slight differences emerge close to the interface and within the subsurface. These discrepancies are corroborated by the results extracted along the vertical intersections of the transversal slices at y=4.95 and y=5.45 m with the longitudinal slice at x=1.25 m. Figure 3.11-A and -B illustrate how the hydraulic heads of both models start diverging at the interface and how their dissimilarity increases up to around 0.02 m in the subsurface domain. The head trends also confirm the SW downwelling and GW upwelling through the upstream and downstream sides of the 6$^{th}$ riverbed ripple, respectively. Moreover, analogously to the groundwater velocity, `interFoam` yields higher head values than `darcyInterTransportFoam` in the subsurface. This fact is confirmed by the outcomes obtained along the horizontal intersections with the longitudinal slice at x=1.25 m. As shown in Figure 3.12-C and -D, the hydraulic head difference between both models is higher close to the inlet (~0.04 m at z=-0.1 m), decreasing along the domain until becoming almost negligible at the outlet. A similar behaviour is observed in the surface domain (Figure 3.12-A and -B). However, the discrepancy mainly appears at the inlet in the latter (~0.05 m at z=0.2 m), while the head remains nearly equal for both models along most of the domain.

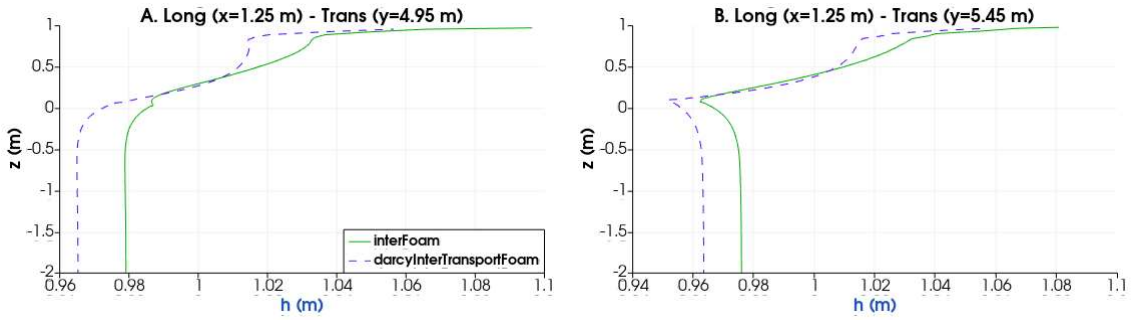



**Figure 3.11.** Hydraulic head (h) along the intersections of the longitudinal slice at x=1.25 m and the transversal slices at y=4.95 m and y=5.45 m.

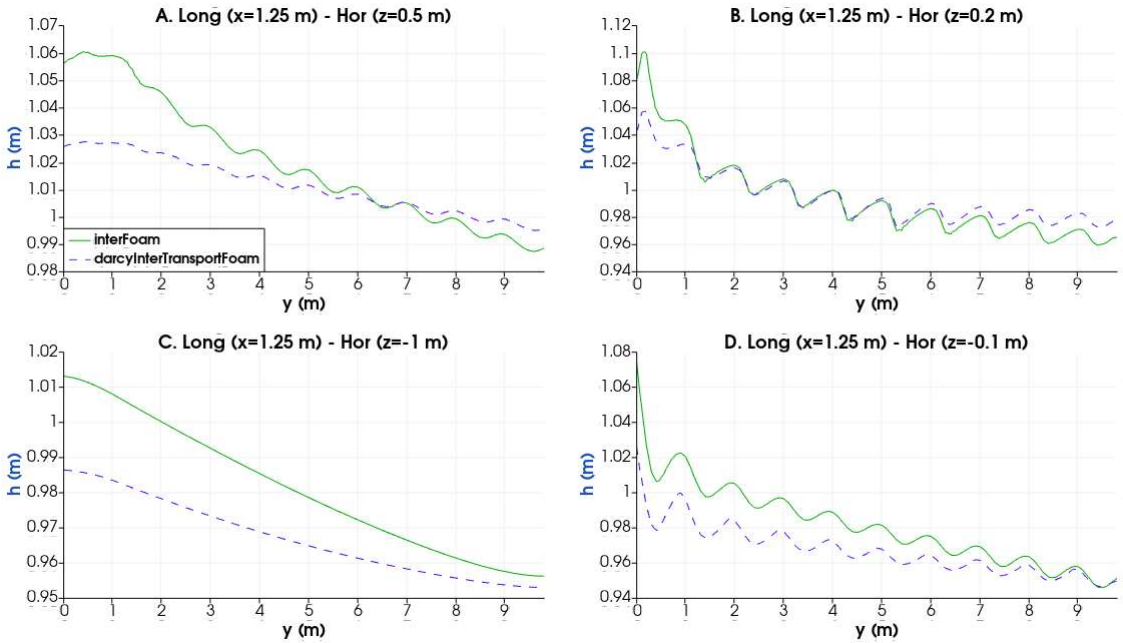



Figure 3.12. Hydraulic head (h) along the intersections of the various horizontal slices with the longitudinal slice at x=1.25 m.

Regarding the solute concentration, the results obtained at the end of the simulation show minimal differences between the two models. Within this similarity, `darcyInterTransportFoam` provides slightly higher final concentrations than `interFoam` in both domains, as shown across the longitudinal slice at x=1.25 m (Figure 3.14) and its vertical intersections with the transversal slices at y=4.95 and y=5.45 m (Figure 3.13). This somewhat higher concentration in `darcyInterTransportFoam` is further corroborated by the outcomes extracted at the multiple horizontal intersections with the aforementioned longitudinal slice (Figure C.25). Despite the similar concentration at the end of the simulation, the evolution of the solute plume over time differs between the two models. As illustrated in Figure 3.14, `interFoam` flushes out the high subsurface concentration (4 kg/m$^3$) almost twice as fast as `darcyInterTransportFoam`. This difference in travel times is directly linked to the faster groundwater flow obtained with the FIM, as previously described (Figure 3.8-C and -D).

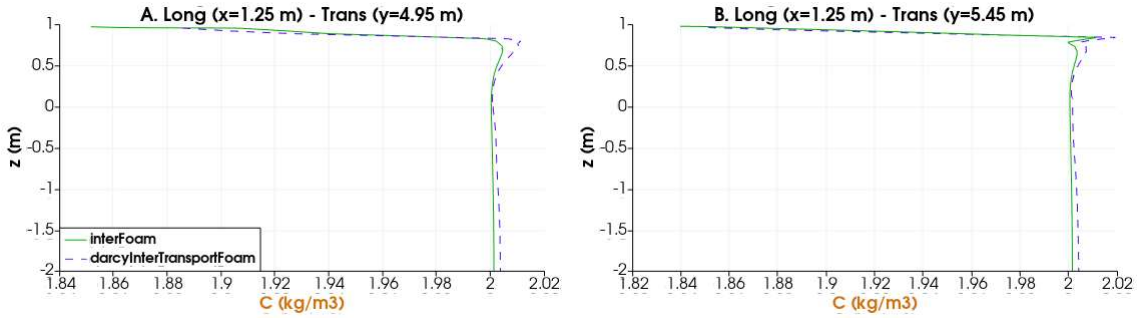



Figure 3.13. Solute concentration (C) along the intersections of the longitudinal slice at x=1.25 m and the transversal slices at y=4.95 m and y=5.45 m.

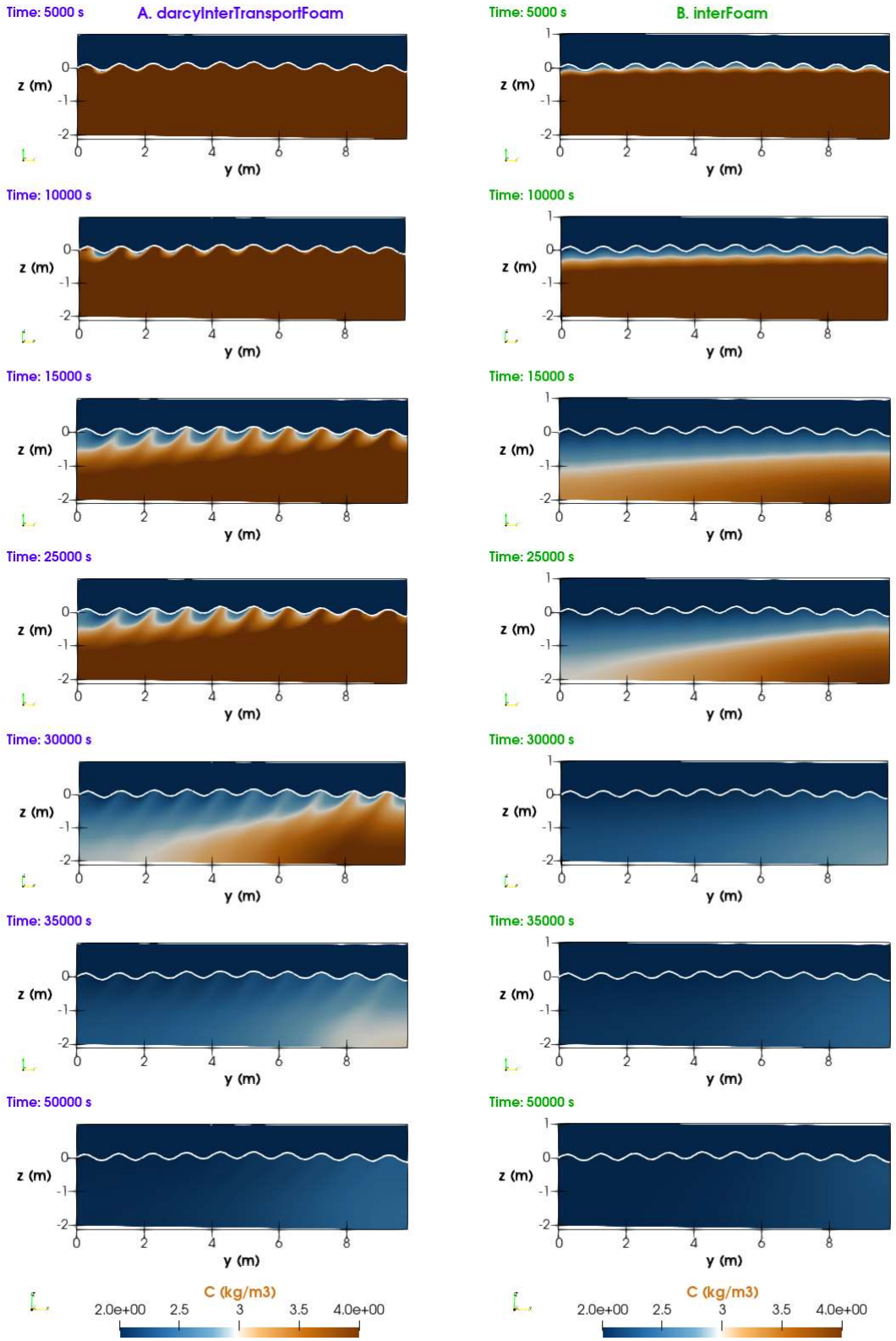



Figure 3.14. Solute concentration (C) evolution throughout the simulation across the longitudinal slice at x=1.25 m.

The aforementioned results demonstrate that both modelling approaches resolve the flow and solute processes similarly. Nevertheless, small discrepancies arise due to their different coupling methods and mathematical configurations. This latter aspect is especially relevant in the subsurface domain, where distinct sets of equations govern the water dynamics. In this regard, the combination of the Navier-Stokes equations with the Darcy-Forchheimer law results in higher groundwater velocities compared to Darcy's law. This faster subsurface flow achieved with `interFoam` leads to higher hydraulic heads and quicker solute travel times in the ground. As for the surface domain, both models use the same set of equations to simulate flow and solute processes. Despite this, dissimilarities between the models that originate at the interface propagate upwards, affecting the dynamic solution across the entire domain. This influence on the results is particularly significant in the region close to this SW-GW boundary. A few decimetres away from it, the models' differences fade and reach values close to zero. Consequently, the main source of discrepancy between the two models in the surface domain is located at the sediment-water interface. A detailed discussion of the reasons that explain its impact on the results is provided in the following section of this document.

### 3.4.2. Water Dynamics and Solute Concentrations Near the Interface

The flow velocities obtained along the intersections of the rippled riverbed with the longitudinal slice at x=1.25 m ($V_y$ and $V_z$) and the transversal slice at y=4.95 m ($V_x$ and $V_z$) are respectively displayed in Figure 3.15 and Figure 3.16. The depicted trends reveal significant differences between the outputs of both models along the main flow directions at each intersection. While `darcyInterTransportFoam` estimates velocities ranging from $-5.9 \times 10^{-5}$ to $6.7 \times 10^{-4}$ m/s, `interFoam` yields values between -0.022 and 0.13 m/s. This substantial difference in flow velocities leads to the dissimilarities observed in the region near the interface (Figure 3.7 and Figure 3.8). Such discrepancies stem from how the SW-GW interface is handled. `interFoam` simulates both the SW and GW domains using a single computational mesh. Since both domains are modelled as a continuum, no BCs are required at their interface (Section 3.3.3), which eliminates the physical constraints that these impose on fields. In particular, the absence of turbulence BC prevents the specification of roughness at the riverbed, and thus, the associated energy loss is not accounted for in the simulation. As a result, the surface flow velocities do not decrease as much as they do in `darcyInterTransportFoam` when approaching the interface, where BCs are prescribed to all fields to effectively couple both domains.

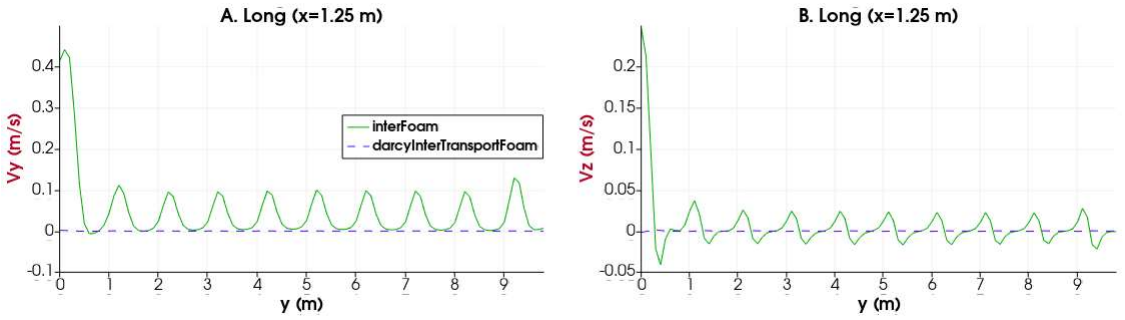



Figure 3.15. Flow velocities ($V_y$ and $V_z$) along the intersection of the longitudinal slice at x=1.25 m with the riverbed.

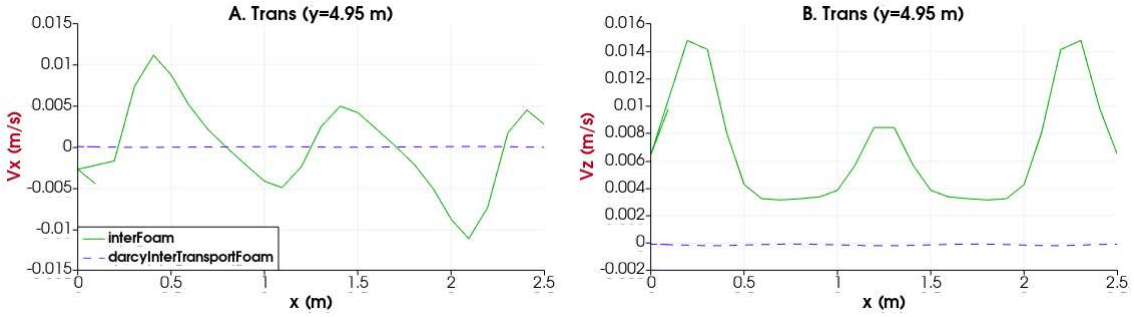



Figure 3.16. Flow velocities ($V_x$ and $V_z$) along the intersection of the transversal slice at x=4.95 m with the riverbed.

Despite the notable velocity dissimilarity at the SW-GW interface, its impact on both the hydraulic head and solute concentration distributions is relatively small. As illustrated in Figure 3.17, the head differences are minor at the previously selected intersections with the riverbed, with `interFoam` still producing slightly higher values due to the larger velocities. Likewise, even smaller discrepancies are observed in the solute concentration along the same intersections (Figure 3.18).

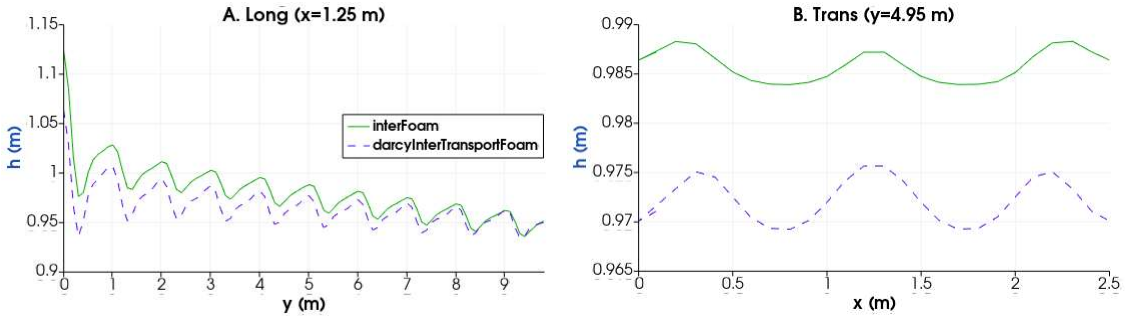



Figure 3.17. Hydraulic head (h) along the intersections of the riverbed with the longitudinal slice at x=1.25 m (A) and the transversal slice at y=4.95 m (B).

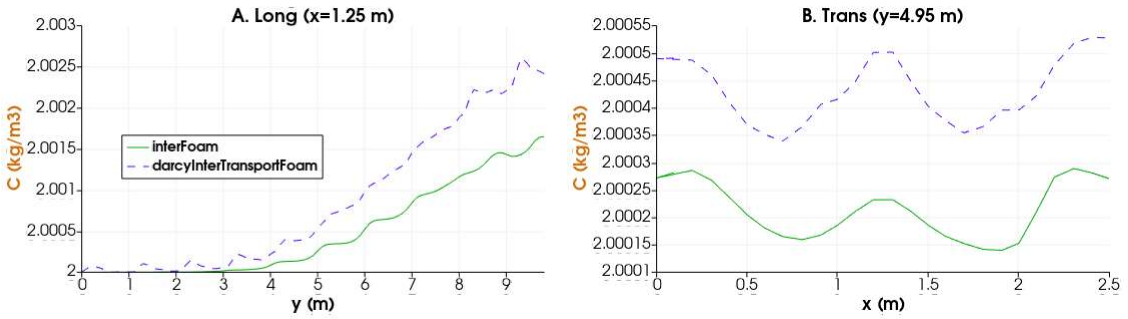



Figure 3.18. Solute concentration (C) along the intersections of the riverbed with the longitudinal slice at x=1.25 m (A) and the transversal slice at y=4.95 m (B).

A closer look at the region near the interface confirms the earlier described velocity results in the surface domain (Section 3.4.1). In this regard, higher flow velocities are observed within the troughs between the ripples in `darcyInterTransportFoam` than in `interFoam` (Figure 3.19). Moreover, the exhibited pattern aligns with the velocity trends extracted along the various vertical and horizontal intersections with the longitudinal slice at x=1.25 m (Figure 3.7 and Figure 3.8-B, respectively) as well as along its intersection with the riverbed (Figure 3.15). Therefore, despite `darcyInterTransportFoam` producing higher flow velocities throughout most of the surface domain, these velocities decrease dramatically as they approach the riverbed as a result of the coupling BCs specified at the SW-GW interface.

Regarding the groundwater flow, additional insights can be gained by separately looking at the subsurface domain. As shown in Figure 3.19, the flow velocity patterns differ slightly between the two models, particularly in the region close to the interface. In line with the results discussed in Section 3.4.1, the velocity field produced by `interFoam` is generally higher than that computed by `darcyInterTransportFoam`. Furthermore, the high interfacial velocities obtained with the FIM (Figure 3.15 and Figure 3.16) penetrate into the sediment, leading to groundwater velocities as high as 0.08 m/s. Such values exceed the limits of Darcy's law applicability (Reynolds number (−) > 10). Given the relatively small grain size of the subsurface top layer ($1.5 < d_{10} < 2$ mm), these elevated velocities are limited to the sediment adjacent to the interface, with a deeper penetration at the crest of the ripples (~2 cm) compared to the troughs between them (~1 cm). However, in scenarios with larger grain diameters, the SW turbulence could extend further into the GW, generating zones with velocities similar to those found in the surface domain. Such non-Darcy flows cannot be simulated by any groundwater model based on Darcy's law, such as `darcyInterTransportFoam`.

The resulting flowpaths exhibit a similar SW-GW exchange throughout the entire simulation domain in both model cases (Figure C.1). Moreover, they show how the surface water infiltrates from the upstream side of the ripples, travels through the sediment within the upper subsurface layer and exfiltrates back to the surface from the downstream side of the same structures (Figure 3.19).

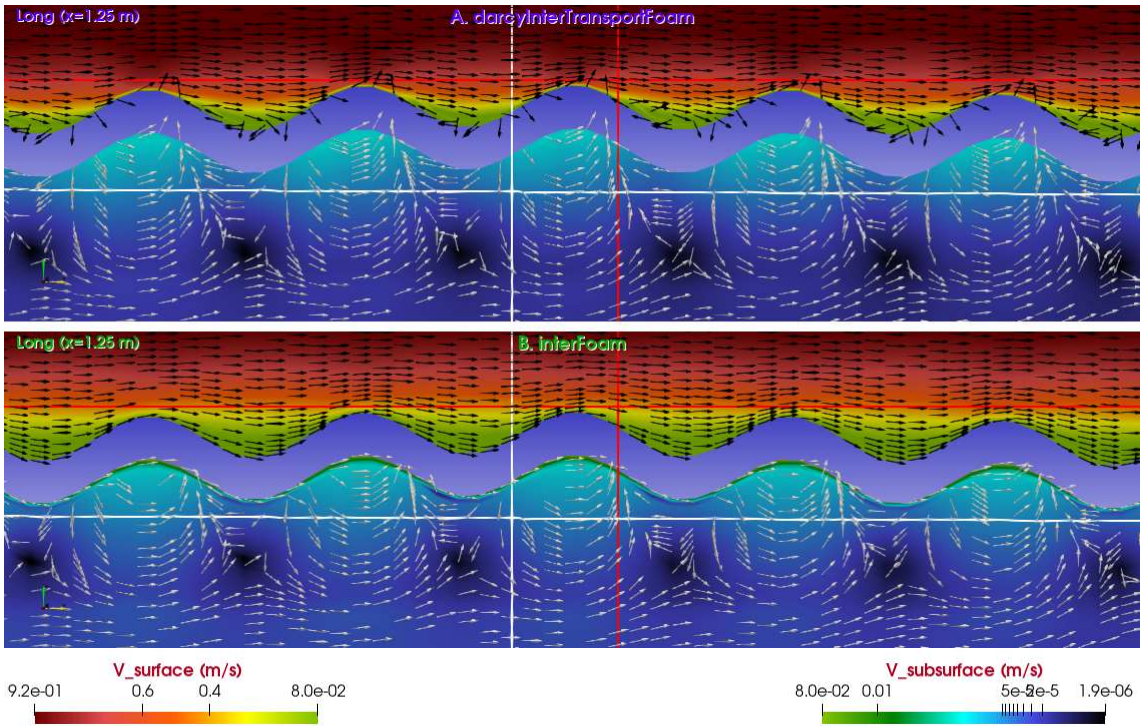



**Figure 3.19. Velocity field and flow directions near the interface across the longitudinal slice at x=1.25 m. The purple band separating the surface and subsurface domains is used to improve the visibility of the flow vectors originating from both domains. A separate colour scale is used for each domain. The red and white lines indicate the transversal and horizontal slices intersecting this region (Figure 3.4).**

The flow vector field further illustrates how the definition of the sediment-water interface affects the interaction dynamics across this SW-GW boundary. As shown in Figure 3.19A, the velocity vectors bend upon reaching the interface in `darcyInterTransportFoam`. Depending on the impact angle, they either enter the other domain or are deflected back, creating turbulent eddies in the trough between the ripples. In contrast, the high-velocity stripe at the top of the sediment functions as an underground channel in `interFoam` (Figure 3.19B), effectively acting as an extension of the surface water. As a result, the bidirectional exchange does not occur through the actual domain's interface, as it does in `darcyInterTransportFoam`, but rather across this non-Darcy flux zone within the subsurface domain. The pronounced $V_y$ gradient along this area traps most of the exchanges, orienting them in the longitudinal flow direction. This dominant flow orientation contrasts with that shown in the FCM, where multiple flux directions are observed across the interface. These different interface treatments result in slightly dissimilar water and solute flux exchanges, with `interFoam` yielding $3.7 \times 10^{-3}$ $m^3/s$ and $7.3 \times 10^{-3}$ kg/s, respectively, and `darcyInterTransportFoam` producing $2.9 \times 10^{-3}$ $m^3/s$ and $5.8 \times 10^{-3}$ kg/s.

Despite the above-mentioned discrepancies between both modelling approaches near the interface, their impact on the overall functioning of the system is relatively minor. This statement is supported by the similar flow and solute-related fields resulting from

the two models, as well as their total flux exchanges across the SW-GW interface. Given the comparable performance of both models despite their different coupling approaches and mathematical configurations, an additional evaluation of their computational requirements is conducted to determine the most suitable choice for studying high-resolution SW-GW interactions.

### 3.4.3. Computational Efficiency

Both `interFoam` and `darcyInterTransportFoam` cases were run in parallel using 10 CPUs on 1 node of the local High-Performance Computing (HPC) cluster (2.9 GHz, Ubuntu 22.04.1 LTS operating system) at the Centre for Hydrogeology and Geothermics (CHYN) of the University of Neuchâtel. Accordingly, the surface and subsurface meshes of both model cases were manually decomposed into 10 subdomains. This manual decomposition ensured an equal division of the two domains (~6500 and ~10500 cells for each surface and subsurface subdomain, respectively).

Using the aforementioned number of processors, `darcyInterTransportFoam` takes about 74 hours to reach a steady-state solution and, thus, finish the simulation, while `interFoam` requires approximately 92 hours to fully complete its run. This difference in runtime highlights the higher computational demands of the FIM, a behaviour expected due to the use of the Navier-Stokes equations to solve the groundwater flow. Such computational requirements will substantially increase in larger domains than that studied here. Therefore, since both models have proven to perform similarly far from the interface (Section 3.4.1), `darcyInterTransportFoam` will always be the more efficient choice in cases where the effect of the interface is negligible and/or irrelevant.

Despite the runtimes of the integral and coupled approaches differing by around 18 hours, the need for significant computational resources to ensure an efficient computation remains the same for both models. While standard computers are rarely equipped with more than 16 cores, massively parallel machines (e.g., HPC supercomputers or multi-core processors) normally consist of hundreds or thousands of CPUs and random-access memory (RAM) banks. Consequently, the latter provide the necessary resources to efficiently run any CFD model, regardless of its scale or resolution, and, thus, are well-suited for both `interFoam` and `darcyInterTransportFoam` (Chapter 2).

## 3.5. Summary and Conclusions

The computational efficiency and capability of `darcyInterTransportFoam` and `interFoam` in solving SW-GW interactions were evaluated in this study, along with their

computational demands, by comparing the flow and solute dynamics simulated by each model across a 3D, synthetic rippled riverbed.

The results illustrate an overall similarity between the two modelling approaches, especially in the regions far from the SW-GW interface. Near the riverbed, discrepancies arise due to the distinct interface definitions inherent to each model's coupling approach. This variation in interface treatment results in significantly different velocity fields around this shared boundary, affecting both the direction and location of flux exchanges between the surface and subsurface domains. Nevertheless, these differences are confined to the first few centimetres above and below the interface, with minimal impact on the magnitude of the total SW-GW exchanges. Furthermore, minor discrepancies appear in the subsurface domain due to the different mathematical configuration of each model. Specifically, the combination of the Navier-Stokes equations with the Darcy-Forchheimer law in `interFoam` produces higher groundwater velocities than Darcy's law in the FCM, leading to slightly higher hydraulic heads and faster solute travel times. Despite these latter discrepancies, the hydraulic head and solute concentration fields converge to very similar solutions in both models when reaching steady-state, even in the region near the sediment-water interface. As for the computational requirements, the use of the Navier-Stokes equations in the subsurface domain causes `interFoam` to require about 18 hours more of runtime than `darcyInterTransportFoam` to finish the simulation, highlighting its higher computational demands.

The absence of field observations in the present study prevents verification of which model provides simulated fields closer to reality. Consequently, additional analysis incorporating field data will still be needed to address in detail the differing dynamics observed at the SW-GW interface. Moreover, a further comparison of both models under transient conditions will be required to fully assess the strengths and limitations of each approach for studying complex SW-GW interactions. However, based on obtained outcomes and computational demands, the conducted analysis has implications for the choice of fully-integrated vs. fully-coupled codes. When deciding between `darcyInterTransportFoam` and `interFoam`, the choice will largely depend on the spatial scale of interest and the processes under consideration. `darcyInterTransportFoam` offers significant computational advantages and is well-suited for simulations focusing on larger-scale hydrological behaviour, as both models produce consistent results at the larger scale far from the sediment-water interface. However, in regions close to the stream, particularly within riverbed sediments, where biogeochemical reactions are concentrated, notable differences between the model predictions occur. In these cases, `interFoam` should be considered as a viable alternative to `darcyInterTransportFoam`, because conceptually, it provides a more robust description of the fine-scale hydrodynamics and mass transport processes that control reaction rates. Therefore, for questions focused on riverbed biogeochemistry or small-scale heterogeneity, `interFoam` is essential despite its higher computational cost.

For broader-scale assessments or exploratory simulations, the faster **darcyInterTransportFoam** solver is sufficient and more efficient.

Additionally, this comparative study further tests the efficacy of **darcyInterTransportFoam** in simulating high-resolution processes across the SW-GW interface, while also demonstrating the capability of **interFoam** to model these processes at scales beyond the local level.

# References


[1] Barthel, R., & Banzhaf, S. (2016). Groundwater and surface water interaction at the regional-scale – A review with focus on regional integrated models. *Water Resources Management, 30*(1), 1-32. https://doi.org/10.1007/s11269-015-1163-z

[2] Berberović, E., van Hinsberg, N. P., Jakirlić, S., Roisman, I. V., & Tropea, C. (2009). Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution. *Physical Review E, 79*(3), 036306. https://doi.org/10.1103/PhysRevE.79.036306

[3] Boano, F., Harvey, J. W., Marion, A., Packman, A. I., Revelli, R., Ridolfi, L., & Wörman, A. (2014). Hyporheic flow and transport processes: Mechanisms, models, and biogeochemical implications. *Reviews of Geophysics, 41*, 603-679. https://doi.org/10.1002/2012RG000417

[4] Broecker, T., Elsesser, W., Teuber, K., Özgen, I., Nützmann, G., & Hinkelmann, R. (2017). High-resolution simulation of free-surface flow and tracer retention over streambeds with ripples. *Limnologica, 68*, 46-58. https://doi.org/10.1016/j.limno.2017.06.005

[5] Broecker, T., Gollo, V. S., Fox, A., Lewandowski, J., Nützmann, G., Arnon, S., & Hinkelmann, R. (2021). High-Resolution Integrated Transport Model for Studying Surface Water-Groundwater Interaction. *Groundwater, 59*(4), 488-502. https://doi.org/10.1111/gwat.13071

[6] Broecker, T., Teuber, K., Sobhi Gollo, V., Nützmann, G., Lewandowski, J., & Hinkelmann, R. (2019). Integral Flow Modelling Approach for Surface Water-Groundwater Interactions along a Rippled Streambed. *Water, 11*(7), 1517. https://doi.org/10.3390/w11071517

[7] Brunner, P., & Simmons, C. T. (2012). HydroGeoSphere: A Fully Integrated, Physically Based Hydrological Model. *Groundwater, 50*(2), 170-176. https://doi.org/10.1111/j.1745-6584.2011.00882.x

[8] Brunner, P., Simmons, C. T., Cook, P. G., & Therrien, R. (2010). Modeling Surface Water-Groundwater Interaction with MODFLOW: Some Considerations. *Groundwater, 48*(2), 174-180. https://doi.org/10.1111/j.1745-6584.2009.00644.x

[9] Brunner, P., Therrien, R., Renard, P., Simmons, C. T., & Hendricks Franssen, H.-J. (2017). Advances in understanding river-groundwater interactions. *Reviews of Geophysics, 55*(3), 818-854. https://doi.org/10.1002/2017RG000556

[10] Chen, X., M. Bayani, C., & Chen, L. (2015). Three-dimensional versus two-dimensional bed form-induced hyporheic exchange. *Water Resources Research, 51*(4), 2923-2936. https://doi.org/10.1002/2014WR016848

[11] Chen, X., M. Bayani, C., & Chen, L. (2018). Hyporheic Exchange Driven by Three-Dimensional Sandy Bed Forms: Sensitivity to and Prediction from Bed Form Geometry. *Water Resources Research, 54*(6), 4131-4149. https://doi.org/10.1029/2018WR022663

[12] Chow, R., Bennet, J., Dugge, J., Wöhling, T., & Nowak, W. (2020). Evaluating Subsurface Parameterization to Simulate Hyporheic Exchange: The Steinlach River Test Site. *Groundwater, 58*(1), 93-109. https://doi.org/10.1111/gwat.12884

[13] Chow, R., Wu, H., Bennet, J. P., Dugge, J., Wöhling, T., & Nowak, W. (2019). Sensitivity of Simulated Hyporheic Exchange to River Bathymetry: The Steinlach River Test Site. *Groundwater, 57*(3), 378-391. https://doi.org/10.1111/gwat.12816

[14] Deshpande, S. S., Anumolu, L., & Trujillo, M. F. (2012). Evaluating the performance of the two-phase flow solver interFoam. *Computational Science & Discovery, 5*(1), 014016. https://doi.org/10.1088/1749-4699/5/1/014016

[15] Dichgans, F., Boos, J.-P., Ahmadi, P., Frei, S., & Fleckenstein, J. H. (2023). Integrated numerical modeling to quantify transport and fate of microplastics in the hyporheic zone. *Water Research, 243*, 120349. https://doi.org/10.1016/j.watres.2023.120349

[16] Fleckenstein, J. H., Krause, S., Hannah, D. M., & Boano, F. (2010). Groundwater-surface water interactions: New methods and models to improve understanding of processes and dynamics. *Advances in Water Resources, 33*, 1291-1295. https://doi.org/10.1016/j.advwatres.2010.09.011

[17] Haitjema, H. M., & Anderson, M. P. (2016). Darcy Velocity Is Not a Velocity. *Groundwater, 54*(1), 1-1. https://doi.org/10.1111/gwat.12386

[18] Hirt, C. W., & Nichols, B. D. (1981). Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics, 39*(1), 201-225. https://doi.org/10.1016/0021-9991(81)90145-5

[19] Janssen, F., Cardenas, M. B., Sawyer, A. H., Dammrich, T., Krietsch, J., & de Beer, D. (2012). A comparative experimental and multiphysics computational fluid

dynamics study of coupled surface-subsurface flow in bed forms. *Water Resources Research, 48*(8), W08514. https://doi.org/10.1029/2012WR011982

[20] Kollet, S. J., & Maxwell, R. M. (2006). Integrated surface–groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model. *Advances in Water Resources, 29*(7), 945-958. https://doi.org/10.1016/j.advwatres.2005.08.006

[21] Kollet, S. J., & Maxwell, R. M. (2008). Capturing the influence of groundwater dynamics on land surface processes using an integrated, distributed watershed model. *Water Resources Research, 44*(2). https://doi.org/10.1029/2007WR006004

[22] Krause, S., Hannah, D., & Grimm, N. (Eds.). (2023). *Ecohydrological Interfaces* (1st ed.). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119489702

[23] Kuffour, B. N., Engdahl, N. B., Woodward, C. S., Condon, L. E., Kollet, S., & Maxwell, R. M. (2020). Simulating coupled surface–subsurface flows with ParFlow v3.5.0: capabilities, applications, and ongoing development of an open-source, massively parallel, integrated hydrologic model. *Geoscientific Model Development, 13*(3), 1373–1397. https://doi.org/10.5194/gmd-13-1373-2020

[24] Langevin, C. D., Hughes, J. D., Provost, A. M., Russcher, M. J., & Panday, S. (2023). MODFLOW as a Configurable Multi-Model Hydrologic Simulator. *Groundwater, 62*(1), 111-123. https://doi.org/10.1111/gwat.13351

[25] Lee, A., Aubeneau, A., & Cardenas, M. (2020). The Sensitivity of Hyporheic Exchange to Fractal Properties of Riverbeds. *Water Resources Research, 56*(5), e2019WR026560. https://doi.org/10.1029/2019WR026560

[26] Lee, A., Aubeneau, A., Cardenas, M., & Liu, X. (2022). Hyporheic Exchange Due to Cobbles on Sandy Beds. *Water Resources Research, 58*(1), e2021WR030164. https://doi.org/10.1029/2021WR030164

[27] Lee, A., Aubeneau, A., Liu, X., & Cardenas, M. B. (2021). Hyporheic Exchange in Sand Dunes Under a Freely Deforming River Water Surface. *Water Resources Research, 57*(3), e2020WR028817. https://doi.org/10.1029/2020WR028817

[28] Lewandowski, J., Arnon, S., Banks, E., Batelaan, O., Betterle, A., Broecker, T., Coll, C., Drummond, J. D., Gaona Garcia, J., Galloway, J., Gomez-Velez, J., Grabowski, R. C., Herzog, S. P., Hinkelmann, R., Höhne, A., Hollender, J., Horn, M. A., Jaeger, A., Krause, S., Löchner Prats, A., Magliozzi, C., Meinikmann, K., Mojarrad, B. B., Mueller, B. M., Peralta-Maraver, I., Popp, A. L., Posselt, M., Putschew, A., Radke, M., Raza, M., Riml, J., Robertson, A., Rutere, C., Schaper, J. L., Schirmer, M., Schulz, H., Shanafield, M., Singh, T., Ward, A. S., Wolke, P., Wörman, A., & Wu, L. (2019). Is the Hyporheic Zone Relevant beyond the Scientific Community? *Water, 11*(11), 2230. https://doi.org/10.3390/w11112230

[29] Lewandowski, J., Meinikmann, K., & Krause, S. (2020). Groundwater-Surface Water Interactions: Recent Advances and Interdisciplinary Challenges. *Water, 12*(1), 296. https://doi.org/10.3390/w12010296

[30] Li, B., Liu, X., Kaufman, M. H., Turetcaia, A., Chen, X., & Cardenas, B. M. (2020). Flexible and Modular Simultaneous Modeling of Flow and Reactive Transport in Rivers and Hyporheic Zones. *Water Resources Research, 56*(2), e2019WR026528. https://doi.org/10.1029/2019WR026528

[31] Marquez-Damian, S. (2012). *Description and utilization of interFoam multiphase solver.* Buenos Aires: CIMEC-INTEC. Retrieved from https://www.cfdyna.com/Home/OpenFOAM/of_Tut_Web/of_Suites_Description.pdf

[32] Maxwell, R. M., Putti, M., Meyerhoff, S., Delfs, J.-O., Ferguson, I. M., Ivanov, V., Kim, J., Kolditz, O., Kollet, S. J., Kumar, M., Lopez, S., Niu, J., Paniconi, C., Park, Y.-J., Phanikumar, M. S., Shen, C., Sudicky, E. A., & Sulis, M. (2014). Surface-subsurface model intercomparison: A first set of benchmark results to diagnose integrated hydrology and feedbacks. *Water Resources Research*, 50(2), 1531-1549. https://doi.org/10.1002/2013WR013725

[33] Menter, F. R. (1994). Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal, 32*(8), 1598-1605. https://doi.org/10.2514/3.12149

[34] OpenCFD Ltd. (2023, December). *OpenFOAM® v2312 User Guide*. Retrieved December 25, 2024, from https://doc.openfoam.com/2312/

[35] OpenCFD Ltd. (2024, June). OpenFOAM® v2406. OpenFOAM Download [code]. Retrieved from https://dl.openfoam.com/source/v2406/

[36] Oxtoby, O., Heyns, J., & Suliman, R. (2013). A finite-volume solver for two-fluid flow in heterogeneous porous media based on OpenFOAM. Hamburg: 7th Open Source CFD International Conference. https://doi.org/10.13140/2.1.3075.8400

[37] Paniconi, C., & Putti, M. (2015). Physically based modeling in catchment hydrology at 50: Survey and outlook. *Water Resources Research, 51*(9), 7090-7129. https://doi.org/10.1002/2015WR017780

[38] Rusche, H. (2003). *Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions.* PhD Thesis, Imperial College of Science, Technology and Medicine, London, UK.

[39] Schilling, O. S., Cook, P. G., & Brunner, P. (2019). Beyond Classical Observations in Hydrogeology: The Advantages of Including Exchange Flux, Temperature, Tracer Concentration, Residence Time, and Soil Moisture Observations in Groundwater Model Calibration. *Reviews of Geophysics, 57*(1), 146-182. https://doi.org/10.1029/2018RG000619

[40] Schulze, L., & Thorenz, C. (2014). The Multiphase Capabilities of the CFD Toolbox OpenFOAM for hydraulic engineering applications. En R. Lehfeldt, & R. Kopmann (Ed.). (pp. 1007-1016). Hamburg: Proceedings of the 11th International Conference on Hydroscience & Engineering. https://doi.org/20.500.11970/99530

[41] Schulze, L., & Thorenz, C. (2015). *Mehrphasenmodellierung im Wasserbau.* Karlsruhe: Bundesanstalt für Wasserbau (Hg.): Wasserbauwerke - Vom hydraulischen Entwurf bis zum Betrieb. https://doi.org/20.500.11970/102274

[42] Sobhi Gollo, V., Broecker, T., Marx, C., Lewandowski, J., Nützmann, G., & Hinkelmann, R. (2022). A comparative study of integral and coupled approaches formodeling hydraulic exchange processes across a rippled streambed. *International Journal on Geomathematics, 13*(16), 1-27. https://doi.org/10.1007/s13137-022-00206-5

[43] Therrien, R., McLaren, R., Sudicky, E., & Panday, S. (2010). *HydroGeoSphere: A Three-Dimensional Numerical Model Describing Fully-Integrated Subsurface and Surface Flow and Solute Transport.* Groundwater Simulations Group. Waterloo: University of Waterloo.

[44] Trauth, N., & Fleckenstein, J. H. (2017). Single discharge events increase reactive efficiency of the hyporheic zone. *Water Resources Research, 53*(1), 779-798. https://doi.org/10.1002/2016WR019488

[45] Trauth, N., Schmidt, C., Vieweg, M., Maier, U., & Fleckenstein, J. H. (2014). Hyporheic transport and biogeochemical reactions in pool-riffle systems under varying ambient groundwater flow conditions. *Journal of Geophysical Research: Biogeosciences, 119*(5), 910-928. https://doi.org/10.1002/2013JG002586

[46] Trauth, N., Schmidt, C., Vieweg, M., Oswald, S. E., & Fleckenstein, J. H. (2015). Hydraulic controls of in-stream gravel bar hyporheic exchange and reactions. *Water Resources Research, 51*(4), 2243-2263. https://doi.org/10.1002/2014WR015857

[47] Ubbink, O. (1997). *Numerical prediction of two fluid systems with sharp interfaces.* PhD Thesis, Imperial College of Science, Technology & Medicine, Department of Mechanical Engineering, London, UK.

[48] Weller, H. G. (2002). *Derivation, modelling and solution of the conditionally averaged two-phase flow equations.* Boston, USA: Nabla Ltd.

[49] Weller, H. G., Tabor, G., Jasak, H., & Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics, 12*(6), 620-631. https://doi.org/10.1063/1.168744

[50] Whitaker, S. (1996). The Forchheimer equation: A theoretical development. *Transport in Porous Media, 25*, 27-61. https://doi.org/10.1007/BF00141261

# 4

# Code Developments: Solver Upgrades and OpenFOAM Dictionary Generation Scripts

# 4.1. Introduction

This chapter provides a detailed documentation on the modifications made to the original `hyporheicScalarInterFoam` solver (Lee, et al., 2021), resulting in its updated version, `darcyInterTransportFoam` (Pardo-Álvarez, 2025), previously described in Chapter 2. The code enhancements include both internal solver updates and external applications. The information presented here is crucial for future users and developers, as it transparently outlines the key technical changes required to address the limitations described in Chapter 1. By bridging the gap between the underlying conceptual framework and its realization, the present chapter ensures a clear understanding of the model's capabilities and limitations.

Additionally, this chapter includes supporting information on the **MATLAB** scripts (The MathWorks Inc., 2024) developed to generate all the mesh-related dictionaries (e.g., *blockMeshDict*, *topoSetDict*) and input data files (e.g., *setFieldsDict*) needed to set up an **OpenFOAM** simulation case.

The detailed description of the solver updates and new add-ons of `darcyInterTransportFoam` is provided in Section 4.2, while a brief overview of the functionalities of the new script suite and its associated output files is offered in Section 4.3.

# 4.2. darcyInterTransportFoam v1.0: Solver Updates and New Add-Ons

## 4.2.1. Code Modifications

Numerous changes were implemented in the code of the model as well as in the structure and organization of both files and directories. Such modifications aim not only to simplify the code but also to make it more accessible for other users. The applied actions are enumerated as follows:

1. Code update to the latest **OpenFOAM** version, v2412, released in December 2024.
2. File reordering and simplification. Only the model-specific files are now contained in the solver directory. The rest are added through the Make/options file, where the paths to the standard-released solver directories are included within the applications section (`EXE_INC`).
3. Removal of unnecessary and unused files and pieces of code (i.e., variables, fields and instructions), as well as code reordering to improve readability.

4. Renaming of files, fields and variables to make them simpler and more intuitive for the user.
5. Definition of the necessary variables, fields and dictionaries for the newly implemented modelling options (described below).

### 4.2.2. Upgraded Modelling Features

The simulation capabilities of the prior solver version, **hyporheicScalarInterFoam**, are upgraded in **darcyInterTransportFoam** through new modelling features that either improve previously existing functionalities or provide additional simulation options. The complete description of each model novelty is provided below, including the names of all the cited parameters (i.e., files, variables, fields, dictionaries, etc.) to facilitate their identification in the code snippets provided:

1. As opposed to the original model, the subsurface flow solution is computed based on the hydraulic head field (`hSub`) rather than the modified pressure field (`pTildeSub`). This fact not only makes the code more intuitive for any user but also facilitates its coupling to any other groundwater model.

**File: subsurface\darcyFoam.H**

```
28:     fvScalarMatrix hSubEqn
29:     (
30:         fvm::laplacian(K,hSub) == fvc::div(phiUgradP)
31:     );
```

Furthermore, the model calculates the piezometric level and hydraulic head fields (`piezoSur`, `hSur`) in the surface domain, the specific discharge field (`q`) in the subsurface domain and the water depth fields (`dSur`, `dSub`) in both domains.

**File: surface\interFoam.H**

```
21: // Water depth (L)
22: dSur == p/(rho*mag(g));
23:
24: // Piezometric level (L)
25: piezoSur == dSur + mesh.C().component(vector::Z);
26:
27: // Hydraulic head (L)
28: hSur == (0.5*magSqr(U)/mag(g)) + piezoSur;
```

**File: subsurface\darcyFoam.H**

```
42:         // Specific discharge (L/T)
43:         q = UgradP - (K&fvc::grad(hSub));

    ...

57: // Static pressure (pressure head) (M/L*T^-2)
58: pSub == rho1*mag(g)*(hSub-meshSub.C().component(vector::Z));
59:
60: // Water depth (L)
```

```
61: dSub == pSub/(rho1*mag(g));
```

A comprehensive description of the equations corresponding to all the aforementioned hydraulic fields can be found in Sections 2.2.1.1.1 and 2.2.1.2.1.

2. The `dSur`, `piezoSur` and `hSur` fields can be optionally clipped every timestep (`alphaAirClip`) according to an $\alpha_i$ threshold (`alphaWaterThres`) defined in the *transportProperties* dictionary of the surface domain. As a result of the clipping, the three hydraulic variables turn zero at those cells where $\alpha_i$ is below the specified threshold. This action facilities the visualization of the computed water phase as well as its exportation to another flow model.

**File: surface\interFoam.H**

```
45: if (alphaAirClip)
46: {
47:     dSur *= pos0(alpha1 - alphaWaterThres);
48:     piezoSur *= pos0(alpha1 - alphaWaterThres);
49:     hSur *= pos0(alpha1 - alphaWaterThres);
50: }
```

3. Analogously to other multiphase **OpenFOAM** solvers, **`darcyInterTransportFoam`** allows to define a free-surface reference (`hRef`) in the surface domain. This feature was unable in the original model, which prevented the user to set a (subcritical) water height at the surface outlet patch.

**File: src\finiteVolume\cfdTools\general\include\gh.H**

```
01:     Info<< "Calculating field g.h\n" << endl;
02:     dimensionedScalar ghRef
03:     (
04:         mag(g.value()) > SMALL
05:       ? g & (cmptMag(g.value())/mag(g.value()))*hRef
06:       : dimensionedScalar("ghRef", g.dimensions()*dimLength, 0)
07:     );
08:     const int oldLocal = volVectorField::Boundary...
            ::localConsistency;
09:     volVectorField::Boundary::localConsistency = 0;
10:
11:     volScalarField gh("gh", (g & mesh.C()) - ghRef);
12:     surfaceScalarField ghf("ghf", (g & mesh.Cf()) - ghRef);
13:
14:     volVectorField::Boundary::localConsistency = oldLocal;
```

The surface water height is commonly combined with a Dirichlet hydrostatic pressure BC and a von Neumann BC for velocity to model the free exist of the flow from the simulation domain. This approach not only avoids potential stability issues but also prevents unrealistic hydrodynamic scenarios, such as the decrease of the free-surface towards the outlet of the surface domain (supercritical flow).

4. The adjustable body force, $gradP_i$ (`gradP`), required to make $p_{sur,i}$ compatible with periodic boundary conditions (referred to as `cyclic` in **OpenFOAM**) is now considered in the computation of all the subsurface hydraulic variables. $gradP_i$

enables to adjust the surface volume-averaged streamwise velocity field of the momentum equation, $U_{sur,i}$ (`magUbarAve`), to a user-defined mean flow velocity, $\overline{U}$ (`Ubar`), equivalent to an enforcing discharge. Its value is calculated through an external function object (**`VoFmeanVelocityForce`**) at every timestep as:

$$gradP_i = \lambda \frac{\left(|\overline{U}| - |U_{sur,i}|\right)}{a_{p_i}} \tag{4.1}$$

where $\lambda$ (`relaxation`) is a relaxation factor and $a_{p_i}$ (`rAUave`) are the diagonal elements of the linear system of equations of $U_{sur,i}$.

**File: VoFmeanVelocityForce.C**

```
216:     // Calculate the pressure gradient increment needed to adjust
217:     // the average flow-rate to the desired value
218:     dGradP_ = relaxation_*(mag(Ubar_) - magUbarAve)/rAUave;
```

In the original model, **`hyporheicScalarInterFoam`**, the pressure gradient $gradP_i$ was prescribed to Darcy's law (`q`) when required by periodicity but left it aside in the hydraulic head (`hSub`) and groundwater flux (`phi_q`) calculation. **`darcyInterTransportFoam`** fills this computational gap by additionally adjusting `hSub` (model novelty 1) and `phi_q` via the velocity flux resulting from $gradP_i$.

**File: subsurface\darcyFoam.H**

```
19: // Predicted velocity from gradP, not the pressure distribution yet
20: UgradP == gradPMulti*(K&gradP/(rho1*mag(g)));
21:
22: // Flow flux due to the predicted velocity
23: phiUgradP = fvc::flux(UgradP);

    ...

37:         // Darcy flux (returned from hSubEqn.flux())
38:         phi_q = phiUgradP - hSubEqn.flux();
```

5. The original model defined the hydraulic conductivity (`K`), the porosity (`theta`) as well as the longitudinal and transverse dispersivity (`alpha_L` and `alpha_T`) fields through single values specified in the *transportProperties* dictionary of the subsurface domain. Furthermore, the model did not require a `K` value but calculated it via the intrinsic permeability of the porous medium (`kSub`) and the dynamic viscosity of the water (`muSub`). Consequently, all the abovementioned subsurface fields could be simply defined as homogeneous and isotropic. This fact prevented the user to, e.g., distinguishing different `theta` and `K` zones in the aquifer or from establishing a heterogeneous distribution of both parameters throughout the domain. In **`darcyInterTransportFoam`**, however, `K` is directly specified as a tensor field and `theta`, `alpha_L` and `alpha_T` as scalar fields.

**File: subsurface\createSubFields.H**

```
20: // Hydraulic conductivity (L/T)
    ...
22: volTensorField K
23: (
24:     IOobject
25:     (
26:         "K",
27:         runTime.timeName(),
28:         meshSub,
29:         IOobject::MUST_READ,
30:         IOobject::AUTO_WRITE
31:     ),
32:     meshSub
33: );

    ...

39: // Porosity (-)
    ...
41: volScalarField theta
42: (
43:     IOobject
44:     (
45:         "theta",
46:         runTime.timeName(),
47:         meshSub,
48:         IOobject::READ_IF_PRESENT,
49:         IOobject::AUTO_WRITE
50:     ),
51:     meshSub,
52:     dimensionedScalar(dimless,1)
53: );

    ...

495: // Longitudinal dispersivity (L)
    ...
497: volScalarField alpha_L
498: (
499:     IOobject
500:     (
501:         "alpha_L",
            ...
506:     ),
507:     meshSub,
508:     dimensionedScalar(dimLength,Zero)
509: );

    ...

515: // Transverse dispersivity (L)
    ...
517: volScalarField alpha_T
518: (
519:     IOobject
520:     (
521:         "alpha_T",
            ...
526:     ),
527:     0.1*alpha_L
528: );
```

This new setup provides full flexibility for the distribution of the previous parameters in the subsurface domain (beyond the impervious zone derived from the surface dry areas – see next point). The definition of the non-uniform fields can be performed following different approaches. These include, e.g., the field data assignment before the start of the simulation or the use of *setFieldsDict* through its multiple geometric selection options.

6. **hyporheicScalarInterFoam** could not handle situations where areas of the surface bottom patch were partially dry. This circumstance no longer represents a limitation in **darcyInterTransportFoam**. A newly implemented approach allows the automatic specification of impermeable regions in the subsurface domain according to the surface dry areas. This novel methodology comprises two steps: first, the dry faces at the surface bottom patch are selected (`dryBottomSur`) every timestep based on `alphaWaterThres` (*transportProperties* dictionary of the surface domain).

**File: surface\interFoam.H**

```
32: // Dry faces at the bottom patch of the surface domain (-)
33: forAll(dryBottomSur,i)
34: {
35:     dryBottomSur[i] = dryBottomSur_orig[i];
36:
37:     if (dryBottomSur[i] != -1)
38:     {
39:         dryBottomSur[i] = neg(alpha1[i] - alphaWaterThres);
40:     }
41: }
```

Next, the selected faces are optionally used (`imperRegion`) to define impervious regions in the subsurface domain, whose shape and extension correspond to the surface areas previously chosen. The depth (`d_imper`, must be always negative), the geological properties (`K_imper`, `theta_imper`, `alpha_L_imper` and `alpha_T_imper`) as well as the solute concentration (`CSub_imper`) and water temperature (`TSub_imper`) of such regions are supplied within the *imperRegionProperties* dictionary (*constant* directory of the subsurface domain). Moreover, the user can optionally set a time limit for the update of the specified impermeable zones via `endTimeUpdateRegion` in the same dictionary.

**File: subsurface\impermeableRegion.H**

```
01: if (imperRegion and runTime.timeName() <= endTimeUpdateRegion)
02: {
03:     forAll(cellCtr_meshSub,i)
04:     {

            ...

49:         if (dryBottomSur.boundaryField()...
                [surfaceBottomPatchID][minPointi] == 1.0 and ...
50:             meshSub.cellCentres()[i].component(vector::Z)...
```

```
                    > mesh.C().boundaryField()[surfaceBottomPatchID]...
                  [minPointi].component(vector::Z)+d_imper.value())
51:         {
52:             if (imperProperty == "K" or imperProperty == "all")
53:             {
54:                 K[i] = K_imper.value()*tensor(I);
55:             }
56:
57:             if (imperProperty == "theta" or imperProperty == "all")
58:             {
59:                 theta[i] = theta_imper.value();
60:             }
61:
62:             if (soluteTransport or soluteTransportSub)
63:             {
64:                 CSub[i] = CSub_imper.value();
65:
66:                 if (imperProperty == "alpha_L" or ...
                        imperProperty == "all")
67:                 {
68:                     alpha_L[i] = alpha_L_imper.value();
69:                 }
70:
71:                 if (imperProperty == "alpha_T" or ...
                        imperProperty == "all")
72:                 {
73:                     alpha_T[i] = alpha_T_imper.value();
74:                 }
75:             }
76:
77:             if (heatTransfer or heatTransferSub)
78:             {
79:                 TSub[i] = TSub_imper.value();
80:             }
81:         }
82:     }

        ...

90: }
```

The use of this new functionality is limited to quasi-steady-state scenarios in which the dry areas of the surface bottom patch remain constant throughout the entire simulation. Applying it to transient scenarios would result in unrealistic behaviour, as steep hydraulic gradients would be generated at the newly permeable cells each timestep. As an alternative, the user can manually define as many steady, impermeable regions as desired taking advantage of the model novelty number 5.

7. The new model modifies the process selection scheme implemented in **`hyporheicScalarInterFoam`**, which required to first choose the domain/s to be simulated (i.e., surface and/or subsurface) and next the process/es to be solved (i.e., flow, solute transport and/or heat transfer). Additionally, the solute transport simulation had to be separately selected for each domain. In **`darcyInterTransportFoam`**, the choice of the flow, the solute transport as well as the heat transfer simulation can be made jointly for both domains in the *simulationProperties* dictionary (`flow`, `soluteTransport` and `heatTransfer`

options, respectively). As an alternative to the joint selection, the simulation of each process can also be specified for a single domain in its corresponding *transportProperties* dictionary (*constant* directory). In this latter case, the simulation options would be `flowSur`, `soluteTransportSur` and `heatTransferSur` for the surface domain and `flowSub`, `soluteTransportSub` and `heatTransferSub` for the subsurface domain.

**File: darcyInterTransportFoam.C**

```
175:                 if (flow or flowSur)
176:                 {
177:                     // Solve the surface flow equation
178:                     #include "interFoam.H"
179:                 }
180:
181:                 if (heatTransfer or heatTransferSur)
182:                 {
183:                     // Solve the surface heat transfer equation
184:                     #include "TSurEqn.H"
185:                 }
186:
187:                 if (soluteTransport or soluteTransportSur)
188:                 {
189:                     // Solve the surface solute transport equation
190:                     #include "CSurEqn.H"
191:                 }

                ...

204:             if (flow or flowSub)
205:             {
206:                 // Solve the subsurface flow equation
207:                 #include "darcyFoam.H"
208:             }
209:
210:             if (heatTransfer or heatTransferSub)
211:             {
212:                 // Solve the subsurface heat transfer equation
213:                 #include "TSubEqn.H"
214:             }
215:
216:             if (soluteTransport or soluteTransportSub)
217:             {
218:                 // Solve the subsurface solute transport equation
219:                 #include "CSubEqn.H"
220:             }
```

The entire selection suite and its corresponding implications can be checked in Figure 2.1.

8. The solute transport module has been completely reformulated in the new model. The implemented changes affect both the code organization and the modelling approach. Regarding the former, the diffusion-dispersion coefficient expressions as well as the solute transport governing equations are now placed within one single header file (*.H) in each domain. These new *.H files for the surface and subsurface domains are respectively named as *CSurEqn.H* and *CSubEqn.H*. As

for the computational approach, both surface diffusion and subsurface hydrodynamic dispersion fields (`DSur` and `DSub`, respectively) are calculated in a different way than in **`hyporheicScalarInterFoam`**. `DSur` is computed through a new expression, whereas `DSub` results from a partial modification of the original equation. Additionally, both `DSur` and `DSub` can be alternatively specified as uniform and constant over the entire domain through a single value (`constDSur` or `constDSub`).

**File: surface\CSurEqn.H**

```
03: // Surface diffusion coefficient =
04: if (mixture.readIfPresent("constDSur",constDSur))
05: {
06:     // = constant surface diffusion coefficient
07:     DSur == constDSur;
08: }
09: else
10: {
11:     // = physical diffusivity
12:     DSur == molDSur;
13:
14:     if (turbModelType != "laminar")
15:     {
16:         // = physical diffusivity + turbulent diffusivity
17:         DSur += turbulence->nut()/Sct;
18:     }
19: }
```

**File: subsurface\CSubEqn.H**

```
03: // Hydrodynamic dispersion coefficient =
04: if (transportProperties.readIfPresent("constDSub",constDSub))
05: {
06:     // = constant hydrodynamic dispersion coefficient
07:     DSub == constDSub*tensor::one;
08: }
09: else
10: {
11:     // = effective molecular diffusion + mechanical dispersion
12:     DSub == (molDSub/(1-2*log(theta)))*tensor::one...
            + alpha_T*mag(USub)*tensor(I) + (alpha_L-alpha_T)...
            * ((USub*USub)/(mag(USub)+USmall));
13: }
```

Furthermore, **`darcyInterTransportFoam`** provides a new simulation option to solve the conservative solute transport (`CSur`) in just one phase at the surface domain (`solutePhases == "phase1"`). This novel feature is based on the advection-diffusion equation contained in the standard **`scalarTransport`** function object of **OpenFOAM**. Yet, the two-phase solute simulation remains as an option through the original computational approach (`solutePhases == "all"`).

**File: surface\CSurEqn.H**

```
27: if (solutePhases == "all")
28: {
```

```
        ...

55: }
56: else if (solutePhases == "phase1")
57: {
58:     DSur *= pos0(alpha1 - 0.99);
59:
60:     // Reset D dimensions consistent with alphaPhiUn
61:     DSur.dimensions().reset(alphaPhiUn.dimensions()/dimLength);
62:
63:     for (label i = 0; i <= nCorrCSur; i++)
64:     {
65:         fvScalarMatrix CSurEqn
66:         (
67:             fvm::ddt(CSur)
68:           + fvm::div(alphaPhiUn, CSur, "div(phi,C)")
69:           - fvm::laplacian(DSur, CSur, "laplacian(D,C)")
70:           ==
71:             fvOptions(alpha1, CSur)
72:         );
73:
74:         CSurEqn.relax();
75:         fvOptions.constrain(CSurEqn);
76:         CSurEqn.solve();
77:     }

        ...

81: }
```

All the modelling options and parameters required to conduct the solute transport simulation have to be specified in the *transportProperties* dictionary of each domain. Further details of the newly implemented equations can be found in Sections 2.2.1.1.2 and 2.2.1.2.2.

9. The simulation of the temperature distribution in both domains is now available thanks to the heat transfer module implemented in **darcyInterTransportFoam**. The code organization as well as the computational scheme of this new module are very similar to those applied to the solute transport module. In this way, all the equations required to conduct the heat transfer simulation are included within the *TSurEqn.H* and *TSubEqn.H* files, each belonging to one domain. The latter contains the partial differential equation required to compute the temperature distribution in the ground (`TSub`), whereas the former comprises the same phase-based simulation options (`heatPhases`) available for solute transport to solve the transfer of heat in the surface domain (`TSur`).

**File: surface\TSurEqn.H**

```
07: if (heatPhases == "all")
08: {
09:     for (label i = 0; i <= nCorrTSur; i++)
10:     {
11:         fvScalarMatrix TSurEqn
12:         (
13:             fvm::ddt(rhoCp, TSur)
14:           + fvm::div(rhoCpPhi, TSur, "div(rhoCpPhi,T)")
```

```
15:             - fvm::laplacian(kappaSur, TSur, "laplacian(kappa,T)")
16:             ==
17:               fvOptions(rhoCp, TSur)
18:         );
19:
20:         TSurEqn.relax();
21:         fvOptions.constrain(TSurEqn);
22:         TSurEqn.solve();
23:     }
24: }
25: else if (heatPhases == "phase1")
26: {
27:     kappaSur *= pos0(alpha1 - 0.99);
28:
29:     for (label i = 0; i <= nCorrTSur; i++)
30:     {
31:         fvScalarMatrix TSurEqn
32:         (
33:             fvm::ddt(rho1*Cp1, TSur)
34:           + fvm::div(rho1*Cp1*alphaPhi10, TSur, "div(rhoCpPhi,T)")
35:           - fvm::laplacian(kappaSur, TSur, "laplacian(kappa,T)")
36:           ==
37:             fvOptions(rho1*Cp1*alpha1, TSur)
38:         );
39:
40:         TSurEqn.relax();
41:         fvOptions.constrain(TSurEqn);
42:         TSurEqn.solve();
43:     }
44: }
```

**File: subsurface\TSubEqn.H**

```
03: for (label i = 0; i <= nCorrTSub; i++)
04: {
05:     fvScalarMatrix TSubEqn
06:     (
07:         fvm::ddt(theta*rho_f*Cp_f + (1-theta)*rho_s*Cp_s, TSub)
08:       + fvm::div(rho_f*Cp_f*phi_USub, TSub, "div(rhoCpPhi,T)")
09:       - fvm::laplacian(theta*kappa_f + (1-theta)*kappa_s, TSub, ...
            "laplacian(kappa,T)")
10:       ==
11:         fvOptions(theta*rho_f*Cp_f + (1-theta)*rho_s*Cp_s, TSub)
12:     );
13:
14:     TSubEqn.relax();
15:     fvOptions.constrain(TSubEqn);
16:     TSubEqn.solve();
17: }
```

The necessary parameters and simulation options to run the new temperature module need to be defined in the *transportProperties* dictionary corresponding to each domain. A detailed description of the novel heat transfer equations is provided in Sections 2.2.1.1.3 and 2.2.1.2.3.

10. The sequential iterative coupling scheme (Section 2.2.2.2) has been modified in the new model to improve the transfer of information between the domains (Figure 2.2). In this way, the data is no longer interpolated but directly assigned to the matching nodes of both meshes at their respective interface patch.

**File: mappingSurToSub.H**

```
35:     // Selection of the matching nodes of both meshes at the...
            interface patch
36:     // for the suitable mapping of values between them
37:     forAll(faceCtr_subTop,i)
38:     {
39:         label minPointi = 0;
40:         scalar minDistSqr = magSqr(faceCtr_surBottom[minPointi]...
                - faceCtr_subTop[i]);
41:
42:         for (label j=1;j<faceCtr_surBottom.size();j++)
43:         {
44:             scalar distSqr = magSqr(faceCtr_surBottom[j] ...
                    - faceCtr_subTop[i]);
45:
46:             if (distSqr < minDistSqr)
47:             {
48:                 minDistSqr = distSqr;
49:                 minPointi = j;
50:             }
51:         }
52:
53:         // Assignment of values from the surface to the ...
                subsurface domain
54:         if (flow or (flowSur and flowSub))
55:         {
56:             hSub.boundaryFieldRef()[subSurfaceTopPatchID][i] = ...
                    hSur.boundaryField()...
                    [surfaceBottomPatchID][minPointi];
57:         }
58:
59:         if (heatTransfer or (heatTransferSur and heatTransferSub))
60:         {
61:             Ttmp_sub[i] = TSur.boundaryField()...
                    [surfaceBottomPatchID][minPointi];
62:         }
63:
64:         if (soluteTransport or (soluteTransportSur and ...
                soluteTransportSub))
65:         {
66:             Ctmp_sub[i] = CSur.boundaryField()...
                    [surfaceBottomPatchID][minPointi];
67:         }
68:     }
```

**File: mappingSubToSur.H**

```
41:     forAll(faceCtr_surBottom,i)
42:     {

             ...

58:         if (flow or (flowSur and flowSub))
59:         {
60:             U.boundaryFieldRef()[surfaceBottomPatchID][i] = ...
                    q.boundaryField()...
                    [subSurfaceTopPatchID][minPointi];
61:         }

            ...

72:     }
```

This new approach guarantees the exact conveyance of the solution from one domain to another (`hSur`, `q`, `CSur`/`CSub` and `TSur`/`TSub`), avoiding the potential loss of information resulting from the former path-to-patch interpolation. As a result, the two-way data mapping is performed in a more accurate way than in the original model, **`hyporheicScalarInterFoam`**.

11. The convergence check of both domain solutions is reformulated in the new model. Flow and solute convergences are now evaluated every timestep by means of flux mismatch at the interface patch of the two meshes, that is, the surface bottom and subsurface top boundaries.

**File: domainsConvergence.H**

```
01: // Flow convergence check
02: if (flow or (flowSur and flowSub))
03: {
04:     scalarField phiBottomSur = ...
            phi.boundaryField()[surfaceBottomPatchID];
05:     scalarField phiTopSub = ...
            phi_q.boundaryField()[subSurfaceTopPatchID];
06:
07:     scalarField couplePhiDiff = 0.*phiBottomSur;
08:
09:     forAll (couplePhiDiff,i)
10:     {
11:         couplePhiDiff[i] = phiBottomSur[i] - phiTopSub[i];
12:     }
13:
14:     couplePhiMaxMismatch = gMax(mag(couplePhiDiff));
15: }
16: else
17: {
18:     couplePhiMaxMismatch = 0.0;
19: }

    ...

31: // Advective scalar convergence check
32: if (soluteTransport or (soluteTransportSur and soluteTransportSub))
33: {
34:     scalarField solutePhiBottomSur = ...
            phiCSur.boundaryField()[surfaceBottomPatchID];
35:     scalarField solutePhiTopSub = ...
            phiCSub.boundaryField()[subSurfaceTopPatchID];
36:
37:     scalarField coupleSolutePhiDiff = 0.*solutePhiBottomSur;
38:
39:     forAll (coupleSolutePhiDiff,i)
40:     {
41:         coupleSolutePhiDiff[i] = ...
                solutePhiBottomSur[i] - solutePhiTopSub[i];
42:     }
43:
44:     coupleSolutePhiMaxMismatch = gMax(mag(coupleSolutePhiDiff));
45: }
46: else
47: {
48:     coupleSolutePhiMaxMismatch = 0.0;
```

```
49: }
```

The solution convergence is reached once both water and conservative solute flux mismatches (`couplePhiMaxMismatch` and `coupleSolutePhiMaxMismatch`, respectively) are below the specified tolerances (`couplePhiTol` and `couplePhiSoluteTol`, respectively) in the *simulationProperties* dictionary (**Figure 2.1**).

**File: darcyInterTransportFoam.C**

```
162:         // Surface/subsurface coupling iteration loop
163:         scalar couplePhiMaxMismatch = 1.0;
164:         scalar coupleSolutePhiMaxMismatch = 1.0;
165:         label coupleIterNum = 1;
166:         while (couplePhiMaxMismatch > couplePhiTol and ...
                coupleSolutePhiMaxMismatch > couplePhiSoluteTol)
167:         {
168:             Info << "Coupling iteration #" << coupleIterNum++ ...
                    << " ..." << endl;
169:
170:             Info << nl << "Surface domain ...\n" << endl;
171:
172:             // Pressure-velocity PIMPLE corrector loop
173:             while (pimple.loop())
174:             {

                    ...

194:             }
195:
196:             // Mapping the surface bottom variables to the ...
                    subsurface top patch
197:             #include "mappingSurToSub.H"
198:
199:             Info << "Subsurface domain ..." << endl;

                ...

224:             // Mapping the subsurface top variables to the ...
                    surface bottom patch
225:             #include "mappingSubToSur.H"
226:
227:             // Domains convergence check
228:             #include "domainsConvergence.H"
229:         }
```

Further work will be still required in the future to improve the implemented convergence scheme.

12. To make the most of the existing coupling modes to run the model (one and two-way), a single-domain simulation feature was implemented in **`darcyInterTransportFoam`**. The new add-on enables to define different boundary conditions (BCs) at the interface patch of each domain when the latter are run alone. In this way, both the surface velocity (`U`) and the subsurface hydraulic head (`hSub`) fields can be either specified as the solution from a previous fully-coupled simulation (e.g., `interUSur == "latestTime"`) or set as homogeneous (e.g.,

`interhSub == "consth"`) via a user-defined single value (`Ubottom` and `hSubTop`, respectively) at the corresponding interface boundary.

**File: singleDomainSimulation.H**

```
147:     if (interhSub == "consth")
148:     {
149:         // Constant hydraulic head at the subsurface top patch
150:         scalar hSubTop
151:         (
152:             transportProperties.getOrDefault<scalar>...
                    ("hSubTop",Zero)
153:         );
154:
155:         hSub.boundaryFieldRef()[subSurfaceTopPatchID] == hSubTop;
156:
157:         Info << "homogeneous\n" << endl;
158:     }
159:     else
160:     {
161:         Info << "the one resulting from the surface-subsurface ...
                simulation\n" << endl;
162:     }
```

In addition to the aforementioned, the solute concentration (`CSur` and `CSub`) and the water temperature (`TSur` and `TSub`) fields also allow to set a `zeroGradient` condition at the surface bottom and subsurface top patches. The latter is defined as the default BC for both `CSu*` and `Tsu*` fields, in contrast to the `latestTime` option used by `U` and `hSub`.

**File: singleDomainSimulation.H**

```
54:     if (interTSur == "latestTime" or interTSur == "constT")
55:     {
56:         scalarField Ttmp_sur(TSur.boundaryField()...
                [surfaceBottomPatchID].size(),Zero);
57:
58:         if (interTSur == "latestTime")
59:         {
60:             Ttmp_sur = TSur.boundaryField()[surfaceBottomPatchID];
61:             Info << "the one resulting from the ...
                    surface-subsurface simulation\n" << endl;
62:         }
63:         else
64:         {
65:             // Constant temperature at the surface bottom patch
66:             scalar TSurBottom
67:             (
68:                 mixture.getOrDefault<scalar>("TSurBottom",Zero)
69:             );
70:
71:             Ttmp_sur = TSurBottom;
72:             Info << "homogeneous\n" << endl;
73:         }
74:
75:         surInterfaceHeatFvPatchScalarField& tempTSurp = ...
76:         dynamic_cast<surInterfaceHeatFvPatchScalarField&>...
                (TSur.boundaryFieldRef()[surfaceBottomPatchID]);
77:
```

```
78:         tempTSurp.TFromSub() = Ttmp_sur;
79:     }
80:     else
81:     {
82:         Info << "zeroGradient\n" << endl;
83:     }
```

Both the BC choice (e.g., `interTSur = "zeroGradient"` or `interCSub = "constC"`) and its value (in case the homogeneous option is selected) need to be specified in the *transportProperties* dictionary of the corresponding domain. This novel model feature enables alternative switching between coupling modes within the same simulation in order to take full advantage of the benefits of each to optimize the computation.

13. **`darcyInterTransportFoam`** allows to reinitiate the data fields at any time of the simulation skipping the time-consuming data-preprocessing procedure. This can be done thanks to the multiple, newly implemented reset options. To perform the reset, the new model requires from at least three data: (1) the initial time of the new simulation (`initialTime`), (2) the simulated time from which the reset values are taken (`resetTime`) and (3) the reset option that indicates the data to be restarted (e.g., `resetSur` or `resetHeatSub`). The definitions of both `initialTime` and `resetTime` are optional. By default, the model assigns the latest simulated time to `initialTime` and the initial simulated time to `resetTime`. Additionally, when `initialTime` is not defined by default, the timesteps between `initialTime` and the latest simulated time can optionally be removed via `rmInterTimes`.

**File: resetDomains.H**

```
67:         // Removal of the timesteps between the selected ...
                initial time and the latest
68:         for (label t=times.size()-1; ...
                t>=Time::findClosestTimeIndex(times,initialTime_); t--)
69:         {
70:             runTime.setTime(times[t], 0);
71:             if (rmInterTimes and t != ...
                    Time::findClosestTimeIndex(times,initialTime_))
72:             {
73:                 rmDir(mesh.time().path()/runTime.timeName());
74:             }
75:         }
```

Regarding the choice of the data fields to be reset, the `resetSur` and `resetSub` options allow to restart, respectively, all the surface and subsurface fields.

**File: resetDomains.H**

```
109:     // Reset of all the surface domain fields
110:     if (resetSur)
111:     {
112:         Info << "    Reset of all the surface domain fields" ...
                 << endl;
113:         rmDir(mesh.time().path()/initialTime/"surface");
114:         cp(mesh.time().path()/resetTime/"surface", ...
```

```
                mesh.time().path()/initialTime/"surface");
115:    }
116:
117:    // Reset of all the subsurface domain fields
118:    if (resetSub)
119:    {
120:        Info << "    Reset of all the subsurface domain ...
                fields" << endl;
121:        rmDir(meshSub.time().path()/initialTime/"subsurface");
122:        cp(meshSub.time().path()/resetTime/"subsurface", ...
                meshSub.time().path()/initialTime/"subsurface");
123:    }
```

As an alternative to these, the reset can be limited to the solute transport and heat transfer fields through the `resetSoluteSur`/`resetSoluteSub` options for the surface domain and the `resetHeatSur`/`resetHeatSub` options for the subsurface domain.

**File: resetDomains.H**

```
133:    // Reset of the heat transfer fields at the subsurface domain
134:    if (resetHeatSub)
135:    {
136:        Info << "    Reset of the heat transfer fields at the ...
                subsurface domain" << endl;
137:        cp(meshSub.time().path()/resetTime/"subsurface"/"T", ...
                meshSub.time().path()/initialTime/"subsurface"/"T");
138:    }
139:
140:
141:    // Reset of the solute transport fields at the surface domain
142:    if (resetSoluteSur)
143:    {
144:        Info << "    Reset of the solute transport fields at ...
                the surface domain" << endl;
145:        dictionary soluteSurProp(IFstream("constant/...
                surface/transportProperties")());
146:        word soluteNameSur(soluteSurProp.getOrDefault<word>...
                ("soluteName","C"));
147:
148:        cp(mesh.time().path()/resetTime/"surface"/...
                soluteNameSur, mesh.time().path()/...
                initialTime/"surface"/soluteNameSur);
149:        cp(mesh.time().path()/resetTime/"surface"/...
                "D_" + soluteNameSur, mesh.time().path()/...
                initialTime/"surface"/"D_" + soluteNameSur);
150:
151:        // phiC file is removed because it is not always ...
                present in the time directories
152:        rm(mesh.time().path()/initialTime/"surface"/"phiC");
153:        cp(mesh.time().path()/resetTime/"surface"/"phiC", ...
                mesh.time().path()/initialTime/"surface"/"phiC");
154:    }
```

Both reset times and data field options have to be specified in the *resetOptions* dictionary located in the *constant* directory.

### 4.2.3. Utilities

Four new utilities are available to be used in combination with the new fully-coupled model. Some of them, additionally, are not exclusive of **darcyInterTransportFoam** but also allow its application with other OpenFOAM solvers (e.g., **interFoam** or **pisoFoam**). An extended description of each novel utility is provided below:

- **checkMeshesMatch**: checks whether the number of faces of both meshes and their coordinates match at the interface patch (interfacePatch). The coordinate check is done via a user-specified distance threshold (distThres) after the face number condition is verified. Failure to comply with any of the previous requirements will immediately stop the utility execution and provide an error message.

**File: checkMeshesMatch.C**

```
109:     // Face centre nodes
110:     vectorField pts_inter_mesh = mesh.C().boundaryField()...
             [surfaceBottomPatchID];
111:     vectorField pts_inter_meshSub = meshSub.C().boundaryField()...
             [subSurfaceTopPatchID];

         ...

119:     label count = 0;
120:     while (count < 2)
121:     {
122:         if (Pstream::parRun())
123:         {
124:             // Parallel mode

                 ...

247:         }
248:         else
249:         {
250:             // Serial mode
251:             if (count == 0)
252:             {
253:                 // Check whether the number of faces of ...
                         both meshes is the same at the interface patch
254:                 Info << "    The case is run in serial mode\n" ...
                         << endl;
255:
256:                 Info << "    Number of faces of both meshes ...
                         at the interface patch:"
257:                      << " surface (" << pts_inter_mesh.size() ...
                             << ") - subsurface (" ...
                             << pts_inter_meshSub.size() << ")" ...
                             << endl;
258:
259:                 if (pts_inter_mesh.size() != ...
                         pts_inter_meshSub.size())
260:                 {
261:                     FatalError
262:                     << "The number of faces of both meshes ...
                             differs at the interface patch"
263:                     << ": surface (" << pts_inter_mesh.size() ...
                             << ") - subsurface (" ...
                             << pts_inter_meshSub.size() << ")\n"
264:                     << "    This is a necessary condition to ...
```

```
                            achieve a correct output\n"
265:                    << "    Check blockMeshDict for potential ...
                            errors and try again" << nl
266:                    << exit(FatalError);
267:                }
268:                else
269:                {
270:                    Info << nl << "    The number of faces ...
                            of both meshes is the same at the ...
                            interface patch" << endl;
271:                }
272:            }
273:            else
274:            {
275:                // Check whether the face coordinates ...
                        of both meshes match at the interface patch
276:                forAll(pts_inter_mesh,i)
277:                {
278:                    label minPointi = 0;
279:                    scalar minDistSqr = ...
                            magSqr(pts_inter_meshSub[minPointi] ...
                            - pts_inter_mesh[i]);
280:
281:                    for (label j=1;j<pts_inter_meshSub.size();j++)
282:                    {
283:                        scalar distSqr = ...
                                magSqr(pts_inter_meshSub[j] ...
                                - pts_inter_mesh[i]);
284:
285:                        if (distSqr < minDistSqr)
286:                        {
287:                            minDistSqr = distSqr;
288:                            minPointi = j;
289:                        }
290:                    }
291:
292:                    if (mag(pts_inter_meshSub[minPointi]...
                            -pts_inter_mesh[i]) > distThres)
293:                    {
294:                        FatalError
295:                        << "The face coordinates of both ...
                                meshes do not match at the ...
                                interface patch\n"
296:                        << "    This is a necessary condition ...
                                to achieve a correct output\n"
297:                        << "    Check blockMeshDict for ...
                                potential errors and try again" << nl
298:                        << exit(FatalError);
299:                    }
300:                }
301:
302:                Info << "    The face coordinates of both ...
                        meshes also match at the interface patch\n"
                        ... << endl;
303:                }
304:        }
305:
306:        count++;
307:    }
```

The fulfilment of both conditions guarantees the appropriate fitting of the surface and subsurface meshes. This utility is model-specific, meaning it can only be

applied with OpenFOAM models consisting of two adjacent simulation meshes, such as **darcyInterTransportFoam**.

- **changeZRefMesh**: optionally (1) translates the mesh points (`transformZmesh`) according to a user-specified reference mesh elevation (`zRef`) and (2) adjusts the `hRef` value to the actual mesh location (`waterHeightOutlet`). These two actions are independent; the former does not depend on the latter, and vice versa. Whereas the first option allows to move the mesh to the location specified by `zRef`, the second transforms the user-defined water depth (i.e., `hRef`) into water height by adding the minimum mesh elevation at the outlet to `hRef`.

**File: changeZRefMesh.C**

```
94:     if (numDomains == 1)
95:     {

            ...

111:        scalar mesh_minZ = min(meshPoints.component(vector::Z));
112:        ...
113:        Info << "Minimum surface mesh elevation: " ...
                << mesh_minZ << " [m]" << endl;
114:
115:        mesh_shiftZ = zRef - mesh_minZ;
116:        Info << "Vertical mesh translation: " ...
                << mesh_shiftZ << " [m]\n" << endl;
117:
118:        if (args.found("transformZmesh"))
119:        {
120:            meshPoints += mesh_shiftZ * vector(0,0,1);
121:            meshPoints.write();
122:        }
123:
124:        ...
125:    }
126:    else if (numDomains == 2)
127:    {

            ...

191:    }

        ...

228:    if (args.found("waterHeightOutlet"))
229:    {

            ...

242:        label patchID = mesh.boundaryMesh().findPatchID("outlet");
243:
244:        dimensionedScalar minOutlet
245:        (
246:            "minOutlet",dimLength,min(mesh.boundaryMesh()...
                    [patchID].localPoints()).component(vector::Z)
247:        );
248:
249:        Info << "Minimum outlet elevation in the surface ...
```

```
                domain: " << minOutlet.value() << " [m]" << endl;
250:
251:        hRef += minOutlet;
252:
253:        Info << "Updated hRef: " << hRef.value() << ...
                " [m]\n" << endl;
254:
255:        hRef.write();
256:    }
```

This utility can be applied to one (surface) or two (surface and subsurface) domains (`numDomains`), being thus possible its use beyond **`darcyInterTransportFoam`**.

- **`restoreZRefMesh`**: optionally (1, 2) restores the two actions performed by **`changeZRefMesh`** (i.e., `transformZmesh` and `waterHeightOutlet`) and (3) additionally adjusts the computed hydrological fields `piezoSur`, `hSur` and `hSub` to the final mesh location (`updateHydroFields`). All these actions are independent, meaning that none of them requires from another to be performed. This utility translates the mesh points (*polyMesh* directory) based on the `meshShiftZ` value, which is either previously determined by **`changeZRefMesh`** or supplied by the user in the *constant* directory. The latter option will be always required if **`changeZRefMesh`** is not executed before **`restoreZRefMesh`**.

**File: `restoreZRefMesh.C`**

```
185:    if (args.found("transformZmesh"))
186:    {
187:        scalar mesh_shiftZ = meshShiftZ.value();
188:        Info << "Vertical mesh translation: " << ...
                -1 * mesh_shiftZ << " [m]\n" << endl;
189:
190:        if (numDomains == 1)
191:        {

                ...

207:            meshPoints -= mesh_shiftZ * vector(0,0,1);
208:            meshPoints.write();
209:        }
210:        else if (numDomains == 2)
211:        {

                ...

247:        }
248:    }
```

`meshShiftZ` is additionally used to update the `piezoSur`, `hSur` and `hSub` fields according to the resulting mesh elevation.

**File: `restoreZRefMesh.C`**

```
250:    if (args.found("updateHydroFields"))
251:    {
252:        // Piezometric level (L)

            ...
```

```
266:          piezoSur == piezoSur - meshShiftZ;
267:
268:          // Hydraulic head (L)

              ...

282:          hSur == hSur - meshShiftZ;

              ...

309:          if (alphaAirClip)
310:          {

                  ...

325:              piezoSur *= pos0(alpha1 - alphaWaterThres);
326:              hSur *= pos0(alpha1 - alphaWaterThres);
327:          }
328:
329:          Info << "Piezometric level updated." << endl;
330:          piezoSur.write();
331:
332:          Info << "Surface hydraulic head updated.\n" << endl;
333:          hSur.write();
334:
335:          if (numDomains == 2)
336:          {

                  ...

362:              hSub == hSub - meshShiftZ;
363:
364:              Info << "Subsurface hydraulic head updated.\n" ...
                      << endl;
365:              hSub.write();
366:          }
367:      }
```

Moreover, `hRef` is reestablished to its original, user-defined value by subtracting the minimum mesh elevation at the outlet to the formerly calculated water height.

**File: restoreZRefMesh.C**

```
142:      if (args.found("waterHeightOutlet"))
143:      {

              ...

156:          label patchID = mesh.boundaryMesh().findPatchID("outlet");
157:
158:          dimensionedScalar minOutlet
159:          (
160:              "minOutlet",dimLength,min(mesh.boundaryMesh()...
                      [patchID].localPoints()).component(vector::Z)
161:          );
162:
163:          Info << "Minimum outlet elevation in the surface ...
                  domain: " << minOutlet.value() << " [m]" << endl;
164:
165:          hRef -= minOutlet;
166:
```

```
167:          Info << "Restored hRef: " << hRef.value() << ...
                  " [m]\n" << endl;
168:
169:          hRef.write();
170:      }
```

Analogously to **changeZRefMesh**, this utility is also applicable to one (surface) or two (surface and subsurface) domains (numDomains). Consequently, its use is not limited to the new fully-coupled model but also open to other **OpenFOAM** solvers. The utility setup and its outcomes will ultimately depend on the prior application of **changeZRefMesh**.

- **transferMeshDecomp**: transfers the domain decomposition from one mesh to another through the interface patch of both meshes (interfacePatch). Moreover, it optionally overwrites the *cellDecomposition* file contained in the *constant* directory of each domain. This utility requires a previous decomposition of each domain (**decomposePar** utility) that generates the necessary *cellDist* and *cellDecomposition* files for its execution. The decomposition transfer is then conducted by subdomain in two steps (nSubDomains). First, the subdomain extension is delimited at the interface path of the reference mesh (rbDeltaX, rbDeltaY and rbDeltaZ).

**File: transferMeshDecomp.C**

```
277:      for (label nSub=0; nSub<nSubDomains; nSub++)
278:      {

              ...

301:          // Vectors defining the rotated boxes which ...
                  contain the cell centres of each meshTo subdomain
302:          vector origin = vector(min(patchFrom.component(vector::X))
                  ...-rbDeltaX,min(patchFrom.component(vector::Y))
                  ...-rbDeltaY,min(patchFrom.component(vector::Z))
                  ...-rbDeltaZ);
303:          vector i_ = vector(max(patchFrom.component(vector::X))...
                  +rbDeltaX-origin.component(vector::X),...
                  min(patchFrom.component(vector::Y))-rbDeltaY...
                  -origin.component(vector::Y),0.0);
304:          vector j_ = vector(min(patchFrom.component(vector::X))...
                  -rbDeltaX-origin.component(vector::X),...
                  max(patchFrom.component(vector::Y))+rbDeltaY...
                  -origin.component(vector::Y),0.0);
305:          vector k_ = vector(0.0,0.0,...
                  max(patchFrom.component(vector::Z))+rbDeltaZ...
                  -origin.component(vector::Z));
306:
307:          // Rotated box vertices
308:          pointField boxPoints(8);
309:          boxPoints[0] = origin;
310:          boxPoints[1] = origin + i_;
311:          boxPoints[2] = origin + i_ + j_;
312:          boxPoints[3] = origin + j_;
313:          boxPoints[4] = origin + k_;
314:          boxPoints[5] = origin + k_ + i_;
```

```
315:          boxPoints[6] = origin + k_ + i_ + j_;
316:          boxPoints[7] = origin + k_ + j_;

              ...

395:      }
```

Next, the corresponding subdomain ID is assigned to all the cells of the target mesh falling inside the previously defined limits (`distThres`). This procedure is possible in two ways, that is, from the surface to the subsurface and vice versa (`transferDir`).

**File: transferMeshDecomp.C**

```
341:          // Cell centres of the meshTo
342:          pointField ctrsTo = meshTo.cellCentres();

              ...

347:          // Check whether the meshTo cell centres are ...
                  inside all the faces of the rotated boxes
348:          forAll(ctrsTo,celli)
349:          {
350:              bool inside = true;
351:
352:              forAll(boxFaces,i)
353:              {
354:                  const face& f = boxFaces[i];
355:
356:                  if (((ctrsTo[celli] - boxPoints[f[0]]) ...
                      & boxFaceNormals[i]) > 0)
357:                  {
358:                      inside = false;
359:                      break;
360:                  }
361:              }
362:
363:              if (inside)
364:              {
365:                  // Set to zero the third dimension of ctrsTo ...
                          for the correct selection of meshTo ...
                          cell centres
366:                  ctrsTo[celli].component(vector::Z) *= 0.0;
367:
368:                  // Selection of the meshTo cells corresponding ...
                          to each extended meshFrom subdomain
369:                  label minPointi = 0;
370:                  scalar minDistSqr = magSqr(patchFrom[minPointi]...
                      - ctrsTo[celli]);
371:
372:                  for (label i=1;i<patchFrom.size();i++)
373:                  {
374:                      scalar distSqr = magSqr(patchFrom[i]...
                              - ctrsTo[celli]);
375:
376:                      if (distSqr < minDistSqr)
377:                      {
378:                          minDistSqr = distSqr;
379:                          minPointi = i;
380:                      }
381:                  }
```

```
382:
383:                 ...
384:
385:                 // Subdomain ID assignment to each selected ...
                         meshTo cell
386:                 if (mag(patchFrom[minPointi]-ctrsTo[celli]) ...
                         <= distThres)
387:                 {
388:                     cellDistTo[celli] = nSub;
389:                     cellDecompositionTo[celli] = nSub;
390:                 }
391:             }
392:         }
```

Once the transfer is done, the user can decide whether or not overwriting the *cellDecomposition* file of the reference (`overWriteCellDecompBase`) and/or target mesh (`overWriteCellDecomp`).

**File: transferMeshDecomp.C**

```
399:     // Overwrite cellDecomposition file of meshTo
400:     if (overWriteCellDecomp)
401:     {
402:         cellDecompositionTo.write();
403:     }
404:
405:     // Overwrite cellDecomposition file of meshFrom
406:     if (overWriteCellDecompBase)
407:     {
408:         forAll (cellDistFrom,i)
409:         {
410:             cellDecompositionFrom[i] = cellDistFrom[i];
411:         }
412:
413:         cellDecompositionFrom.write();
414:     }
```

In the same way as **checkMeshesMatch**, the application of this utility is not strictly limited to **darcyInterTransportFoam** but to OpenFOAM models comprised by two adjacent meshes.

### 4.2.4. Boundary Conditions

Three new boundary conditions, one for the surface velocity field, $U_{sur,i}$ (m/s), and two for the subsurface hydraulic head field, $h_{sub,i}$ (m), were developed to be used with **darcyInterTransportFoam**. The groundwater BCs set implicitly the heads, since their value is either retrieved from the surface domain or inferred from the prescribed velocity at the patch. Moreover, as happens with the aforementioned utilities, some of the novel BCs do not restrict its application to **darcyInterTransportFoam** but also allow its use with other standard OpenFOAM solvers. A detailed description of all the new BCs can be found within the following points:

- **`flowRateInletParabolicVelocity`**: optionally prescribes at the selected patch an inward-pointing normal, parabolic $U_{sur,i}$ (`U`) based on the specified flow rate, $Q_{sur}$ (m³/s – `volumetricFlowRate`), the dimensions of the velocity profile (`Uprofile == "parabolic"` – 2D – or `"paraboloidal"` – 3D), the type of inlet geometry (`geometryInlet == "open"` or `"closed"`), the average velocity multiplier, $multi_{\bar{U}}$ (`avgUmulti`), and the water depth direction (`depthDir`). The shape of the resulting velocity profile (either a half or a quarter of a parabola) will primarily depend on the geometry type of the inlet (Figure 4.1).

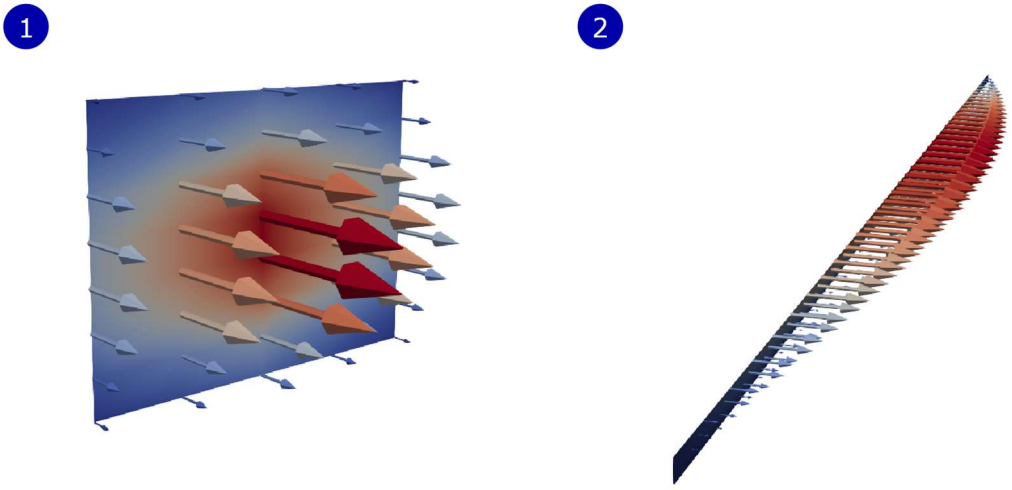



**Figure 4.1. `flowRateInletParabolicVelocity` BC paraboloidal velocity profile: (1) half paraboloid generated at a closed inlet patch and (2) quarter of a paraboloid prescribed at an open inlet patch.**

This new BC allows to avoid potential numerical issues at the conjunction with no-slip patches (commonly bottom and side boundaries) when applying a uniform velocity field. The boundary condition setup results from the combination of the standard **`flowRateInletVelocity`** BC and the **`parabolicVelocity`** BC developed by Hrvoje Jasak, freely available in **foam-extend-5.0**. As an example of its functionality, the following equation shows how the paraboloidal velocity profile is determined at a closed inlet patch:

$$U_{sur,i} = \vec{n}_i \cdot multi_{\bar{U}} \cdot \frac{Q_{sur}}{A_{p_{w,i}}} \cdot \left(1 - \sqrt{\frac{(\vec{f}_{c_i} - \vec{p}_c) \cdot \overrightarrow{dir}_H}{0.5 \cdot H}}\right) \cdot \left(1 - \sqrt{\frac{(\vec{f}_{c_i} - \vec{p}_c) \cdot \overrightarrow{dir}_W}{0.5 \cdot W}}\right) \tag{4.2}$$

where $n_i$ (`n`) is the patch face unit normal vector field (-), $A_{p_{w,i}}$ (`patch().magSf()`) is the area of the patch faces through which water flows into the domain (m²), $\vec{f}_{c_i}$

(`faceCtr`) is the patch face centre vector field (m), $\vec{p}_c$ (`paraCtr`) is the paraboloid centre vector (m), $\overrightarrow{dir}_H$ (`depthDir`) and $\overrightarrow{dir}_W$ (`transDir`) are respectively the patch height and width direction vectors (-) and $H$ (`pHeight`) and $W$ (`pWidth`) are the height and width of the patch (m).

**File: flowRateInletParabolicVelocityFvPatchVectorField.C**

```
169:     const scalar t = db().time().timeOutputValue();
170:
171:     const vectorField n(patch().nf());
172:
173:     // Get range and orientation
174:     boundBox bb(patch().patch().localPoints(), true);
175:     //Info << "Bounding box: " << bb << endl;
176:
177:     const vectorField& faceCtr = patch().Cf();
178:     //Info << "Patch faces centres: " << faceCtr << endl;
179:
180:     // Patch normal vector: weighted mean of patch face ...
             normal vectors
181:     vector n_vector = gSum(patch().magSf()*n)/...
             gSum(patch().magSf());
182:     //Info << "Patch mean normal: " << n_vector << endl;
183:
184:     vector transDir = n_vector ^ depthDir_;
185:     //Info << "Transversal direction: " << transDir << endl;
186:
187:     scalar pHeight = ((bb.max() - bb.min()) & depthDir_);
188:     scalar pWidth = ((bb.max() - bb.min()) & transDir);

         ...

204:         vector paraCtr = vector(0.5*(bb.max() + bb.min()).x(),...
                 0.5*(bb.max() + bb.min()).y(),...
                 0.5*(bb.max() + bb.min()).z());
205:         //Info << "Parabola centre: " << paraCtr << endl;
206:
207:         // Calculate local 1-D coordinate for the ...
                parabolic profile
208:         verCoord = ((faceCtr - paraCtr) & depthDir_)/...
                 (0.5*pHeight);
209:         horCoord = ((faceCtr - paraCtr) & transDir)/...
                 (0.5*pWidth);

         ...

218:     const scalar avgU = -flowRate_->value(t)/...
             gSum(rho*patch().magSf());

         ...

227:         // Parabolidal U profile
228:         operator==(n*avgUmulti_*avgU*(1.0 - sqr(verCoord))*...
                 (1.0 - sqr(horCoord)));
```

Analogously to most standard velocity BCs, the use of **`flowRateInletParabolicVelocity`** is not limited to the new model but extensible to other **OpenFOAM** models.

- **darcyGradHead**: sets at the patch the $\nabla h_{sub,i}$ resulting from the flux generated by the specific discharge, $q_i$ (m/s), boundary condition. Such $h_{sub,i} - q_i$ (`hSub - q`) relationship is inferred from Darcy's law as:

$$\vec{n}_i \cdot \nabla h_{sub,i} = -\frac{1}{K_{ij}} \cdot \vec{n}_i \cdot q_i \tag{4.3}$$

where $K_{ij}$ (`K`) is the hydraulic conductivity tensor field (m/s). This new BC provides an alternative way to specify $h_{sub,i}$ for cases where the definition of $q_i$ is more convenient than the direct assignment of $h_{sub,i}$ at the patch.

**File: darcyGradHeadFvPatchScalarField.C**

```
122:     const fvPatchVectorField& q =
123:         patch().lookupPatchField<volVectorField,vector>(qName_);
124:
125:     const fvPatchTensorField& K =
126:         patch().lookupPatchField<volTensorField,tensor>(KName_);
127:
128:     gradient() = -inv(K) & q & patch().nf();
```

Originally inspired by the standard **fixedFluxPressure** BC, **darcyGradHead** has been adapted from **darcyGradPressure**, a head BC previously developed by Cyprien Soulaine as part of the **porousMultiphaseFoam** toolbox (Horgue, et al., 2015). Similarly to **flowRateInletParabolicVelocity**, its application to **OpenFOAM** models other than **darcyInterTransportFoam** is also possible.

- **surfaceHydrostaticHead**: specifies at the inlet, outlet or aquifer (i.e., bottom) patch of the subsurface domain a hydrostatic head BC based on the $h_{sub,i}$ values obtained in the surface domain. In this way, if applied to the subsurface inlet or outlet, this BC will take and assign the hydrostatic head value, $max(piezo_{sur,i})$ (`gMax(piezo)`), computed at the equivalent patch of the surface domain.

**File: surface\patchHydrostaticHead.H**

```
13:             fvPatchScalarField piezo = ...
                    piezoSur.boundaryField()[patchID];

                ...

17:                 piezo *= pos0(alpha1.boundaryField()[patchID] ...
                        - 0.5);

                    ...

35:                 if (patchName == "outlet")
36:                 {
37:                     label subPatchID = ...
                            meshSub.boundaryMesh()...
                            .findPatchID(patchName);
38:
39:                     // Check the patch type to transfer the ...
                           data to the BC
```

```
40:                     if (isA...
                            <surfaceHydrostaticHeadFvPatchScalarField>
                         ...(hSub.boundaryField()[subPatchID]))
41:                     {
42:                         surfaceHydrostaticHeadFvPatchScalarField&
                             ...temphOutSubp = ...
43:                             dynamic_cast...
                            <surfaceHydrostaticHeadFvPatchScalarField&>
                         ...(hSub.boundaryFieldRef()[subPatchID]);
44:
45:                         temphOutSubp.hOutFromSur() = gMax(piezo);
46:                     }
47:                 }
```

**File: surfaceHydrostaticHeadFvPatchScalarField.C**

```
142:     scalarField hPatch = *this;

         ...

151:         if (patch().name() == "outlet")
152:         {
153:             hPatch[i] = hOutFromSur_;
154:         }

         ...

162:     operator==(hPatch);
```

Analogously, if set at the subsurface bottom boundary, it will retrieve the hydrostatic head field, $piezo_{sur,i}$ (`piezo[minPointi]`), from its surface counterpart.

**File: surface\patchHydrostaticHead.H**

```
55:                 if (isA...
                        <surfaceHydrostaticHeadFvPatchScalarField>...
                        (hSub.boundaryField()...
                        [subSurfaceBottomPatchID]))
56:                 {
57:                     scalarField htmp_sub(hSub.boundaryField()...
                            [subSurfaceBottomPatchID].size(),Zero);
58:
59:                     forAll(faceCtr_subBottom,i)
60:                     {
61:                         label minPointi = 0;
62:                         scalar minDistSqr = magSqr(...
                                faceCtr_surBottom[minPointi] ...
                                - faceCtr_subBottom[i]);
63:
64:                         for (label j=1;...
                                j<faceCtr_surBottom.size();j++)
65:                         {
66:                             scalar distSqr = magSqr(...
                                    faceCtr_surBottom[j] ...
                                    - faceCtr_subBottom[i]);
67:
68:                             if (distSqr < minDistSqr)
69:                             {
70:                                 minDistSqr = distSqr;
71:                                 minPointi = j;
```

```
72:                                 }
73:                             }
74:
75:                             htmp_sub[i] = piezo[minPointi];
76:                         }
77:
78:                         surfaceHydrostaticHeadFvPatchScalarField& ...
                                temphRiverSubp = ...
79:                             dynamic_cast...
                                <surfaceHydrostaticHeadFvPatchScalarField&>
                             ...(hSub.boundaryFieldRef()...
                                [subSurfaceBottomPatchID]);
80:
81:                         ...
82:                         temphRiverSubp.hRiverFromSur() = htmp_sub;
83:                     }
```

**File: surfaceHydrostaticHeadFvPatchScalarField.C**

```
156:          if (patch().name() == "aquifer")
157:          {
158:              hPatch[i] = hRiverFromSur_[i];
159:          }
```

The current version of **surfaceHydrostaticHead** does not allow its application to any patches beyond the cited. This novel boundary condition avoids the unrealistic behaviour caused by the commonly used **zeroGradient** BC. The latter works as an impermeable layer, forcing the flow paths to change their direction in the vicinity of the patch. **surfaceHydrostaticHead**, however, provides a closer approximation to reality and allows the water to flow across the boundary. The direction of the fluxes will depend on the sign of the $\nabla h_{sub,i}$ at the patch. Moreover, in contrast to the other two new BCs, the present one is model-specific, meaning it can only be used with **darcyInterTransportFoam**.

## 4.3. MATLAB Scripts for Efficient Generation of OpenFOAM Dictionaries

A set of MATLAB scripts was developed in parallel to **darcyInterTransportFoam** to streamline the model case preparation. The script suite provides functionalities such as generating both surface and subsurface meshes and splitting the inlet boundary of the surface mesh into two patches. Although originally designed as a dictionary generator for **darcyInterTransportFoam**, the script set is also compatible with other OpenFOAM solvers (OpenCFD Ltd, 2023), extending its utility beyond the original scope.

### 4.3.1. blockMesh Dictionary

In **OpenFOAM**, the *blockMeshDict* file is used to generate three-dimensional structured meshes consisting of hexahedral blocks defined by straight or curved edges. While the features of simple meshes (e.g., vertices, edges or boundaries) can be manually defined by the user within this file, this process becomes increasingly complex when dealing with more sophisticated geometries, such as real topographical scenarios (**Chapter 2**) or periodic shapes like sinusoidal series (**Chapter 3**). In such cases, third-party mesh generators (e.g., **cfMesh**, **Salome**, or **gmsh**) are commonly used to create the desired structured meshes. Some of these tools can directly output meshes in the **OpenFOAM** format, while others may require a conversion step before being used for simulation.

Alternatively, user-defined scripts can streamline the *blockMeshDict* setup, bypassing the time-consuming process of manual definition and providing full flexibility to tailor the code to specific mesh requirements. Furthermore, they are often free to use and help avoid the learning curve associated with any pre-processing meshing software. Accordingly, a custom script, **`mesh_and_data_dicts.m`**, was developed in the present PhD thesis along with various functions and supplementary scripts (**`Froude_no_yc.m`** and **`Manning_eqn_yn.m`**) to generate all the necessary *blockMeshDict* files for the established simulation cases. Their source code is publicly hosted on GitHub and permanently archived on Zenodo.

**`mesh_and_data_dicts.m`** generates the structured mesh based on seven core elements: (1) the two-dimensional, longitudinal outline vertices, (2) the defined transversal sections, (3) the shape of the interface patch, (4) the surface mesh height and subsurface mesh depth, (5) the mesh resolution, (6) the shape of the inlet and outlet patches and (7) the water height at the inlet and outlet patches. All this information must be provided by the user, along with additional case-specific details. Depending on the parameter, the data will either need to be entered directly into the code or imported from a data file. The main meshing actions available in the script, together with their associated input parameters and corresponding section in the code, are numbered and detailed as follows:

1. **Definition of the interface patch topography** `[Sections 1.3 and 1.11]`: based on either a Digital Elevation Model (DEM) or different configurations of rippled, sinusoidal patterns (`topo_file`) (**Figure 4.2** and **Figure 4.4**, respectively). The former is imported from a raster or vector file (`DEM_riverbed`), while the latter is generated according to several specified parameters (e.g., `ripple_pattern`, `ripple_dir` or `smoother`). Additionally, the resulting interface topography can be modified afterwards (`topography_shape`) to create a rectangular or trapezoidal channel (`channel_shape`), using the mean elevation values of its central section as a reference.

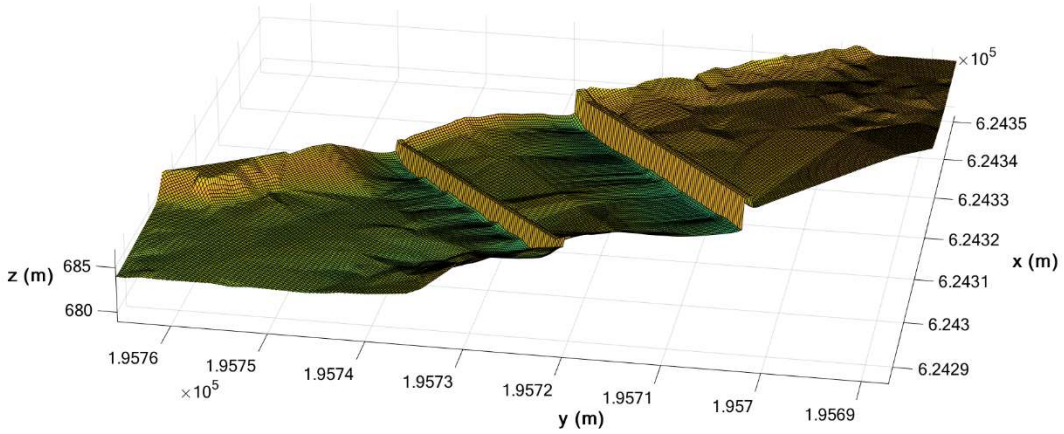



**Figure 4.2.** DEM-based interface patch topography (including the river steps) of TC1 described in Chapter 2.

2. **Setup of the longitudinal discretization of the mesh** `[Section 1.5]`: either by defining vertices along the mesh outlines (`interp_tool == 'matlab'`) according to a user-specified longitudinal resolution (`y_resol`), or by importing the 2D coordinates of these vertices from a data file (`interp_tool == 'CAD'`).
3. **Domain clipping** `[Section 1.7]`: reduction of the domain's extent based on user-defined boundary limits (`domain_clipping`).
4. **Coordinate reduction** `[Section 1.8]`: translation of the mesh by subtracting the minimum coordinate value in the specified direction(s) (`domain_coord`). This reduction can be applied in one, two or all three spatial directions (`reduced_coord`).
5. **Mesh resolution definition** `[Section 1.9]`: the resolution can be separately specified for each spatial direction (`x_resol`, `y_resol`, `z_resol`). Note that `y_resol` will only be required when using the `matlab` option for mesh discretization. For optimal results, `z_resol` should be set to approximately 1/10 of `x_resol`.
6. **Longitudinal domain expansion** `[Section 1.13]`: outward extension of the mesh from the inlet and/or outlet patches in their respective normal directions (`domain_extension`). The expansion is based on a user-specified number of cells to add to each patch (`in_stretch` and `out_stretch`).
7. **Selection of the inlet and outlet patches' locations** `[Section 1.14]`: both patches can be positioned either at the upstream and downstream boundaries of the mesh, respectively, or adjacent to those boundaries at the interface patch

(`inlet_location`, `outlet_location`). Note that, if placed at the upstream boundary, the inlet is additionally divided into two patches: `inletAir` and `inletWater`.

8. **River steps definition** `[Sections 1.12 and 1.15]`: inclusion of artificial steps at specified locations along the riverbed (interface patch - `step_inclusion`). The number, positions and characteristics of these hydraulic structures (e.g., top elevation of the step, length of the hole or transition length hole-riverbed) must be provided in a data file, from which this information is imported for generation. The steps are then adapted to match the riverbed topography (**Figure 4.2** and **Figure 4.3**). Furthermore, mesh refinement around the steps (`steps_refine`) can be applied in one, two or all three spatial directions (`ref_dir`), with settings also specified in the imported dataset. Likewise, the script includes an option to generate an additional downstream weir to maintain a constant water level at the outlet (`downstream_weir`). Unlike the river steps, this structure's features are defined directly within the code, without requiring an imported file.

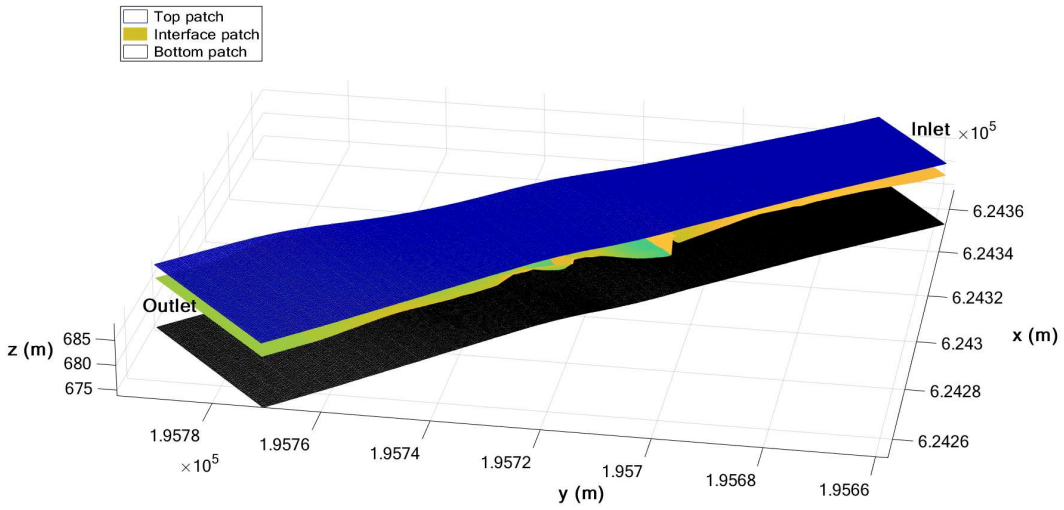



**Figure 4.3.** MATLAB-generated top, interface and bottom patches of the simulation mesh used for TC1 in **Chapter 2**.

9. **Creation of the surface top and subsurface bottom patches** `[Sections 1.17 and 1.19]`: both boundaries are derived from the interface patch by adding (`height`) or subtracting (`depth`) a user-defined value (**Figure 4.3** and **Figure 4.4**). Alternatively, the surface top boundary can be also calculated as a specific percentile of the interface patch elevations (`percentile`). The surface top generation method must be selected beforehand (`air_patch`).

10. **Computation of the water height at the inlet and outlet patches** `[Section 1.20]`: this is done based on Manning's equation (`Manning_para`). The calculation is

performed via the supplementary script **`Manning_eqn_yn.m`**, which combines patch geometry data transferred from the main code (e.g., `riverSlope`, `bottom_width` or `unit_normal`) with user-specified parameters (e.g., `section_shape`, `cross_sec`, `Q` or `n`). The resulting water height at the inlet serves to establish the limit between the `inletAir` and `inletWater` patches.

11. **Definition of the inlet and outlet patches' shapes** `[Section 1.22]`: both boundaries (`reg_patches`) can be set as rectangular or trapezoidal (`inletOutlet`), regardless of their original riverbed-based shape. In such cases, a buffer length can also be specified to smooth the transition between the corresponding regular patch and the actual riverbed profile (`smooth_edges_in` and `smooth_edges_out`).

### 4.3.2. Additional Mesh and Input Data Dictionaries

Beyond generating the *blockMeshDict* file, **`mesh_and_data_dicts.m`** can also export additional **OpenFOAM** input files, making it a comprehensive tool for initializing various aspects of the simulation setup. Some of these files are intended for specific meshing actions, such as *topoSetDict*, *createPatchDict* or *refineMeshDict*, while others, like *setFieldsDict*, are used to assign field values to selected sets of cells or patch faces. The potential actions enabled by the code, along with their corresponding input parameters and related output files, are described below:

1. **Creation of stereolithography (STL) rectangular volumes** `[Section 1.26]`: these are generated (`stl_surface`) based on two surfaces (i.e., top and bottom horizontal faces), with their creation process differing depending on whether the STL volume is applied to the surface or subsurface domain. For surface volumes, the first surface is created by adding a user-specified shift value (`shift_boxTop`) to either an imported scatter surface or the averaged transversal elevations of the riverbed's central section (`surface_type`). For subsurface volumes, this surface is instead created by subtracting a shift value (`shift_boxBottom`). The second surface is then defined by subtracting `shift_boxBottom` from the minimum transversal elevations of the first surface (for surface volumes) or by adding `shift_boxTop` to the maximum transversal elevations (for subsurface volumes). The resulting volumes are intended to be used with the `surfaceToCell` action to delimit mesh regions for assigning field values via *setFieldsDict* or defining cell sets via *topoSetDict*. The directory paths for importing scatter surfaces, as well as the other necessary parameters to generate the STL volumes (e.g., `mesh_transfer`, `surface_type` or `stlRegion`), must be defined within the script.

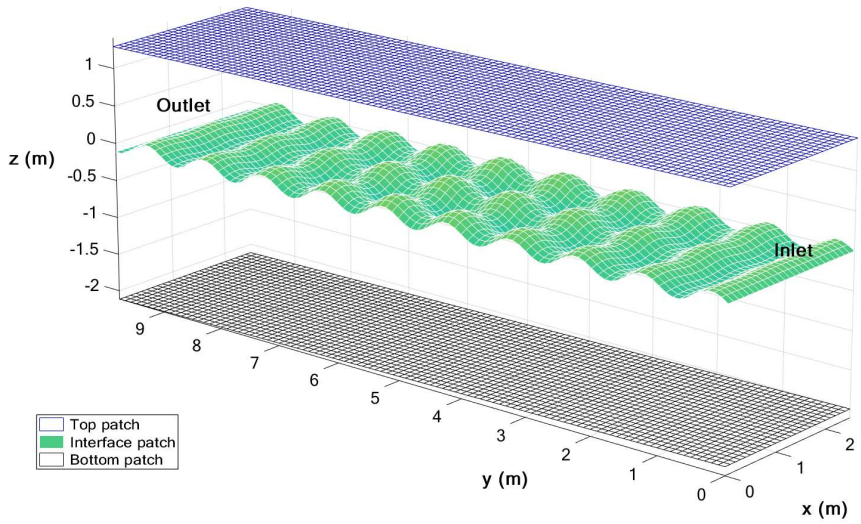



Figure 4.4. MATLAB-generated top, interface and bottom patches of the simulation mesh used in Chapter 3.

2. **Generation of additional mesh-related dictionaries** `[Section 1.28]`: creation of all the necessary files to complete the mesh generation process, including *topoSetDict*, *createPatchDict* and *refineMeshDict* (`add_mesh_data_dic`). The *topoSetDict* file is used in combination with either *createPatchDict* to define new patches or *refineMeshDict* to further refine the mesh at specific locations. The actions `rotatedBoxToCell` and `surfaceToCell` are defined within *topoSetDict* to select the cell sets where the additional mesh actions will be applied. In the current version of the code, *createPatchDict* is used to define the `inletWater` patch, while *refineMeshDict* is applied to refine the mesh around the steps in TC1 of Chapter 2. All the parameters required to generate these files (e.g., `inWaterName`, `stl_region`, `ref_dir`, `closed_box` or `box_region`) must be specified within the code.

3. **Generation of input data dictionaries** `[Sections 1.27 and 1.28]`: creation of the *setFieldsDict* and *topoSetDict* files required to define fields and analyse simulation results. The generated *setFieldsDict* files (`add_mesh_data_dic`) enable the initialization of simulation fields, the assignment of subdomain IDs for mesh decomposition and the definition of subsurface parametrization. Similar to the previously described mesh-related files, the `rotatedBoxToCell` and `surfaceToCell` actions are used to select the cell sets where the defined fields will be applied. Additionally, a separate *topoSetDict* file can be created (`mesh_cross_sec`) to define the transversal cross-sections for measuring user-specified simulation variables, using the `searchableSurfaceToFaceZone` action. As with the mesh dictionaries, all the information required to generate these files

(e.g., `field_type`, `field_var`, `stl_var_values`, `boxes_def` or `sections_def`) must be entered in the code.

## 4.4. Conclusions

The code enhancements and new add-ons introduced in **`darcyInterTransportFoam`** were thoroughly described in this chapter, along with the MATLAB scripts developed to generate the OpenFOAM dictionaries required to set up a simulation case.

The novel developments enhance the applicability of **`hyporheicScalarInterFoam`** by extending its modelling capabilities, enabling the new solver to simulate real SW-GW interaction scenarios across scales that the previous version could not adequately represent.

Regarding the script set, the implemented approach provides full control over the mesh generation process, allowing for quicker and more efficient mesh configurations, customizations and initial condition setups.

Ultimately, the provided descriptions improve the clarity and reproducibility of both codes, strengthening the link between theory and implementation.

## References

[1] Horgue, P., Soulaine, C., Franc, J., Guibert, R., & Debenest, G. (2015). An open-source toolbox for multiphase flow in porous media. *Computer Physics Communications, 187*, 217-226. https://doi.org/10.1016/j.cpc.2014.10.005

[2] Lee, A., Aubeneau, A., Liu, X., & Cardenas, M. B. (2021). Hyporheic Exchange in Sand Dunes Under a Freely Deforming River Water Surface. *Water Resources Research, 57*(3), e2020WR028817. https://doi.org/10.1029/2020WR028817

[3] OpenCFD Ltd. (2023, December). *OpenFOAM® v2312 User Guide*. Retrieved December 25, 2024, from https://doc.openfoam.com/2312/

[4] Pardo-Álvarez, Á. (2025). *darcyInterTransportFoam v1.0*. [Software], Zenodo. https://doi.org/10.5281/zenodo.15857142

[5] The MathWorks Inc. (2024). *MATLAB R2024a*. Retrieved December 25, 2024, from https://www.mathworks.com/products/matlab.html

# 5

# Summary and Conclusions

## 5.1. Summary

Numerical models have proven to be a valuable complement to field observation techniques for studying complex SW-GW interactions across the sediment-water interface. Unlike labour-intensive field experiments, numerical models can digitally replicate complex field sites at a reasonable computational cost and provide high-resolution spatiotemporal data on flow, solute transport and heat transfer processes under varying forcing conditions, thus enabling a wide range of scenario analyses and predictions. Among the various modelling approaches developed in recent years, fully-coupled and fully-integrated models (FCMs and FIMs, respectively) enable capturing the continuous, bidirectional exchange between SW and GW. While both approaches use diverse sets of equations to simulate the surface and subsurface processes, only the full three-dimensional Navier-Stokes equations can resolve the complex dynamics of surface flow, including turbulence, and thus provide high-resolution details on the flow field characteristics. Such a comprehensive simulation of the surface hydrodynamics ultimately allows for a more accurate representation of the actual two-way interactions across the SW-GW interface. These advanced flow equations are solved by Computational Fluid Dynamics (CFD) models, such as most of the solvers within the free and open-source **OpenFOAM** toolbox (Weller, et al., 1998). The modular and extensible code structure of **OpenFOAM** makes this powerful CFD software an ideal platform for implementing new numerical models aimed at the detailed study of continuous, three-dimensional SW-GW exchanges. Accordingly, **OpenFOAM** has been used in recent years to develop several user-defined FCMs. However, limitations in these models have restricted their broader application to SW-GW interaction scenarios. In response to this, the present PhD thesis introduces a new FCM designed to address these limitations and extend the functionality of the previous models. The features of this novel SW-GW model are thoroughly described in this document, along with an evaluation of its performance through several test cases, including a comparison with another standard fully-integrated **OpenFOAM** solver.

A summary of the different chapters that comprise this PhD thesis, their conclusions and future research needs is provided as follows:

- Chapter 2 -- **darcyInterTransportFoam v1.0: an Open-Source, Fully-Coupled 3D Solver for Simulating Surface Water – Saturated Groundwater Processes and Exchanges:** an updated and extended version of `hyporheicScalarInterFoam`, a fully-coupled, three-dimensional SW-GW solver (Lee, et al., 2021), is presented in this chapter along with an evaluation of its performance through two test cases. The upgraded model, named `darcyInterTransportFoam` (Pardo-Álvarez, 2025), introduces new features that enhance the original solver's functionality and capabilities by addressing modelling gaps and adding further simulation options. The code structure and organization have also been refined for improved user

accessibility. The conceptual framework of both solver updates and novel simulation add-ons are thoroughly described throughout this chapter. The results obtained at the two test cases corroborate that, despite its existing limitations, `darcyInterTransportFoam` covers most of the simulation gaps of its predecessor. The new FCM successfully simulates the bidirectional water, solute and heat exchanges at two reach-scale locations within a highly conductive river-aquifer system, validating the performance of the novel modelling features. Moreover, the outcomes demonstrate its potential to investigate SW-GW processes at scales beyond the hyporheic zone. The added modelling capacities pave the way to explore new relevant aspects of SW-GW interactions. Notably, it is the first time that the subsurface characterization (e.g., the heterogeneity of the physical properties of the riverbed and underlying aquifer) can be accounted for in the simulations. Moreover, this heterogeneity can now be considered in the simulation of solute transport and heat transfer processes. Despite the promising results, further code development will still be necessary to address the current model limitations and enable a detailed simulation of the SW-GW dynamics and biogeochemical reactions across any hydrological and physical scenario (Section 5.2).

- Chapter 3 -- **A Comparison Study of a Fully-Coupled vs. a Fully-Integrated OpenFOAM Solver for Simulating Surface Water - Groundwater Interactions:** the computational efficiency and performance of `interFoam`, a standard fully-integrated **OpenFOAM** solver (Deshpande, et al., 2012), and `darcyInterTransportFoam` were compared across a three-dimensional, rippled streambed to evaluate their suitability for simulating high-resolution SW-GW interactions. The comparison of both flow and solute-related fields reveals an overall similarity between the two modelling approaches at steady-state. However, discrepancies appear close to the riverbed and in the subsurface domain. In the latter, the application of the Navier-Stokes equations together with the Darcy-Forchheimer law in `interFoam` yields approximately 33% higher subsurface velocities than those produced by Darcy's law in `darcyInterTransportFoam`. The faster groundwater flow in `interFoam` results in higher hydraulic heads and quicker solute travel times. Near the riverbed, dissimilarities arise from the different interface definitions associated with each model's coupling method. Such distinct treatment leads to significantly different velocity fields around the SW-GW interface, including non-Darcy flows velocities just below the riverbed in `interFoam`. Nevertheless, these differences are limited to the first few centimetres from the sediment-water interface, having a minor effect on the magnitude of the bidirectional exchanges between the surface and subsurface domains. As for the computational efficiency, `interFoam` requires approximately 18 hours more of runtime than `darcyInterTransportFoam` to reach the steady-state, showcasing

its higher computational demands due to the use of the Navier-Stokes equations for the groundwater flow. Therefore, based on the obtained outcomes, the choice of model will mostly depend on the scale of the problem and the processes involved. Accordingly, `darcyInterTransportFoam` is best suited for large-scale hydrological simulations and exploratory studies, where computational efficiency is a priority and fine-scale processes near the riverbed are less critical. On the other hand, `interFoam` is more appropriate for small-scale applications focused on riverbed biogeochemistry or detailed hydrodynamics, where accurately capturing fine-scale transport and reaction processes is essential, despite the higher computational demands. Further analysis under transient conditions, along with field observations, will still be needed to thoroughly assess the strengths and limitations of each modelling approach with respect to complex SW-GW interactions. Yet, despite its limitations, this study helps to further assess the utility of `darcyInterTransportFoam` for investigating high spatiotemporal SW-GW exchanges and the potential of `interFoam` for studying these interactions beyond the local-scale.

- Chapter 4 -- **Code Developments: Solver Upgrades and OpenFOAM Dictionary Generation Scripts:** the clarity and reproducibility of both `darcyInterTransportFoam` new features and the **MATLAB** script set for **OpenFOAM** dictionary generation are enhanced by the descriptions provided in this chapter. Code snippets and the explicit identification of key input parameters support both comprehension and code navigation, thereby reinforcing the link between theory and application.

## 5.2. Limitations and Recommendations for Future Research

Despite the generally satisfactory outcomes, several limitations associated with the conducted research should be highlighted. They serve as a basis for recommendations for future work.

Firstly, despite the evolution of `darcyInterTransportFoam` over `hyporheicScalarInterFoam`, the current version of the newly developed solver still exhibits some modelling limitations that prevent its application across all types of SW-GW scenarios. Among the remaining constraints, the inability to simulate unsaturated flow is particularly relevant, as it restricts the solver's applicability to situations where the groundwater table movement is assumed negligible compared to surface water fluctuations. Consequently, further code development will be required to introduce new modelling strategies–such as the approach used in `groundwaterFoam` (Horgue, et al.

2022) for unsaturated flow—that address the still existing gaps. These model advancements will allow a more comprehensive simulation of the dynamic and biogeochemical processes occurring within SW-GW systems, regardless of their hydrological or physical characteristics.

The limited access to computational resources also represented a constraint on efficiently running transient simulations. All test cases in this PhD thesis were run in parallel on a local High-Performance Computing (HPC) cluster comprising 320 CPUs equally distributed across 10 nodes. However, the resource availability was frequently restricted by the number of users and their respective computational demands, meaning fewer resources were often accessible. As a result, quasi-steady simulations were chosen to assess the capability of both `darcyInterTransportFoam` and `interFoam` in resolving continuous SW-GW interactions. While the steady-state results are considered sufficient to demonstrate the utility of both modelling approaches for untangling such bidirectional exchanges, further assessment under transient conditions remains necessary to fully evaluate the strengths and limitations of each approach.

On the other hand, the flat learning curve of OpenFOAM required a significant time investment at the beginning of this work. Unlike other CFD software, OpenFOAM lacks an integrated GUI, has limited documentation and involves a complex structure, all of which required an additional effort that delayed the initial stages of this PhD thesis. These aspects should be considered when selecting the software to implement a structured workflow that optimizes the learning process as much as possible.

The reduced availability of field observations posed a significant limitation in both studies described in this PhD thesis. In Chapter 2, the difficulty of acquiring high-resolution data on site reduced the number of the observations available for comparison with the model outcomes. In Chapter 3, the synthetic nature of the simulated system precluded the acquisition of real data to validate the simulated results. Consequently, further analysis incorporating in-situ observations will still be needed to better evaluate the efficacy of `darcyInterTransportFoam` in capturing continuous SW-GW flux exchanges.

## 5.3. Conclusions

The development of increasingly complex numerical solvers, such as `darcyInterTransportFoam`, prompts a broader reflection on the role of such tools in advancing our understanding of SW-GW systems. While the capacity to account for intricate subsurface properties and simulate dynamically interlinked processes, such as flow, solute transport, and heat transfer, is a significant technical achievement, it raises an important question: should such detailed codes be used to represent real-world field

sites in their full complexity, or are they better suited to gain insights into fundamental process interactions through more exploratory, synthetic modelling exercises?

Synthetic modelling offers a powerful framework to dissect and investigate the interplay among hydrological, transport and thermal processes under controlled conditions. These explorative simulations help identify key mechanisms, feedbacks and scale-dependent behaviours that may not be apparent from field data alone. Insights gained from such models can then inform the design of field campaigns, guide data acquisition strategies and refine conceptual models.

However, the latest advances in computational science (e.g., high-performance computing, cloud computing and GPU acceleration) and field observation techniques (e.g., remote sensing, geophysical methods and sensor technologies) are rapidly narrowing the gap between synthetic abstraction and real-world complexity. As data availability and quality continue to improve, and as modelling frameworks become more adaptable and integrated, the development of high-fidelity, site-specific numerical representations of field systems is becoming increasingly attainable. For instance, the parametrization and calibration of numerical models have advanced significantly in recent years, driven by the growing incorporation of innovative field observation data into the simulations through data assimilation and highly parameterized, regularized inversion techniques (Schilling, et al., 2019). Therefore, rather than choosing between simplicity and realism, future efforts may focus on hybrid approaches that leverage complex models for both mechanistic insight and practical prediction of coupled SW-GW dynamic processes.

This thesis has contributed to this goal by introducing a new fully-coupled solver designed not only to advance process-based understanding through synthetic experiments, but also to provide the technical capabilities required to represent real-world hydrological systems in detail within one of the most advanced CFD simulation environments. This is achieved through its ability to capture the complex subsurface characteristics of natural systems and the diverse dynamic processes they drive. In doing so, it highlights the critical need for more efficient and accurate methods to characterize subsurface properties and processes. Improving these approaches is essential not only for enhancing model reliability, but also for better understanding and predicting the complex dynamics that govern SW-GW interactions.

# References

[1] Deshpande, S. S., Anumolu, L., & Trujillo, M. F. (2012). Evaluating the performance of the two-phase flow solver interFoam. *Computational Science & Discovery, 5*(1), 014016. https://doi.org/10.1088/1749-4699/5/1/014016

[2] Horgue, P., Renard, F., Gerlero, G., Guibert, R., & Debenest, G. (2022). porousMultiphaseFoam v2107: An open-source tool for modeling saturated/unsaturated water flows and solute transfers at watershed scale. *Computer Physics Communications, 273*, 108278. https://doi.org/10.1016/j.cpc.2021.108278

[3] Lee, A., Aubeneau, A., Liu, X., & Cardenas, M. B. (2021). Hyporheic Exchange in Sand Dunes Under a Freely Deforming River Water Surface. *Water Resources Research, 57*(3), e2020WR028817. https://doi.org/10.1029/2020WR028817

[4] Pardo-Álvarez, Á. (2025). *darcyInterTransportFoam v1.0.* [Software], Zenodo. https://doi.org/10.5281/zenodo.15857142

[5] Schilling, O. S., Cook, P. G., & Brunner, P. (2019). Beyond Classical Observations in Hydrogeology: The Advantages of Including Exchange Flux, Temperature, Tracer Concentration, Residence Time, and Soil Moisture Observations in Groundwater Model Calibration. *Reviews of Geophysics, 57*(1), 146-182. https://doi.org/10.1029/2018RG000619

[6] Weller, H. G., Tabor, G., Jasak, H., & Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics, 12*(6), 620-631. https://doi.org/10.1063/1.168744

# A

# The OpenFOAM Software

# A.1. Software History and Variants

The continuum mechanics library FOAM was initiated in the late ‘80s, early ‘90s by Henry Weller at the Imperial College London (Weller, et al., 1998). Rather than using FORTRAN, the most popular, standard programming language of the time, Weller decided to use C++ to develop a powerful and flexible CFD simulation platform. From this initiation to the founding of the so-called Nabla Ltd company, Henry Weller and Hrvoje Jasak took care of the basic development and maintenance of the software for almost a decade. Throughout this period, FOAM was commercially sold by Nabla Ltd until the 10th of December 2004, when the software was released under the GNU General Public License Version 3 (GPLv3) and renamed to **OpenFOAM**. Thereafter, the development and maintenance of **OpenFOAM** went through different steps and experienced several changes in the process that derived in a number of software branches. A brief introduction to its freely available main variants is provided as follows:

1. **OpenFOAM®**: immediately after Nabla Ltd was dissolved in 2004, Henry Weller, Chris Greenshields and Mattijs Janssens founded OpenCFD Ltd to manage and develop **OpenFOAM**. In 2007, this company also became the owner of the **OpenFOAM** Trademark (**OpenFOAM®**). For a period of almost seven years, OpenCFD Ltd was in charge of the release of all versions of **OpenFOAM**. This fact changes the 8th of August 2011, when the company transferred the IP rights to the US foundation "**OpenFOAM** Foundation, inc.". After that date, OpenCFD Ltd. continued to maintain and develop **OpenFOAM**, whereas The **OpenFOAM** Foundation Ltd. took care of the software releases. One year later, on the 12th of September 2012, the ESI Group acquired OpenCFD Ltd, becoming a wholly-owned subsidiary of the ESI Group. The forementioned managing strategy of the software remained unaltered until the 25th of April 2014, when The **OpenFOAM** Foundation Ltd. was incorporated as a Company Limited by Guarantee (CLG) in England. After that, the software IP was transferred to the UK and the US entity was dissolved. Furthermore, diverse changes in the company governance resulted in Weller and Greenshields leaving OpenCFD. Thereafter, OpenCFD Ltd. has directly released its version of **OpenFOAM** from January 2016 to the present day with a date-of-release identifier (e.g., v2406) (OpenCFD Ltd, 2024). The source code repository can be found on GitLab.
2. **foam-extend**: in parallel to the foundation of OpenCFD Ltd, Hrvoje Jasak created in 2004 the consulting company Wikki Ltd following the successful proprietary development of FOAM at Nabla Ltd. Among the different projects carried out by the company, it is the **foam-extend** project, a fork of the **OpenFOAM®** developed as a community effort and, therefore, not endorsed by any company. The goal of this project is to open the **OpenFOAM** toolbox to contributions from all users and developers. In this way, **foam-extend** contains bug fixes and performance

improvements, as well as and additional features provided by community contributors. The software is mainly maintained and released by Wikki Ltd, being its source code freely accessible at Source Forge.

3. **OpenFOAM**: this **OpenFOAM** variant has been released by The **OpenFOAM** Foundation Ltd. since 2011 with a sequence-based identifier (e.g., 11) (openfoam.org). However, its development and maintenance are managed by CFD Direct Ltd. only since 2015, after Henry Weller and Chris Greenshields left OpenCFD Ltd. and founded this company. The software is developed by a team of individuals led by Henry Weller, Director of both CFD Direct and The **OpenFOAM** Foundation, who contribute their work to the project with the support and consent of the companies that employ them. The source code is openly hosted on GitHub.

In the present thesis, the variant **OpenFOAM®** (hereinafter simply called **OpenFOAM**) was selected to both develop the new fully-coupled solver and conduct the different test cases.

## A.2. Case Structure

**OpenFOAM** cases consist of three directories, each of which containing several plain text input files. The basic directory structure to run a case is described below. Likewise, a representative case example is shown in **Figure A.1**. More detailed information on the case organization and content can be found in the online **OpenFOAM** User's Guide (OpenCFD Ltd, 2023):

- The **<initial time>** directory contains the data files of the solver-specific fields. The data stored in these files corresponds to the initial and boundary condition (BC) values specified for the start of the simulation. As for the name of the directory, this is based on the simulated time at which the data is written. Therefore, since simulations typically start at time 0, the initial values are usually stored in a directory named 0:
  - *<field data files>*: there will be one data file for each field required by the solver being run (e.g., `U`, `p_rgh`, `alpha.water`, etc.).
- The **constant** directory contains the files to define the physics of the problem, including the mesh and any physical property that is required for the solver. The following are the subdirectories and files that can be found within this directory:
  - *polyMesh* subdirectory: contains several files that provide a full description of the mesh. These files are not written by the user but generated after pre-processing the mesh.
  - *<other files>*: the most common are *transportProperties,* to specify necessary physical properties, and *turbulenceProperties,* to select the turbulence model to be

used and its associated parameters. Another typical file included in this dictionary is *fvOptions,* which can also be located in the *system* directory. *fvOptions* provides a collection of runtime selectable finite volume options to manipulate systems of equations by adding sources/sinks, imposing constraints, and applying corrections. Additional files may be required depending on the solver being used.

- *<other subdirectories>*: such as *triSurface*, where to place the triangulated surfaces (.stl, .obj, .vtk, etc.) needed to define the geometry to mesh or to select mesh regions for refinement or field assignment.

➢ The **system** directory contains the files to set the parameters associated with the simulation control and solution procedure. Among all the files potentially included within this dictionary, only three are mandatory: *controlDict*, *fvSchemes*, and *fvSolution*. The rest are optional and depend on the user's purpose:

- *controlDict* file: sets the main simulation controls. This includes, e.g., timing information, parameters for data output and optional libraries and function objects that can be respectively loaded and executed at runtime.
- *fvSchemes* file: specifies the numerical schemes used to discretize the partial differential equations.
- *fvSolution* file: sets for the run the linear solvers, tolerances and other algorithm controls used to solve the discretized equations.
- *<other files>:* an extensive number of optional files can be added to the *system* directory. Among the most popular are *blockMeshDict*, *decomposeParDict and setFieldsDict*. The former defines the mesh parameters if the `blockMesh` utility is selected to generate the mesh. *decomposeParDict* specifies the mesh decomposition method and the number of subdomains in case of parallel computation. As for *setFieldsDict,* it prescribes field values on selected sets of cells/patch-faces. Other files that may be included within this dictionary are those aimed to control runtime post-processing activities and those used to select specific sets of the mesh for further applications.
- *<other subdirectories>*: analogously to the optional files, a number of not mandatory subdirectories can also be found within the *system* directory. These subdirectories may be solver-specific or user-defined (e.g., grouping all the post-processing function objects).

Further directories are generated depending on user's selection. These may include:

➢ The **<result time>** directories: additional time directories that correspond to the solution fields obtained at selected times. Analogously to the **<initial time>** directory, the name of these directories will be the simulation time at which the solution is output. The frequency of this output is controlled via the *controlDict* file.

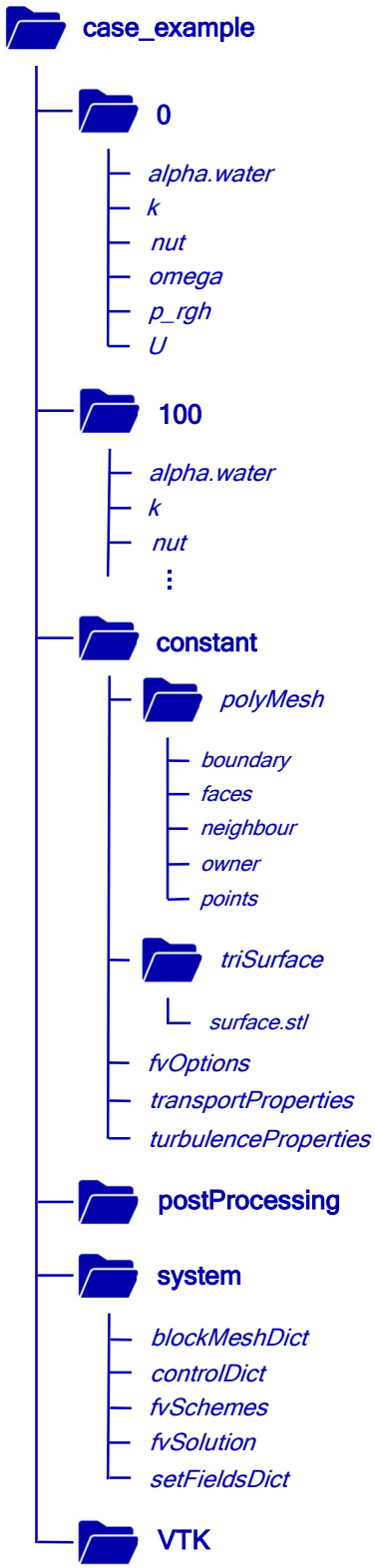



Figure A.1. Representative case example of the OpenFOAM directory structure.

- The **postProcessing** directory: contains the output data files typically generated by function objects (e.g., time data at specific cross sections or boundaries, point or surface field data sampling, field average values, etc.). Each data type will be placed inside its corresponding subdirectory.
- The **processor<number>** directories: contain the field data (time directory) and mesh information (*constant* directory) files corresponding to each decomposed subdomain of the mesh. These subdirectories are only generated if the case is run in parallel.
- The **<data conversion>** directory: contains the output data converted into file formats supported by third-party post-processors, such as **ParaView**, **EnSight**, **Fluent**, etc. The name of the resulting directory will depend on the selected post-processing format. In this way, e.g., the conversion from **OpenFOAM** data into .vtk format will export the transformed files into a *VTK* subdirectory. However, things will be different if the original files are converted into **Fluent** format. In this case, the resulting subdirectory will be named *fluentInterface* and will include two types of converted files: one for the mesh (.msh) and another for the results (.dat).

## A.3. Model Setup

After an overall description of the **OpenFOAM** case structure, a short overview of the entire model setup is provided as follows. The setup of an **OpenFOAM** model can be split into three main steps:

1. **Pre-processing**: generation of the computational mesh and definition of the initial and boundary conditions.

The first step of the model setup consists in the creation of the mesh where to solve the relevant equations (**Figure A.2**). The mesh is an integral part of the numerical simulation and, therefore, must satisfy certain criteria to ensure an accurate solution. In this way, **OpenFOAM** checks during any run that the mesh satisfies a strict set of validity constraints. If these constraints are not satisfied, it will stop running. This practice sometimes complicates the correction of big meshes generated by third-party mesh generators but guarantees that the mesh is valid for the computation. Else, the solution is flawed before the simulation has even started. “The development of a high-quality, three-dimensional mesh is a complex and time-consuming process, but the effect of the mesh’s quality on the simulation results is significant and should not be underestimated” (Teuber, 2020).

By default, **OpenFOAM** defines an unstructured polyhedral mesh in 3D, known as a *polyMesh*. The cells of this mesh can be formed by an unlimited number of faces delimited, at the same time, by an unlimited number of edges with no restriction on its

alignment. This sort of mesh offers great flexibility in mesh generation and manipulation, especially in those cases where the geometry of the domain is complex or changes over time (e.g., a riverbed). However, such freedom in mesh generation can make difficult the conversion of meshes generated with other pre-processing tools, where only some existing polyhedral cell shapes are normally supported. To overcome this, OpenFOAM provides tools (utilities) to convert third-party mesh formats into the *polyMesh* format based on sets of pre-defined 3D cell geometries (e.g., hexahedrons, prims, wedges, etc.). Further details on the OpenFOAM mesh conversion utilities can be found in the User's Guide (OpenCFD Ltd, 2023).

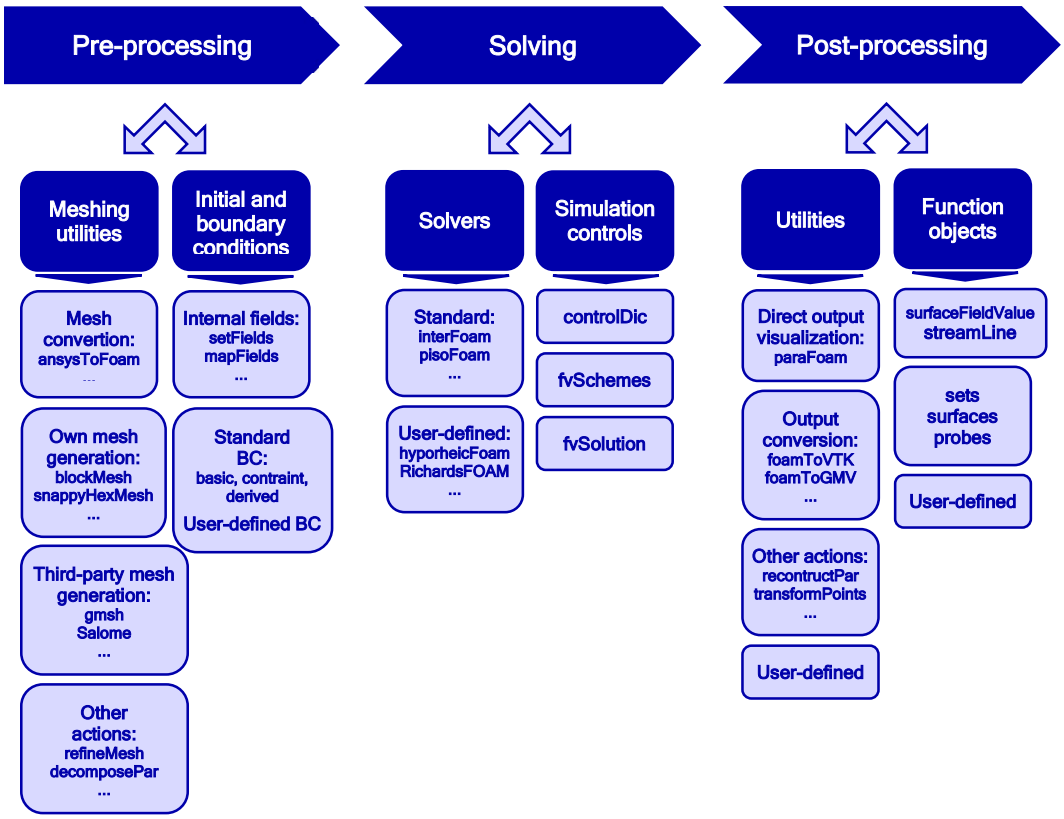



Figure A.2. Overview of the OpenFOAM model setup.

As regards the own mesh generation, the OpenFOAM library contains several utilities aimed at creating the computational mesh where to run the numerical simulation. Structured meshes can be obtained with `blockMesh`. This utility defines the mesh geometry based on a set of 3D hexahedral blocks, whose edges can be straight lines, arcs, or splines. Each of these blocks is specified by 8 vertices, the number of cells in each direction and the cell size grading. `snappyHexMesh` is another meshing utility supplied by OpenFOAM. The `snappyHexMesh` utility generates 3D meshes containing hexahedra and split-hexahedra by iteratively refining a starting mesh (structured or unstructured) and conforming the resulting split-hex mesh to a triangulated surface geometry in Stereolithography (.stl) format. Other mesh generators are the `makeFaMesh`

utility, which creates finite-area meshes from volume-mesh patches, and **cfMesh** (cfmesh.com), a free, third-party mesh generator compatible with all recent versions of **OpenFOAM®** and **foam-extend**. The latter already exports the generated 3D meshes in **OpenFOAM** format, so no posterior conversion is needed. A detailed explanation of all the mesh generation utilities is provided in the online **OpenFOAM** User's Guide (OpenCFD Ltd, 2023). Further third-party meshing tools, such as **gmsh** or **Salome**, also exist and come with different advantages and disadvantages (Kortelainen, 2009). Some of them already deliver the resulting meshes in **OpenFOAM** format, whereas others will require later conversion in order to be used for the simulation. Additional meshing actions, such as mesh refinement, point translation/rotation, or even domain decomposition for parallel computing, are also part of the pre-processing step. Utilities for each of these actions are included in the **OpenFOAM** library, namely **`refineMesh`**, **`transformPoints`** and **`decomposePar`**, respectively.

For the second part of preprocessing, initial and boundary conditions need to be defined for each variable being solved (**Figure A.2**). **OpenFOAM** contains a number of utilities designed for such purposes. In this way, internal field values can be prescribed for the start of the simulation using utilities such as **`setFields`** or **`mapFields`**. The former will set the values on a selected set of cells/patch-faces, whereas the latter will map the fields from another mesh (e.g., belonging to a previous simulation). Good initial conditions can lead to a faster model convergence, so their definition is highly recommended. As for the BCs, it is important to know first that in **OpenFOAM** mesh boundaries are treated as *patches*. A boundary can consist of a single patch but also be broken up into a set of patches (e.g., an inlet boundary split into two patches: one for the air and another for the water). Furthermore, one patch can include one or more areas of any boundary, which do not necessarily need to be physically connected (e.g., right, and left bank boundaries of a river merged into the same patch). There exist several base patch types: `patch`, `symmetryPlane`, `empty`, `wedge`, `cyclic`, `wall` and `processor`. These types are specified both in the mesh and field files of an **OpenFOAM** case. Their extended description can be found in the **OpenFOAM** User's Guide. Among all of them, `patch` is the basic patch type. In contrast to the rest (except for `wall`), it does not contain geometric information about the mesh and, therefore, it is simply used to specify the numerical condition at the boundary. In **OpenFOAM**, the BCs are organized into three categories: *basic*, *constraint* and *derived*. The former contains Dirichlet and Neumann conditions but also variations combining both. The *constraint* category includes all the patch types other than `patch` and `wall`. Finally, the *derived* class comprises **OpenFOAM**-specialized conditions. The boundary conditions contained in each category can be checked in the **OpenFOAM** Doxygen. User-defined BCs may also be used at those cases where the standard conditions do not adjust to the goals of the simulation. Setting appropriate BCs for each simulated field is crucial to achieve a successful

simulation, as ill-posed BCs will lead to physically incorrect solutions and, in many cases, solver failure.

2. **Solving**: numerical simulation of the problem.

Once the computational mesh is tailored to the quality criteria and the initial and boundary conditions are set, the model is ready to start the simulation.

As previously exposed, **OpenFOAM** includes a large range of application solvers, to which an important set of user-defined models have to be added (Figure A.2). Each of these solvers is designed to tackle a specific class of problem in continuum mechanics. The solver choice will thus depend each time on the particular problem to be solved. This selection will primarily establish the equations and algorithms used for the computation, as they differ from one solver to another, and ultimately determine most of the parameters and physical properties required to define the simulation case. Additional modelling parameters will be specified at runtime, if necessary, via optional dictionary files located in the *constant* directory of the case.

However, the simulation setup involves not only the selection of the application solver but also the definition of the options to control the time and output behaviour, the choice of numerical schemes and linear-solvers, as well as the selection of the debug switch to monitor the solution progress (Figure A.2). All the previous modelling features are specified across several dictionary files situated in the *system* directory of the simulation case. A short introduction to these files, namely *controlDict*, *fvSchemes* and *fvSolution*, is provided in Section A.2 of this manuscript, while the **OpenFOAM** User's Guide (OpenCFD Ltd, 2023) supplies an extended and detailed description of each of them.

Regarding the application solvers, **OpenFOAM** classifies them by category of continuum mechanics. Among all the available categories, the solvers included within the incompressible and multiphase flow classes are particularly interesting to approach hydrogeological processes. Common solvers used to study surface water-groundwater dynamics are `pimpleFoam` (incompressible flow) or `interFoam` (multiphase flow), in combination with different turbulence models (i.e., RANS, LES or DES) and function objects. The latter allows the solvers to expand their simulation capabilities and, therefore, to solve processes beyond flow, such as solute transport, particle tracking, etc. Likewise, user-defined solvers also found application in hydrogeology, being them mainly used to study SW-GW interactions (e.g., `hyporheicFoam`).

3. **Post-processing**: data conversion and data output.

The last step of the model setup consists in the post-processing of the simulation outputs. **OpenFOAM** results are post-processed through different data utilities and function objects (Figure A.2), which extract the data at user-specified times. Whereas the majority of utilities are aimed to convert the **OpenFOAM** format of the output field files to

another format compatible with a third-party visualization software (e.g., **ParaView**, **EnSight**, etc.), the function objects allow to extract field data at specific mesh locations (i.e., point, patch, inner face, mesh region, etc.). The file format and content resulting from the latter will depend on the function object selected as well as on the user's choice. Examples of function object output files are 1D time lists of average/total field values in ASCII format or 2D/3D field surfaces in STL format.

Among all the available post-processing utilities, `paraFoam` stands out above all. This utility launches ParaView by default using the reader module supplied with **OpenFOAM**. **ParaView** is an open-source visualization software and the main post-processing tool included in the **OpenFOAM** library. `paraFoam` allows to perform several post-processing actions without the need to convert the output data into another format. These actions include the visualization of the mesh and output fields even at decomposed cases. Alternatively, utilities such as `foamToVTK` or `foamToTecplot360` allow to convert the output data into other formats for its posterior post-processing in third-party graphical programs.

Moreover, additional standard utilities, like `reconstructPar` –which enables the reconstruction of decomposed fields in a parallel case– or user-implemented utilities may also be applied to post-process the simulation outcomes if required. Further details about the data post-processing can be found in the **OpenFOAM** User's Guide (OpenCFD Ltd, 2023).

# References


[1] Kortelainen, J. (2009). *Meshing Tools for Open Source CFD: A Practical Point of View.* Espoo: VTT Technical Research Centre of Finland. VTT Research Report, No. VTT-R-02440-09

[2] OpenCFD Ltd. (2023, December). *OpenFOAM® v2312 User Guide*. Retrieved December 25, 2024, from https://doc.openfoam.com/2312/

[3] OpenCFD Ltd. (2024, June). OpenFOAM® v2406. OpenFOAM Download [code]. Retrieved from https://dl.openfoam.com/source/v2406/

[4] Teuber, K. (2020). *A three-dimensional two-phase model for flow, transport and mass transfer processes in sewers.* PhD Thesis, Technischen Universität Berlin, Water Resources Management and Modeling of Hydrosystems, Berlin, Germany. https://doi.org/10.14279/depositonce-9576

[5] Weller, H. G., Tabor, G., Jasak, H., & Fureby, C. (1998). A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics, 12*(6), 620-631. https://doi.org/10.1063/1.168744

# B

# Supplementary Material for Chapter 2

# B.1. Extended Description of the Field Site

The river segments selected to conduct the two test cases described in Chapter 2 are located along a 1.8 km section of the Emme River (Figure B.1). This river section is placed within the so-called Emme Site, one of the highly instrumented observatories belonging to the Centre of Hydrogeology and Geothermics (CHYN) of the University of Neuchâtel (UniNE). The site lies in the valley of upper Emme River, a pre-alpine, alluvial valley situated at the northern margin of the Swiss Alps, in the region of Emmental (Canton of Bern, Switzerland). This alluvial plain is 22 km long and covers an approximate area of 6 $km^2$, including the valley of the Roethebach tributary (Käser & Hunkeler, 2016). The underlying aquifer is formed by quaternary deposits, composed of sandy gravels (80% gravel and 20% sand) and cobbles, with a variable proportion of silt, as well as lenses of coarse uniform gravel (Popp, et al., 2021; Schilling, et al., 2022). These features result in an extremely conductive, unconfined, alluvial aquifer (Blau & Muchenberger, 1997). The alluvium is limited underneath by well-cemented Tertiary sandstone and conglomerates, as well as marls, forming a bedrock of several hundred meters thick (Käser & Hunkeler, 2016). This bedrock is considered, together with the lower layer of the alluvium (1–3 m thick), as an aquitard relative to the rest of the aquifer (Blau & Muchenberger, 1997). Hydrogeologically, prior pumping test studies revealed higher saturated hydraulic conductivity values (~$2 \cdot 10^{-3}$ m/s) predominantly in the upper 30-35 m of the aquifer, whereas smaller values (~$10^{-4}$ m/s or even lower) shown mainly deeper (Blau & Muchenberger, 1997; Schilling, et al., 2017a; Würsten, 1991). Regarding porosity, values between 0.1 and 0.3 were found based on the same previous in-situ field observations.

The studied domains are located in a river section near an important groundwater abstraction plant (Schilling, et al., 2017a). On the Ramsei plain in Aeschau, groundwater is pumped in roughly equal parts from eight suction wells spaced at approximately 100 m and aligned in parallel to the Emme River (Figure B.1). The total abstraction rate is on average 24000 l/min (Popp, et al., 2021). The water supplied from this station represents approximately 45% of the total drinking water consumed in the Swiss capital Bern and its surroundings (Biaggi, et al., 2005). It is due to this interesting setting that the aquifer is well documented and monitored in the area (Blanc, et al., 2024; Peel, et al., 2022). In this way, the monitoring network consists in nearly 50 piezometers in the proximities of the wellfield, spread over both sides of the river. At this location, the valley has a width of 200–400 m and a mean slope of 0.9%, whereas the aquifer thickness is 25 m on average and 46 m at the most (Popp, et al., 2021; Würsten, 1991). Furthermore, analogously to the aquifer, the Emme River exhibits a coarse gravel bed and, additionally, two couples of small weirs designed for erosion control (Figure B.2). Several studies have been conducted in the last few years at the site to evaluate the impact of groundwater pumping on the general water balance of the system. Beyond

demonstrating the effect of abstraction in the total flow out of the valley, the results obtained provide evidence of substantial river-groundwater interactions across the riverbed (e.g., Blanc, et al., 2024; Käser & Hunkeler, 2016; Peel, et al., 2022; Popp, et al., 2021; Schilling, et al., 2017a; Schilling, et al., 2022; Tang, et al., 2018). Some of these studies also indicated that the magnitude of the exchanges has a direct impact on groundwater storage, stating that the infiltration from the river represents the main source of recharge of the aquifer. This contribution from the river could amount up to 90% of total recharge, as suggested by Poffet (2011).

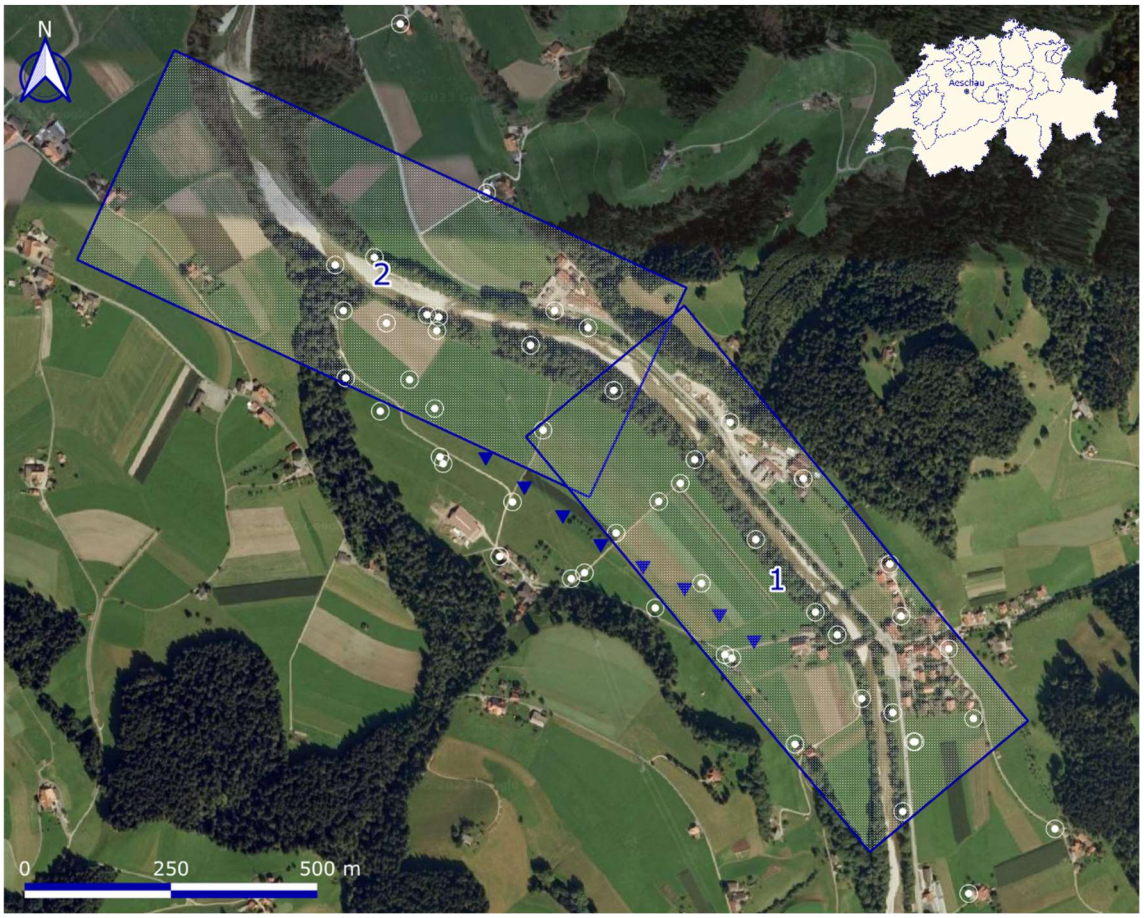



**Figure B.1. Emme field site near Aeschau: pumping wells (blue triangular markers) and piezometers (white circular markers) on the Ramsei plain and its surroundings.**

Regarding the climatic and hydraulic conditions of the site, the average annual precipitation is 1300 mm, the potential evapotranspiration 550 mm and mean annual air temperature 8 °C (Figura, et al., 2013; Käser & Hunkeler, 2016). The long-term average discharge of the river is 4.4 $m^3/s$ (Schilling, et al., 2022), being this variable highly dynamic over the year (Popp, et al., 2021). In this way, the discharge is typically higher during the snowmelt periods (March-April) and especially small in very dry summers and cold winters (Schilling, et al., 2017a). During the latest, segments of the Emme may run completely dry (Würsten, 1991). These dry periods have become more often and

prolonged in the last few years (Michel, et al., 2020) and previous studies have already shown that this tendency will continue in the future as a result of the increasing air temperature (Addor, et al., 2014). Consequently, groundwater recharge patterns will be inevitably altered, potentially affecting the quantity and quality of the drinking water production in the area (Hock, et al., 2019).


**Figure B.2. Respectively, upstream, and downstream flight view of the Emme River at the site (boxes 1 and 2 in Figure B.1). Photos taken on November 2014 with a drone.**

## B.2. Boundary Conditions

The whole set of boundary conditions defined for both simulation test cases of Chapter 2 can be checked in Table B.1, Table B.2 and Table B.3. Likewise, a small description of each of them is provided as follows:

- `totalPressure`: sets the static pressure field ($p_{sur,i}$) at the boundary by subtracting the dynamic pressure to the user-specified stagnation pressure ($p_{0_{sur,i}}$).
- `fixedFluxPressure`: prescribes at the patch the pressure gradient resulting from the specified velocity BC.
- `noSlip`: only applicable to vector fields. It fixes the variable components (e.g. velocity) to zero at the patch. It represents an alternative to the zero `fixedValue` BC. There is no difference between them.
- `slip`: applicable to both vector and scalar fields. In the case of the latter, it works analogously to the `zeroGradient` BC, whereas for a vector field it erases the normal component of the variable at the patch and keeps the tangential components untouched.

- **fixedValue**: provides a specified, constant value at the patch. This boundary condition corresponds the so-called first-type, Dirichlet BC.
- **pressureInletOutletVelocity**: mixed BC assigned to the velocity field ($U_{sur,i}$) and governed by its coupled pressure condition. A **zeroGradient** or a **fixedValue** BC will be applied at the patch depending on the direction of the flux (in or out of the domain).
- **flowRateInletVelocity**: derives from the specified flow rate a uniform velocity field normal to the patch.
- **inletOutlet**: mixed BC. The mode of operation, either **zeroGradient** or **fixedValue**, is determined by the flux direction across the patch (in or out of the domain). A specified inflow must be defined for the case of return flow.
- **zeroGradient**: extrapolates to the patch the internal field values from the nearest cells, so that their gradient is equal to zero in direction perpendicular to the patch.
- **calculated**: the value of this boundary condition is obtained via field assignment, meaning, its value is derived from other variables assigned at the patch.
- **nutkRoughWallFunction**: provides a wall-function for the turbulent kinematic viscosity field ($\nu_{turb,i}$) based on the turbulent kinetic energy field ($k_i$) and the specific energy dissipation field ($\omega_i$) to account for the roughness effects caused at walls (e.g., the riverbed).
- **kqRWallFunction**: in the present study cases, provides a simple wrapper around the zero-gradient condition for the turbulent kinetic energy field ($k_i$). It is not a wall-function condition.
- **omegaWallFunction**: in the present study cases, provides a wall-function for the specific energy dissipation field ($\omega_i$) for low- and high-Reynolds number flows.
- **surInterfaceSolute**: user-defined mixed BC aimed to be used with **darcyInterTransportFoam** at the interface patch of the surface domain. Analogously to other standard mixed boundary conditions, the operation switch of this BC relies on the flux direction. In this way, if the flux sign is negative (inflow), the solute concentration at the interface patch of the subsurface domain is assigned to the boundary. On the contrary, if the flux sign is positive (outflow), a **zeroGradient** condition is prescribed to the patch, so that the solute drains conceptually out of the domain.
- **surInterfaceHeat**: equal to **surInterfaceSolute** but used to compute heat transfer instead of solute transport.
- **subInterfaceSolute**: equivalent to **surInterfaceSolute** but assigned to the interface patch of the subsurface domain. Thus, if the flux points inwards at the selected patch, the solute concentration from the analogous surface patch is

mapped to the boundary. Else, if the flux points outwards, a `zeroGradient` condition is applied.

- `subInterfaceHeat`: the same as `subInterfaceSolute` but used to compute heat transfer instead of solute transport.
- `darcyGradHead` and `surfaceHydrostaticHead`: user-defined BCs. A detailed description of each of them can be found in Section 4.2.4 of this manuscript.

More detailed information on each of the above BC can be found in the online OpenFOAM User's Guide (OpenCFD Ltd, 2023).

**Table B.1.** Initial (IC) and boundary conditions (BC) defined for each surface variable. The purple-coloured BC are test case specific.

| | | p_rgh ($p_{rgh,i}$) | U ($U_{sur,i}$) | alpha.water ($\alpha_i$) | C ($C_{sur,i}$) | T ($T_{sur,i}$) |
|---|---|---|---|---|---|---|
| Dimensions | | [1 -1 -2 0 0 0 0] (kg/m·s²) | [0 1 -1 0 0 0 0] (m/s) | [0 0 0 0 0 0 0] (-) | [1 -3 0 0 0 0 0] (kg/m³) | [0 0 0 1 0 0 0] (°C) |
| Initial conditions | | uniform 0.001 | uniform (0 0 0) | uniform 0 | uniform 0 | uniform -2 |
| Boundaries | *inletAir* | **totalPressure**<br>p0 uniform 0<br>value uniform 0 | **pressureInletOutletVelocity**<br>value uniform (0 0 0) | **inletOutlet**<br>inletValue uniform 0<br>value uniform 0 | **inletOutlet**<br>inletValue uniform 0<br>value uniform 0 | **inletOutlet**<br>inletValue uniform -2<br>value uniform -2 |
| | *inletWater* | **fixedFluxPressure**<br>value uniform 0 | **flowRateInletVelocity**<br>volumetricFlowRate 10<br>value uniform (0 0 0) | **fixedValue**<br>value uniform 1 | **fixedValue**<br>value uniform 5 | **fixedValue**<br>value uniform 2 |
| | *waterJet* | **fixedFluxPressure**<br>value uniform 0 | **flowRateInletVelocity**<br>volumetricFlowRate 0.42<br>value uniform (0 0 0) | **fixedValue**<br>value uniform 1 | **fixedValue**<br>value uniform 10 | **fixedValue**<br>value uniform 8 |
| | *banks* | **slip** | **slip** | **slip** | **slip** | **slip** |
| | *riverbed* | **fixedFluxPressure**<br>value uniform 0 | **fixedValue**<br>value uniform (0 0 0) | **zeroGradient** | **surInterfaceSolute**<br>value uniform 0 | **surInterfaceHeat**<br>value uniform -2 |
| | *steps* | **fixedFluxPressure**<br>value uniform 0 | **noSlip** | **zeroGradient** | **zeroGradient** | **zeroGradient** |
| | *atmosphere* | **totalPressure**<br>p0 uniform 0<br>value uniform 0 | **pressureInletOutletVelocity**<br>value uniform (0 0 0) | **inletOutlet**<br>inletValue uniform 0<br>value uniform 0 | **inletOutlet**<br>inletValue uniform 0<br>value uniform 0 | **inletOutlet**<br>inletValue uniform -2<br>value uniform -2 |
| | *outlet* | **fixedValue**<br>value uniform 0 | **pressureInletOutletVelocity**<br>value uniform (0 0 0) | **zeroGradient** | **zeroGradient** | **zeroGradient** |

Table B.2. Initial (IC) and boundary conditions (BC) defined for each kOmegaSST turbulence model variable. The purple-coloured BC are test case specific.

| | nut ($\nu_{turb,i}$) | k ($k_i$) | omega ($\omega_i$) |
|---|---|---|---|
| Dimensions | [0 2 -1 0 0 0 0] ($m^2/s$) | [0 2 -2 0 0 0 0] ($m^2/s^2$) | [0 0 -1 0 0 0 0] (1/s) |
| Initial conditions | uniform 0.05 | uniform 0.2 | uniform 4.5 |
| Boundaries | | | |
| *inletAir* | **calculated**<br>value uniform 0.05 | **inletOutlet**<br>inletValue uniform 0.2<br>value uniform 0.2 | **inletOutlet**<br>inletValue uniform 4.5<br>value uniform 4.5 |
| *inletWater* | **calculated**<br>value uniform 0.05 | **fixedValue**<br>value uniform 0.1 | **fixedValue**<br>value uniform 9 |
| *waterJet* | **calculated**<br>value uniform 0.05 | **fixedValue**<br>value uniform 0.1 | **fixedValue**<br>value uniform 9 |
| *banks* | **slip** | **slip** | **slip** |
| *riverbed* | **nutkRoughWallFunction**<br>Cs uniform 0.5<br>Ks uniform 0.03<br>value uniform 0.05 | **kqRWallFunction**<br>value uniform 0.2 | **omegaWallFunction**<br>value uniform 4.5 |
| *steps* | **nutkRoughWallFunction**<br>Cs uniform 0.5<br>Ks uniform 0.03<br>value uniform 0.05 | **kqRWallFunction**<br>value uniform 0.2 | **omegaWallFunction**<br>value uniform 4.5 |
| *atmosphere* | **calculated**<br>value uniform 0.05 | **inletOutlet**<br>inletValue uniform 0.2<br>value uniform 0.2 | **inletOutlet**<br>inletValue uniform 4.5<br>value uniform 4.5 |
| *outlet* | **calculated**<br>value uniform 0.05 | **zeroGradient** | **zeroGradient** |

Table B.3. Initial (IC) and boundary conditions (BC) defined for each subsurface variable. The purple-coloured BC boundary condition is test case specific.

| | | h ($h_{sub,i}$) | q ($q_i$) | C ($C_{sub,i}$) | T ($T_{sub,i}$) |
|---|---|---|---|---|---|
| Dimensions | | [0 1 0 0 0 0 0] (m) | [0 1 -1 0 0 0 0] (m/s) | [1 -3 0 0 0 0 0] (kg/m$^3$) | [0 0 0 1 0 0 0] (°C) |
| Initial conditions | | uniform 1 | uniform (-1e-05 1.5e-05 0) | uniform 10 | uniform 8 |
| Boundaries | *inlet* | **darcyGradHead** value uniform 1 | **fixedValue** value uniform (-1e-05 1.5e-05 0) | **fixedValue** value uniform 10 | **fixedValue** value uniform 8 |
| | *banks* | **slip** | **slip** | **slip** | **slip** |
| | *aquifer* | **slip** | **slip** | **slip** | **slip** |
| | *riverbed* | **fixedValue** value uniform 0 | **calculated** value uniform (-1e-05 1.5e-05 0) | **subInterfaceSolute** value uniform 10 | **subInterfaceHeat** value uniform 8 |
| | *steps* | **zeroGradient** | **calculated** value uniform (-1e-05 1.5e-05 0) | **zeroGradient** | **zeroGradient** |
| | *outlet* | **surfaceHydrostaticHead** value uniform 1 | **calculated** value uniform (-1e-05 1.5e-05 0) | **zeroGradient** | **zeroGradient** |

# References

[1] Addor, N., Rössler, O., Köplin, N., Huss, M., Weingartner, R., & Seibert, J. (2014). Robust changes and sources of uncertainty in the projected hydrological regimes of Swiss catchments. *Water Resources Research, 50*(10), 7541-7562. https://doi.org/10.1002/2014WR015549

[2] Biaggi, D., Polack, P., & Backman, N. (2005). *Grundwasserfassungen Aeschau-Gesuch um Konzessionserneuereung-Fachbericht Hydrologie/Hydrogeologie.* Technical Report, Geotechnisches Institut, Bern, Switzerland.

[3] Blanc, T., Peel, M., Brennwald, M. S., Kipfer, R., & Brunner, P. (2024). Efficient injection of gas tracers into rivers: A tool to study Surface water-Groundwater interactions. *Water Research, 254*, 121375. https://doi.org/10.1016/j.watres.2024.121375

[4] Blau, R. V., & Muchenberger, F. (1997). *Grundlagen für Schutz und Bewirtschaftung der Grundwasser des Kantons Bern: Nutzungs-, Schutz- und Überwachungskonzept für die Grundwasserleiter des obersten Emmentals, zwischen Emmenmatt, Langnau und Eggiwil.* Technical Report, Wasser- und Energiewirtschaft des Kantons Bern, Bern, Switzerland.

[5] Figura, S., Livingstone, D. M., & Kipfer, R. (2013). Competing controls on groundwater oxygen concentrations revealed in multidecadal time series from riverbank filtration sites. *Water Resources Research, 49*(11), 7411-7426. https://doi.org/10.1002/2013WR013750

[6] Hock, R., Rasul, G., Adler, C., Cáceres, B., Gruber, S., & Hirabayashi, Y. (2019). *Chapter 2: High mountain areas (Tech. Rep.). In H.-O. Pörtner, D. C. Roberts, V. Masson-Delmotte, P. Zhai, M. Tignor, E. Poloczanska.* PCC special report on the ocean and cryosphere in a changing climate.

[7] Käser, D., & Hunkeler, D. (2016). Contribution of alluvial groundwater to the outflow of mountainous catchments. *Water Resources Research, 52*, 680-697. https://doi.org/10.1002/2014WR016730

[8] Michel, A., Brauchli, T., Lehning, M., Schaefli, B., & Huwald, H. (2020). Stream temperature and discharge evolution in Switzerland over the last 50 years: annual and seasonal behaviour. *Hydrology and Earth System Sciences, 24*(1), 115-142. https://doi.org/10.5194/hess-24-115-2020

[9] OpenCFD Ltd. (2023, December). *OpenFOAM® v2312 User Guide*. Retrieved December 25, 2024, from https://doc.openfoam.com/2312/

[10] Peel, M., Kipfer, R., Hunkeler, D., & Brunner, P. (2022). Variable 222Rn emanation rates in an alluvial aquifer: Limits on using 222Rn as a tracer of surface water -

Groundwater interactions. *Chemical Geology, 599*, 120829. https://doi.org/10.1016/j.chemgeo.2022.120829

[11] Poffet, D. (2011). *Interactions nappe-rivière et stockage dans l'aquifère de la Haute-Emme: Approche par la modélisation numérique.* MSc Thesis, University of Neuchâtel, Centre for Hydrogeology and Geothermics, Neuchâtel, Switzerland.

[12] Popp, A. L., Pardo-Álvarez, Á., Schilling, O. S., Scheidegger, A., Musy, S., Peel, M., Brunner, P., Purtschert, R., Hunkeler, D., & Kipfer, R. (2021). A Framework for Untangling Transient Groundwater Mixing and Travel Times. *Water Resources Research, 57*(4), e2020WR028362. https://doi.org/10.1029/2020WR028362

[13] Schilling, O. S., Gerber, C., Partington, D. J., Purtschert, R., Brennwald, M. S., Kipfer, R., Hunkeler, D., & Brunner, P. (2017a). Advancing Physically-Based Flow Simulations of Alluvial Systems Through Atmospheric Noble Gases and the Novel 37Ar Tracer Method. *Water Resources Research, 53*(12), 10465-10490. https://doi.org/10.1002/2017WR020754

[14] Schilling, O. S., Partington, D. J., Doherty, J., Kipfer, R., Hunkeler, D., & Brunner, P. (2022). Buried Paleo-Channel Detection With a Groundwater Model, Tracer-Based Observations, and Spatially Varying, Preferred Anisotropy Pilot Point Calibration. *Geophysical Research Letters, 49*(14), e2022GL098944. https://doi.org/10.1029/2022GL098944

[15] Tang, Q., Schilling, O. S., Kurtz, W., Brunner, P., Vereecken, H., & Hendricks Franssen, H.-J. (2018). Simulating Flood-Induced Riverbed Transience Using Unmanned Aerial Vehicles, Physically Based Hydrological Modeling, and the Ensemble Kalman Filter. *Water Resources Research, 54*(11), 9342-9363. https://doi.org/10.1029/2018WR023067

[16] Würsten, M. (1991). *GWB–Hydrogeologische Untersuchungen Aeschau: Schlussbericht.* Technical Report, Geotechnisches Institut, Zürich, Switzerland.

# C

# Supplementary Material for Chapter 3

# C.1. Boundary Conditions of Both Model Cases

The complete collection of boundary conditions (BC) used in the two model cases of Chapter 3 is provided in Table C.1 to Table C.5. The selected BC set closely matches those applied to both test cases described in Chapter 2. Consequently, only the newly specified BC are described below. Descriptions of the remaining BC can be found in Section B.2:

- `cyclicAMI`: type of cyclic boundary condition (commonly referred to as periodic BC) used for Arbitrary Mesh Interface (AMI) handling. It enables the seamless connection of two non-conformal mesh patches (e.g., differing in shape, cell alignment or resolution) that must exchange information consistently during a simulation. The continuity of physical fields (e.g., velocity, pressure) is ensured through data interpolation between the patches.

Further details on each applied BC can be obtained from the **OpenFOAM** User's Guide (OpenCFD Ltd, 2023).

Table C.1. `darcyInterTransportFoam` case: initial (IC) and boundary conditions (BC) defined for each surface variable.

| | | p_rgh ($p_{rgh,i}$) | U ($U_{sur,i}$) | alpha.water ($\alpha_i$) | C ($C_{sur,i}$) |
|---|---|---|---|---|---|
| Dimensions | | [1 -1 -2 0 0 0 0] (kg/m·s²) | [0 1 -1 0 0 0 0] (m/s) | [0 0 0 0 0 0 0] (-) | [1 -3 0 0 0 0 0] (kg/m³) |
| Initial conditions | | uniform 0.001 | uniform (0 0 0) | uniform 0 | uniform 0 |
| Boundaries | *inletAir* | **totalPressure**<br>p0 uniform 0<br>value uniform 0 | **pressureInletOutletVelocity**<br>value uniform (0 0 0) | **inletOutlet**<br>inletValue uniform 0<br>value uniform 0 | **inletOutlet**<br>inletValue uniform 0<br>value uniform 0 |
| | *inletWater* | **fixedFluxPressure**<br>value uniform 0 | **flowRateInletVelocity**<br>volumetricFlowRate 2<br>value uniform (0 0 0) | **fixedValue**<br>value uniform 1 | **fixedValue**<br>value uniform 2 |
| | *banks* | **slip** | **slip** | **slip** | **slip** |
| | *riverbed* | **fixedFluxPressure**<br>value uniform 0 | **fixedValue**<br>value uniform (0 0 0) | **zeroGradient** | **surInterfaceSolute**<br>value uniform 0 |
| | *atmosphere* | **totalPressure**<br>p0 uniform 0<br>value uniform 0 | **pressureInletOutletVelocity**<br>value uniform (0 0 0) | **inletOutlet**<br>inletValue uniform 0<br>value uniform 0 | **inletOutlet**<br>inletValue uniform 0<br>value uniform 0 |
| | *outlet* | **fixedValue**<br>value uniform 0 | **pressureInletOutletVelocity**<br>value uniform (0 0 0) | **zeroGradient** | **zeroGradient** |

Table C.2. `darcyInterTransportFoam` case: initial (IC) and boundary conditions (BC) defined for each kOmegaSST turbulence model variable.

| | nut ($\nu_{turb,i}$) | k ($k_i$) | omega ($\omega_i$) |
|---|---|---|---|
| Dimensions | [0 2 -1 0 0 0 0] ($m^2/s$) | [0 2 -2 0 0 0 0] ($m^2/s^2$) | [0 0 -1 0 0 0 0] (1/s) |
| Initial conditions | uniform 0.05 | uniform 0.2 | uniform 4.5 |
| Boundaries *inletAir* | **calculated**<br>value uniform 0.05 | **inletOutlet**<br>inletValue uniform 0.2<br>value uniform 0.2 | **inletOutlet**<br>inletValue uniform 4.5<br>value uniform 4.5 |
| Boundaries *inletWater* | **calculated**<br>value uniform 0.05 | **fixedValue**<br>value uniform 0.1 | **fixedValue**<br>value uniform 9 |
| Boundaries *banks* | **slip** | **slip** | **slip** |
| Boundaries *riverbed* | **nutkRoughWallFunction**<br>Cs uniform 0.5<br>Ks uniform 0.03<br>value uniform 0.05 | **kqRWallFunction**<br>value uniform 0.2 | **omegaWallFunction**<br>value uniform 4.5 |
| Boundaries *atmosphere* | **calculated**<br>value uniform 0.05 | **inletOutlet**<br>inletValue uniform 0.2<br>value uniform 0.2 | **inletOutlet**<br>inletValue uniform 4.5<br>value uniform 4.5 |
| Boundaries *outlet* | **calculated**<br>value uniform 0.05 | **zeroGradient** | **zeroGradient** |

Table C.3. `darcyInterTransportFoam` case: initial (IC) and boundary conditions (BC) defined for each subsurface variable.

| | | h ($h_{sub,i}$) | C ($C_{sub,i}$) |
|---|---|---|---|
| | **Dimensions** | [0 1 0 0 0 0 0] (m) | [1 -3 0 0 0 0 0] (kg/m$^3$) |
| | **Initial conditions** | uniform 1 | uniform 4 |
| **Boundaries** | ***inlet*** | **slip** | **slip** |
| | ***banks*** | **slip** | **slip** |
| | ***aquifer*** | **slip** | **slip** |
| | ***riverbed*** | **fixedValue**<br>value uniform 0 | **subInterfaceSolute**<br>value uniform 4 |
| | ***outlet*** | **slip** | **slip** |

Table C.4. `interFoam` case: initial (IC) and boundary conditions (BC) defined for each surface/subsurface variable.

| | | p_rgh ($p_{rgh,i}$) | U ($U_{sur,i}$) | alpha.water ($\alpha_i$) | C ($C_{sur,i}$) |
|---|---|---|---|---|---|
| Dimensions | | [1 -1 -2 0 0 0 0] (kg/m·s²) | [0 1 -1 0 0 0 0] (m/s) | [0 0 0 0 0 0 0] (-) | [1 -3 0 0 0 0 0] (kg/m³) |
| Initial conditions | | uniform 0.001 | uniform (0 0 0) | uniform 0 | uniform 0 |
| Boundaries | *inletAir* | **totalPressure**<br>p0 uniform 0<br>value uniform 0 | **pressureInletOutletVelocity**<br>value uniform (0 0 0) | **inletOutlet**<br>inletValue uniform 0<br>value uniform 0 | **inletOutlet**<br>inletValue uniform 0<br>value uniform 0 |
| | *inletWater* | **fixedFluxPressure**<br>value uniform 0 | **flowRateInletVelocity**<br>volumetricFlowRate 2<br>value uniform (0 0 0) | **fixedValue**<br>value uniform 1 | **fixedValue**<br>value uniform 2 |
| | *inlet_sub* | **slip** | **slip** | **slip** | **slip** |
| | *banks* | **slip** | **slip** | **slip** | **slip** |
| | *riverbed_sur* | **cyclicAMI**<br>value uniform 0.001 | **cyclicAMI**<br>value uniform (0 0 0) | **cyclicAMI**<br>value uniform 0 | **cyclicAMI**<br>value uniform 0 |
| | *riverbed_sub* | **cyclicAMI**<br>value uniform 0.001 | **cyclicAMI**<br>value uniform (0 0 0) | **cyclicAMI**<br>value uniform 0 | **cyclicAMI**<br>value uniform 0 |
| | *aquifer* | **slip** | **slip** | **slip** | **slip** |
| | *atmosphere* | **totalPressure**<br>p0 uniform 0<br>value uniform 0 | **pressureInletOutletVelocity**<br>value uniform (0 0 0) | **inletOutlet**<br>inletValue uniform 0<br>value uniform 0 | **inletOutlet**<br>inletValue uniform 0<br>value uniform 0 |
| | *outlet* | **fixedValue**<br>value uniform 0 | **pressureInletOutletVelocity**<br>value uniform (0 0 0) | **zeroGradient** | **zeroGradient** |
| | *outlet_sub* | **slip** | **slip** | **slip** | **slip** |

Table C.5. `interFoam` case: initial (IC) and boundary conditions (BC) defined for each kOmegaSST turbulence model variable.

| | | nut ($\nu_{turb,i}$) | k ($k_i$) | omega ($\omega_i$) |
|---|---|---|---|---|
| Dimensions | | [0 2 -1 0 0 0 0] ($m^2/s$) | [0 2 -2 0 0 0 0] ($m^2/s^2$) | [0 0 -1 0 0 0 0] (1/s) |
| Initial conditions | | uniform 0.05 | uniform 0.2 | uniform 4.5 |
| Boundaries | *inletAir* | **calculated**<br>value uniform 0.05 | **inletOutlet**<br>inletValue uniform 0.2<br>value uniform 0.2 | **inletOutlet**<br>inletValue uniform 4.5<br>value uniform 4.5 |
| | *inletWater* | **calculated**<br>value uniform 0.05 | **fixedValue**<br>value uniform 0.1 | **fixedValue**<br>value uniform 9 |
| | *inlet_sub* | **slip** | **slip** | **slip** |
| | *banks* | **slip** | **slip** | **slip** |
| | *riverbed_sur* | **cyclicAMI**<br>value uniform 0.05 | **cyclicAMI**<br>value uniform 0.2 | **cyclicAMI**<br>value uniform 4.5 |
| | *riverbed_sub* | **cyclicAMI**<br>value uniform 0.05 | **cyclicAMI**<br>value uniform 0.2 | **cyclicAMI**<br>value uniform 4.5 |
| | *aquifer* | **slip** | **slip** | **slip** |
| | *atmosphere* | **calculated**<br>value uniform 0.05 | **inletOutlet**<br>inletValue uniform 0.2<br>value uniform 0.2 | **inletOutlet**<br>inletValue uniform 4.5<br>value uniform 4.5 |
| | *outlet* | **calculated**<br>value uniform 0.05 | **zeroGradient** | **zeroGradient** |
| | *outlet_sub* | **slip** | **slip** | **slip** |

# C.2. Additional Flow and Solute Results

## C.2.1. Flowpaths

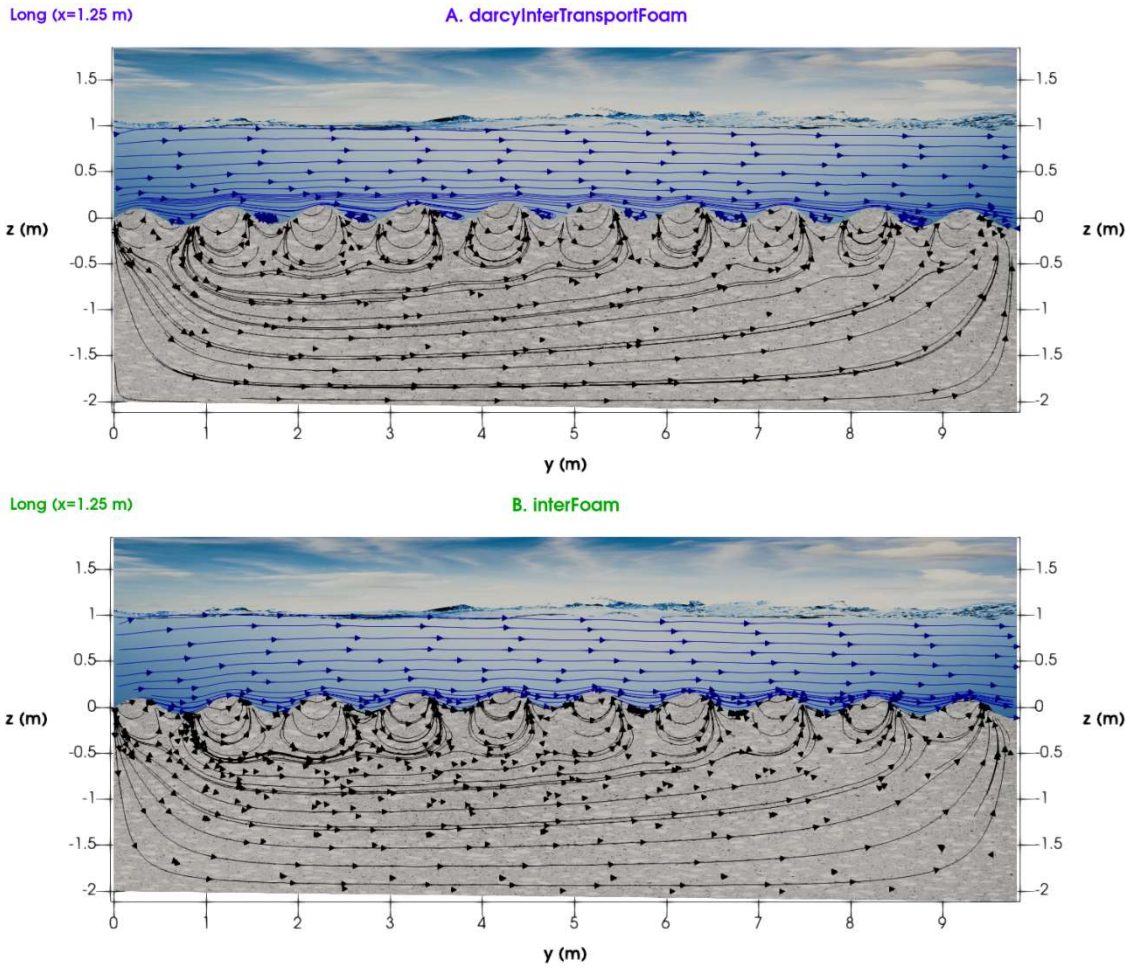



Figure C.1. Surface water and groundwater flowpaths across the longitudinal slice at x=1.25 m.

## C.2.2. Flow Velocity

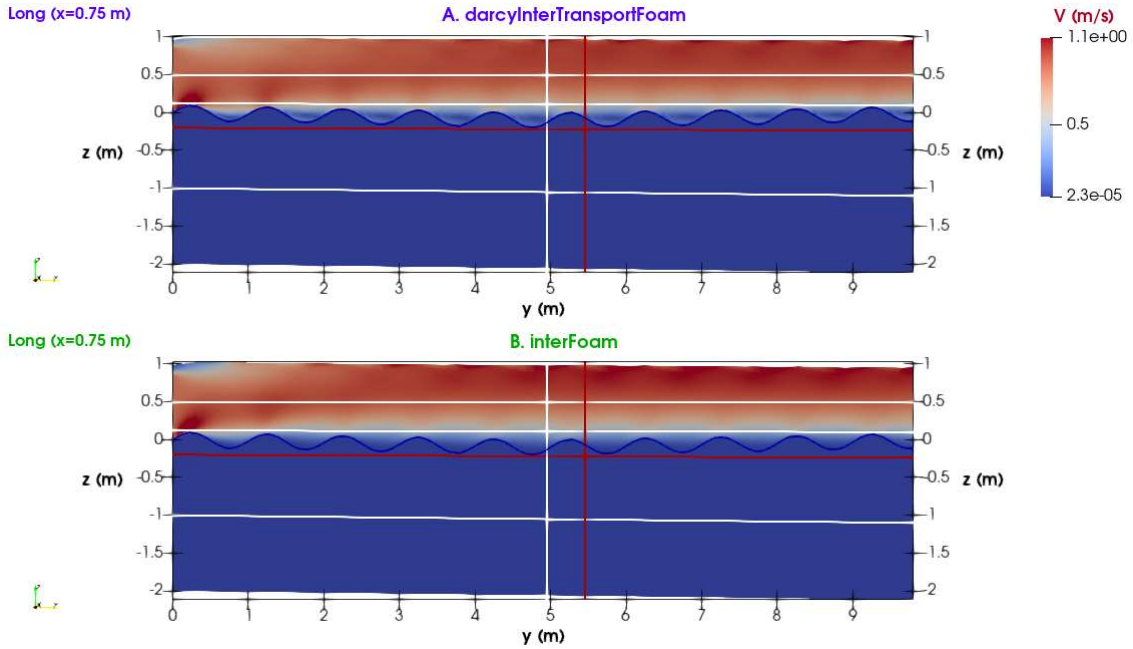



Figure C.2. Flow velocity field (V) across the longitudinal slice at x=0.75 m. The red and white lines indicate the transversal and horizontal slices that intersect it (Figure 3.4).

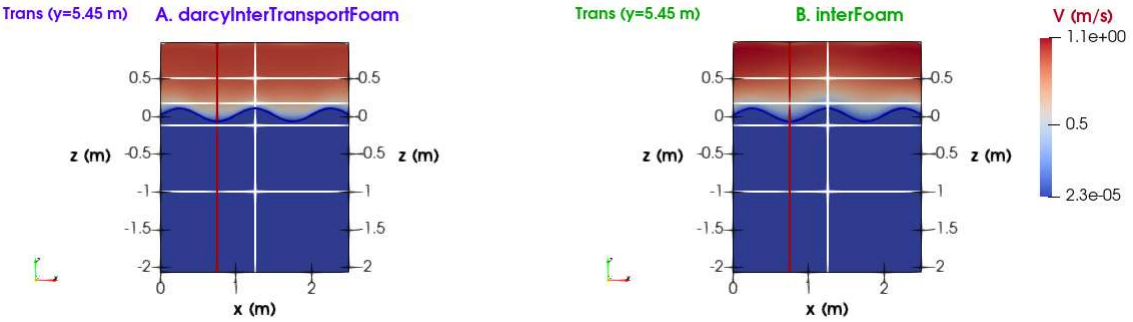
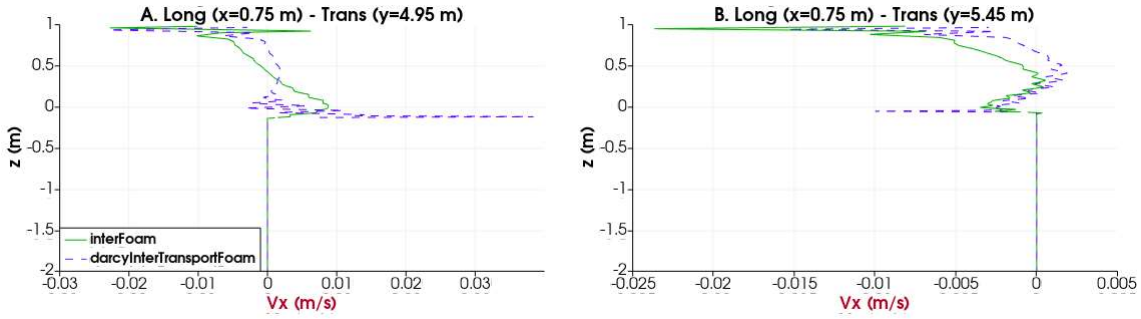



Figure C.3. Flow velocity field (V) across the traversal slice at y=5.45 m. The red and white lines indicate the longitudinal and horizontal slices that intersect it (Figure 3.4).



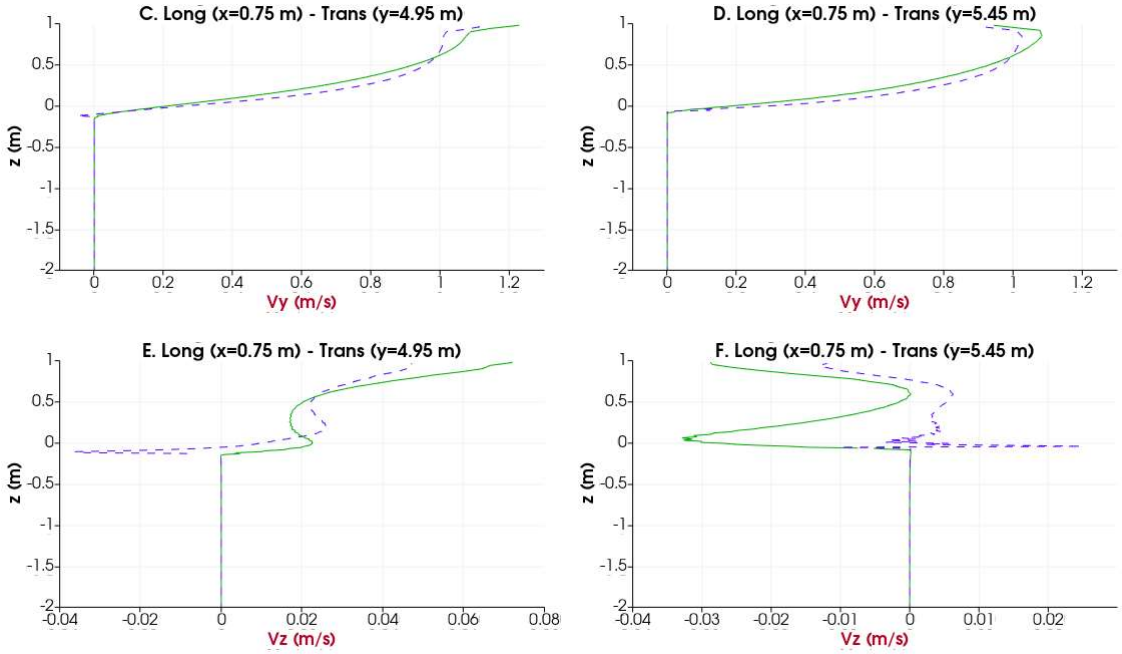



Figure C.4. Transversal (A, B), longitudinal (C, D) and vertical (E, F) flow velocities (Vx, Vy and Vz, respectively) along the intersections of the longitudinal slice at x=0.75 m and the transversal slices at y=4.95 m and y=5.45 m.

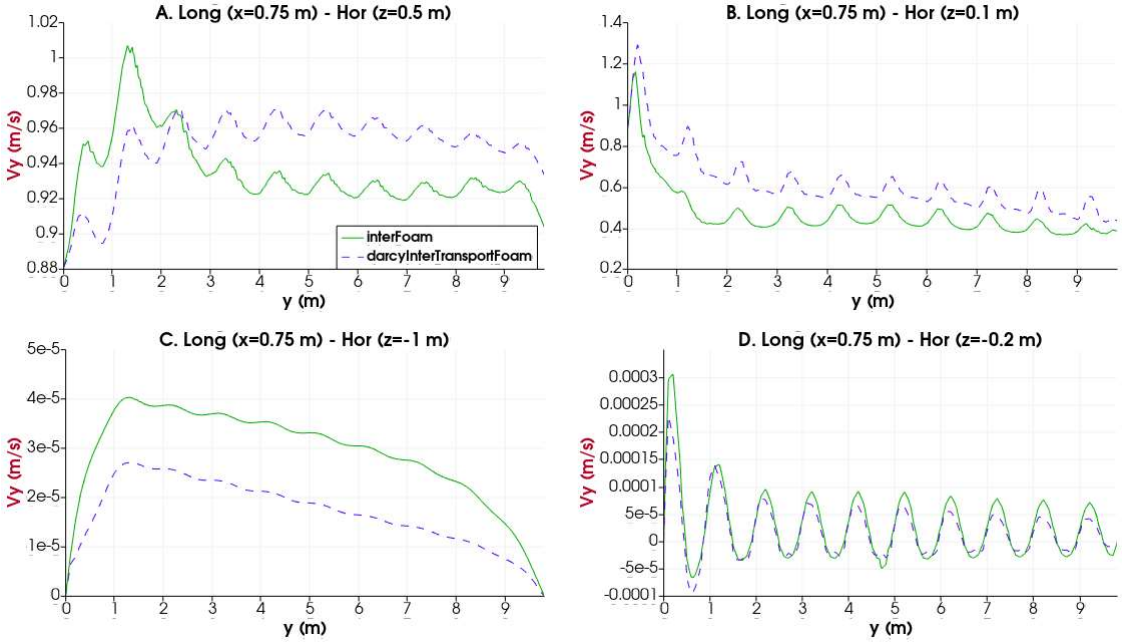



Figure C.5. Longitudinal flow velocities ($V_y$) along the intersections of the various horizontal slices with the longitudinal slice at x=0.75 m.

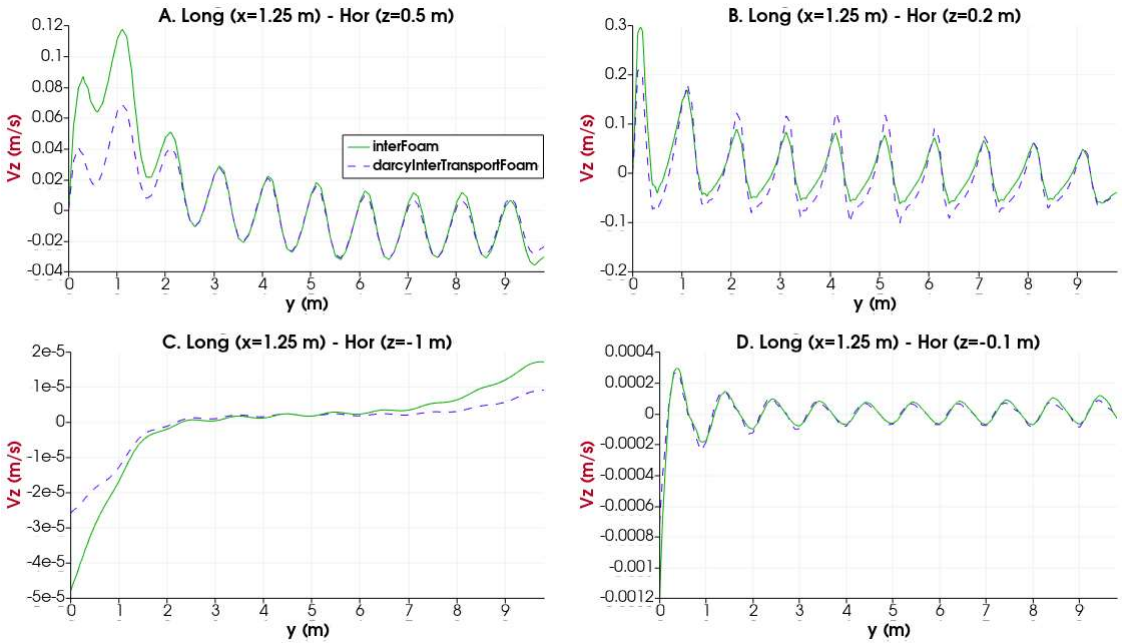



Figure C.6. Vertical flow velocities ($V_z$) along the intersections of the various horizontal slices with the longitudinal slice at x=1.25 m.

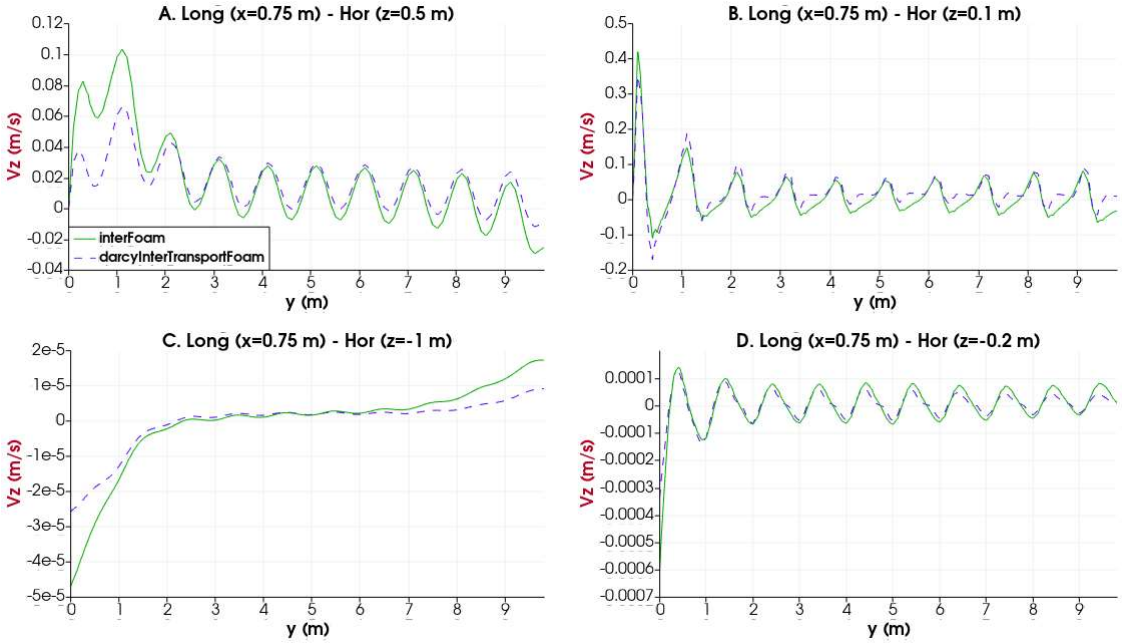



Figure C.7. Vertical flow velocities ($V_z$) along the intersections of the various horizontal slices with the longitudinal slice at x=0.75 m.

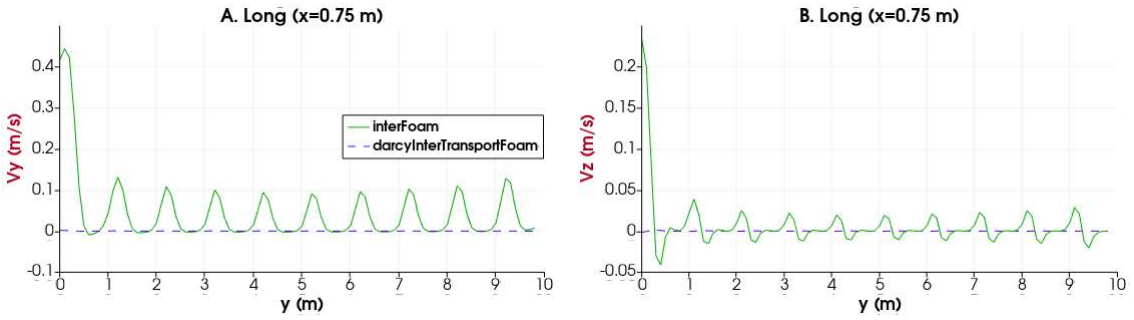



Figure C.8. Flow velocities ($V_y$ and $V_z$) along the intersection of the longitudinal slice at x=0.75 m with the riverbed.

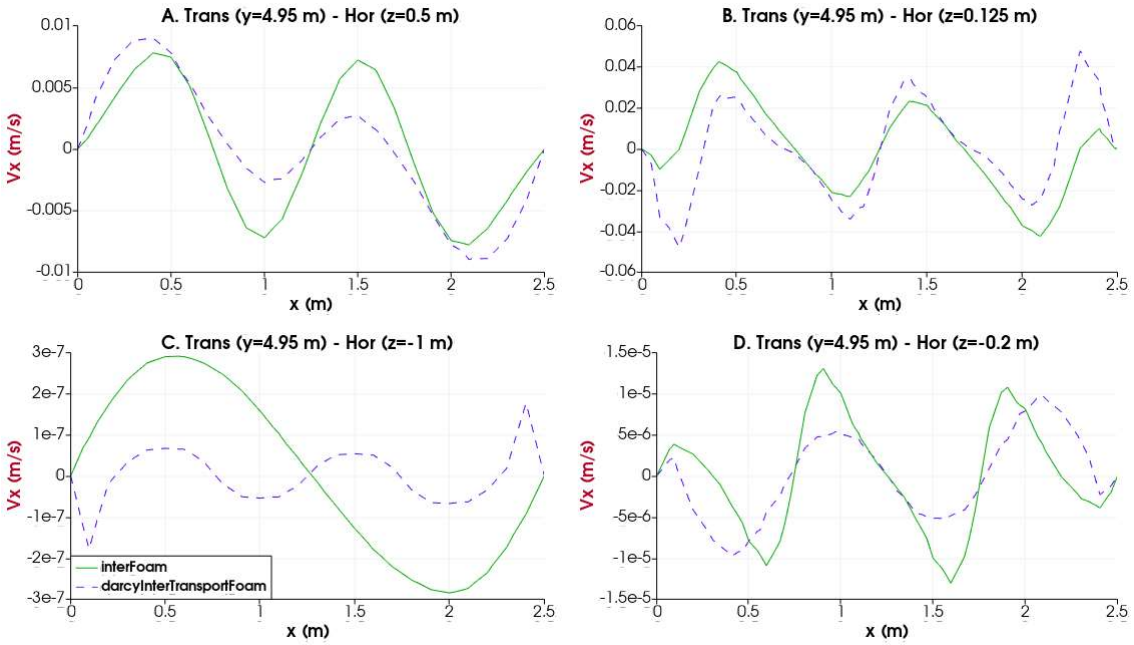



Figure C.9. Transversal flow velocities ($V_x$) along the intersections of the various horizontal slices with the transversal slice at y=4.95 m.

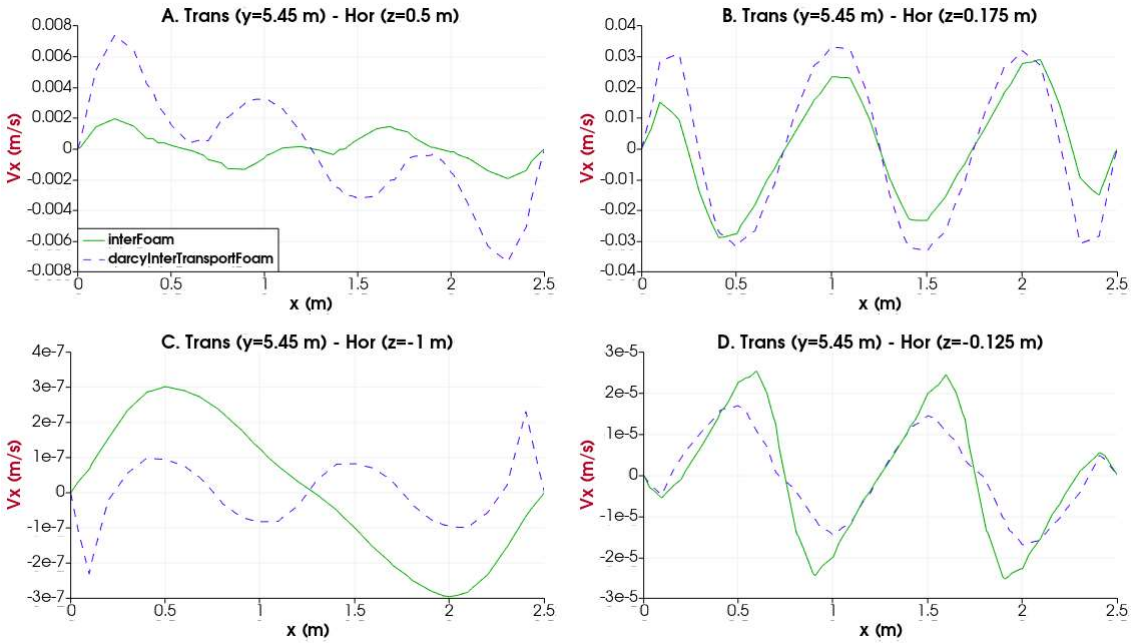



Figure C.10. Transversal flow velocities ($V_x$) along the intersections of the various horizontal slices with the transversal slice at y=5.45 m.

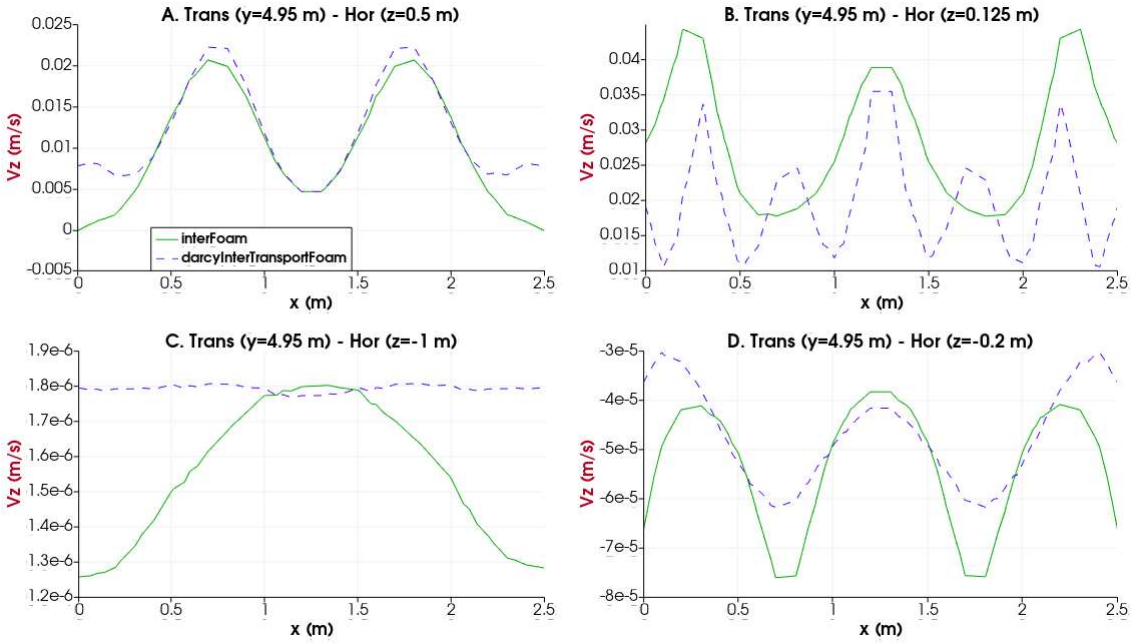



Figure C.11. Vertical flow velocities ($V_z$) along the intersections of the various horizontal slices with the transversal slice at y=4.95 m.

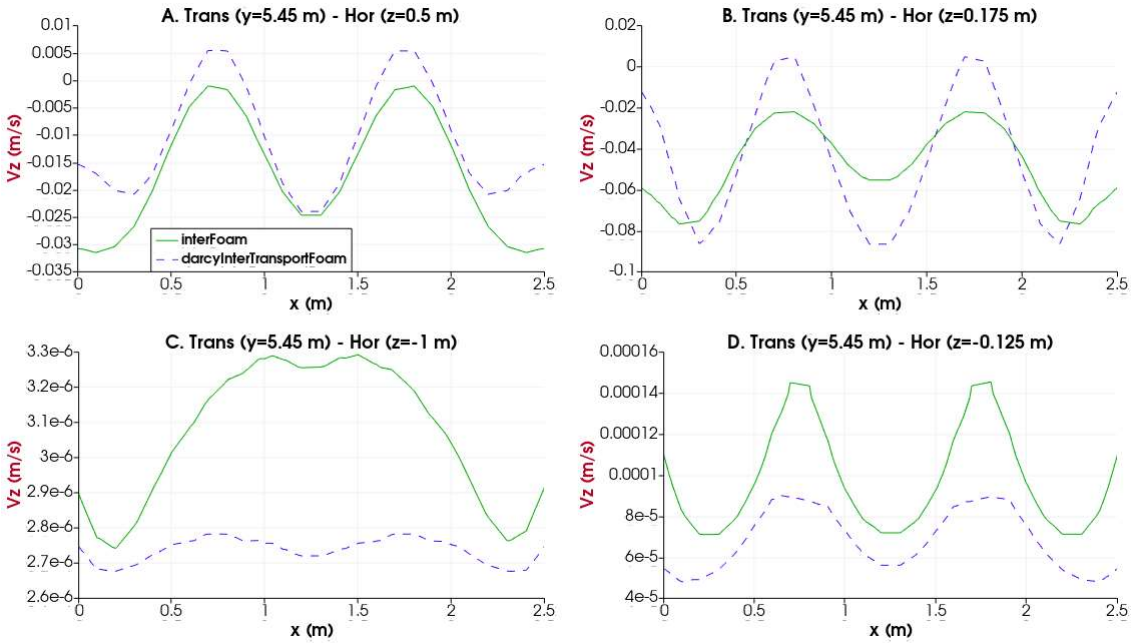



Figure C.12. Vertical flow velocities ($V_z$) along the intersections of the various horizontal slices with the transversal slice at y=5.45 m.

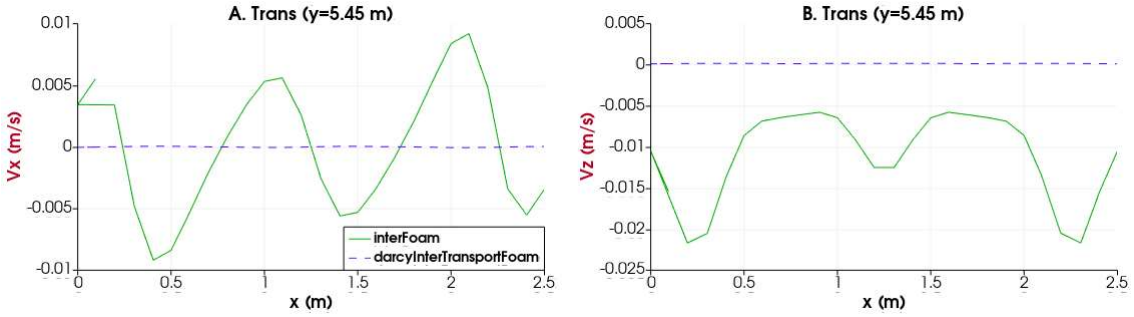



Figure C.13. Flow velocities ($V_x$ and $V_z$) along the intersection of the transversal slice at x=5.45 m with the riverbed.

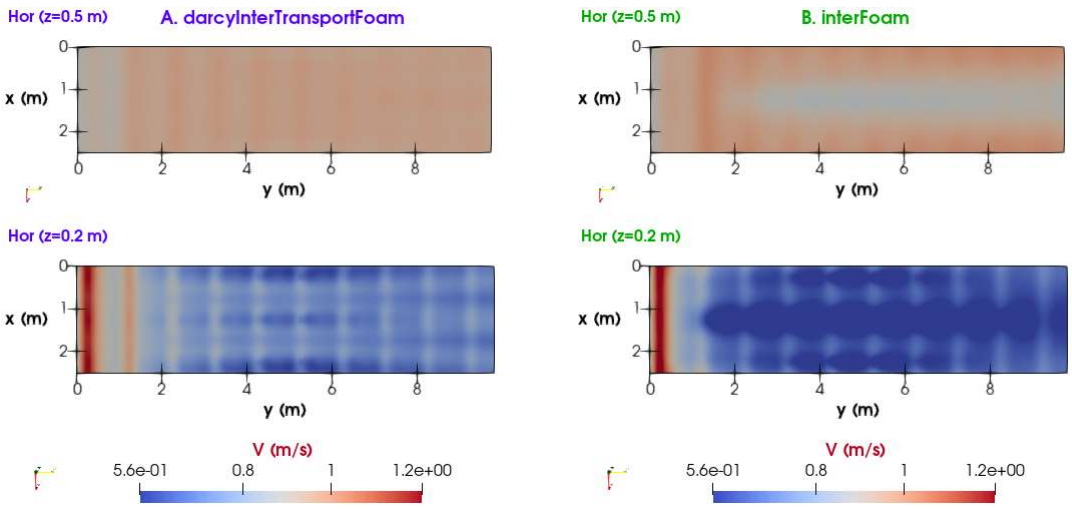



Figure C.14. Flow velocity field (V) across the horizontal slices at z=0.5 m and z=0.2 m.

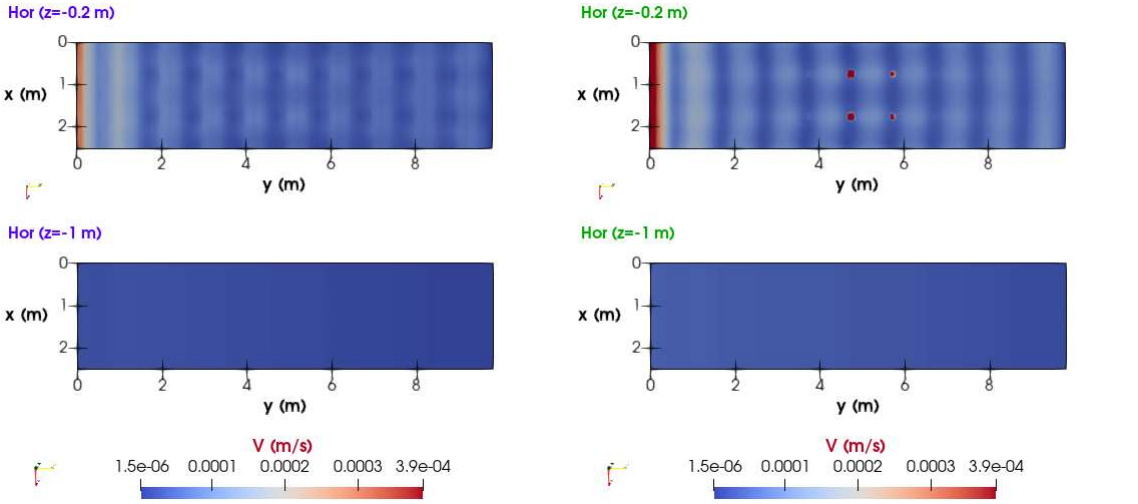



Figure C.15. Flow velocity field (V) across the horizontal slices at z=-0.2 m and z=-1 m.

## C.2.3. Hydraulic Head

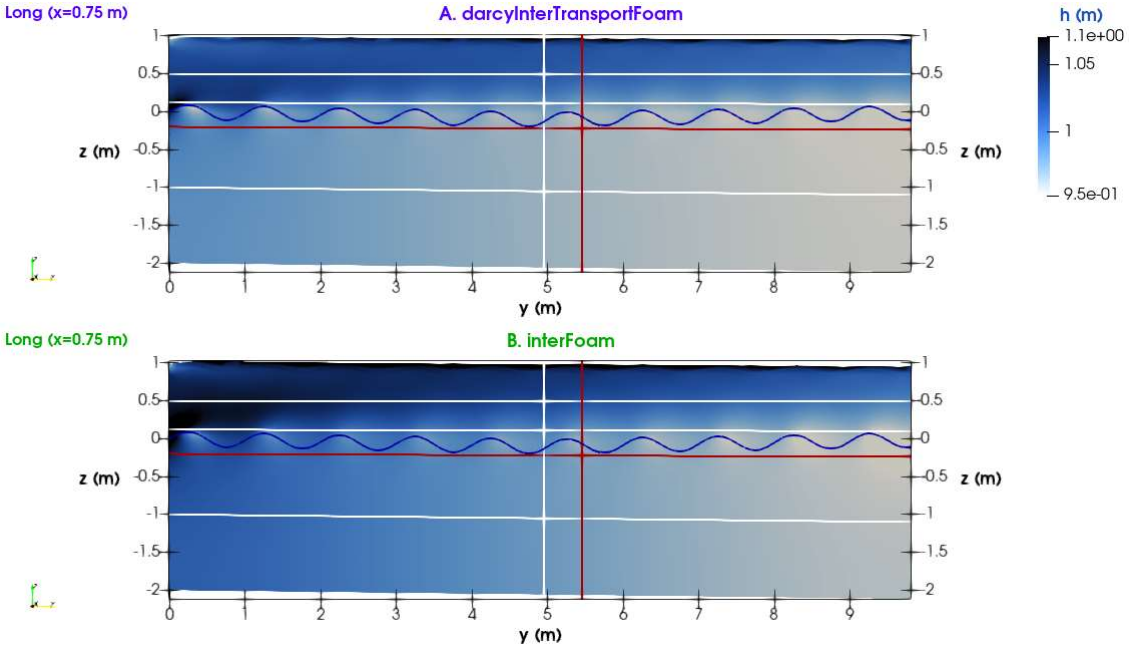



Figure C.16. Hydraulic head field (h) across the longitudinal slice at x=0.75 m. The red and white lines indicate the transversal and horizontal slices that intersect it (Figure 3.4).

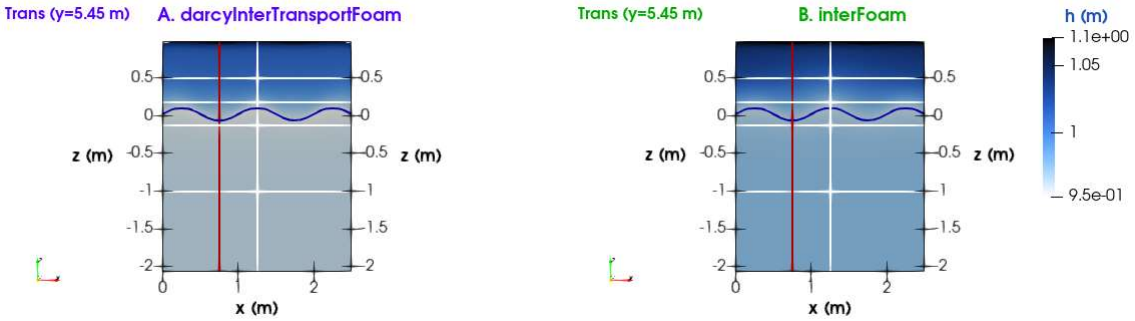



Figure C.17. Hydraulic head field (h) across the traversal slice at y=5.45 m. The red and white lines indicate the longitudinal and horizontal slices that intersect it (Figure 3.4).

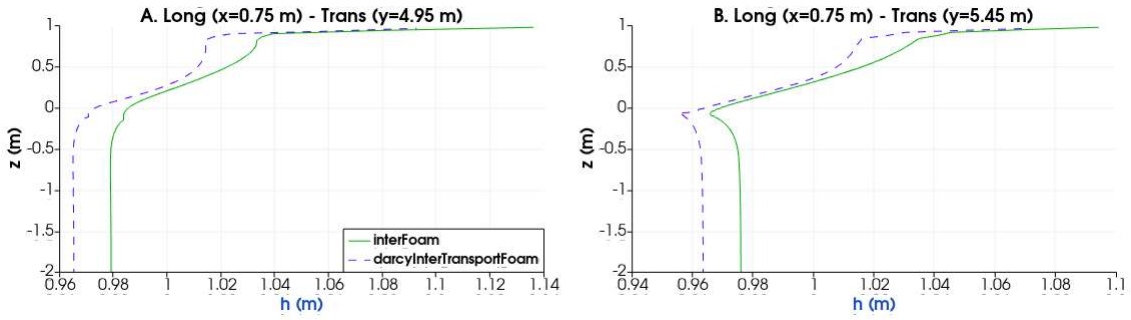



Figure C.18. Hydraulic head (h) along the intersections of the longitudinal slice at x=0.75 m and the transversal slices at y=4.95 m and y=5.45 m.

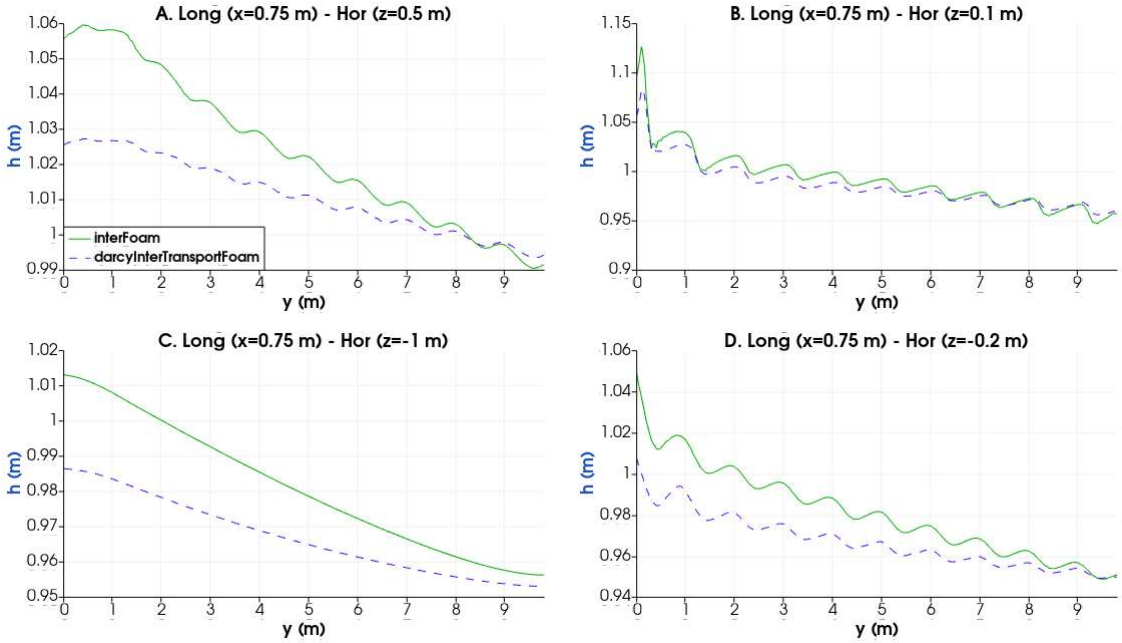



Figure C.19. Hydraulic head (h) along the intersections of the various horizontal slices with the longitudinal slice at x=0.75 m.

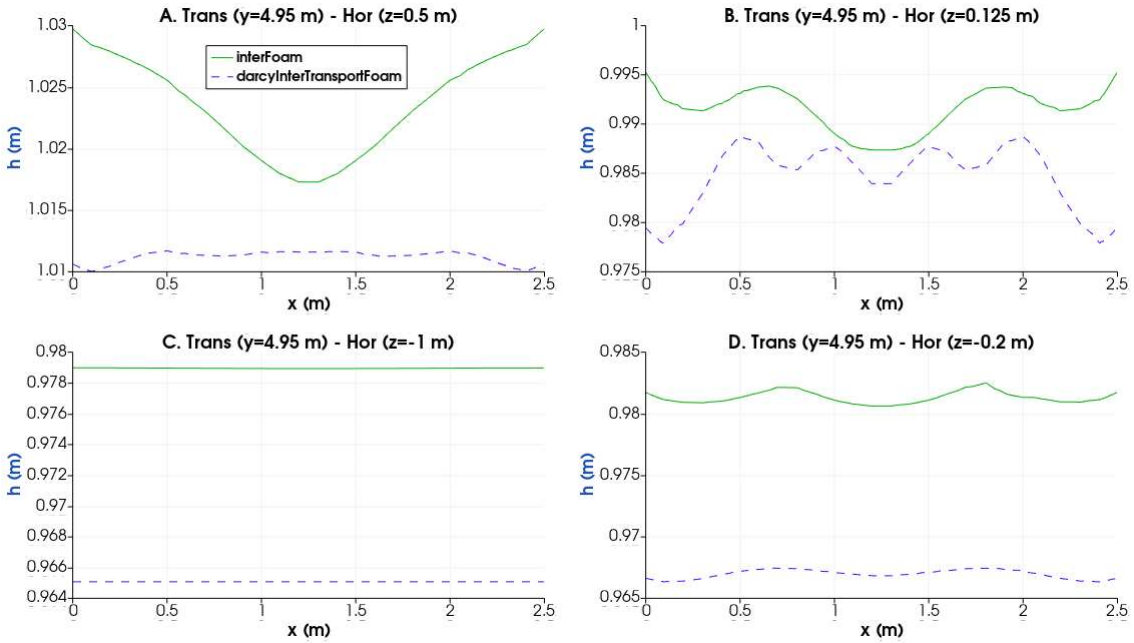



**Figure C.20.** Hydraulic head (h) along the intersections of the various horizontal slices with the transversal slice at y=4.95 m.

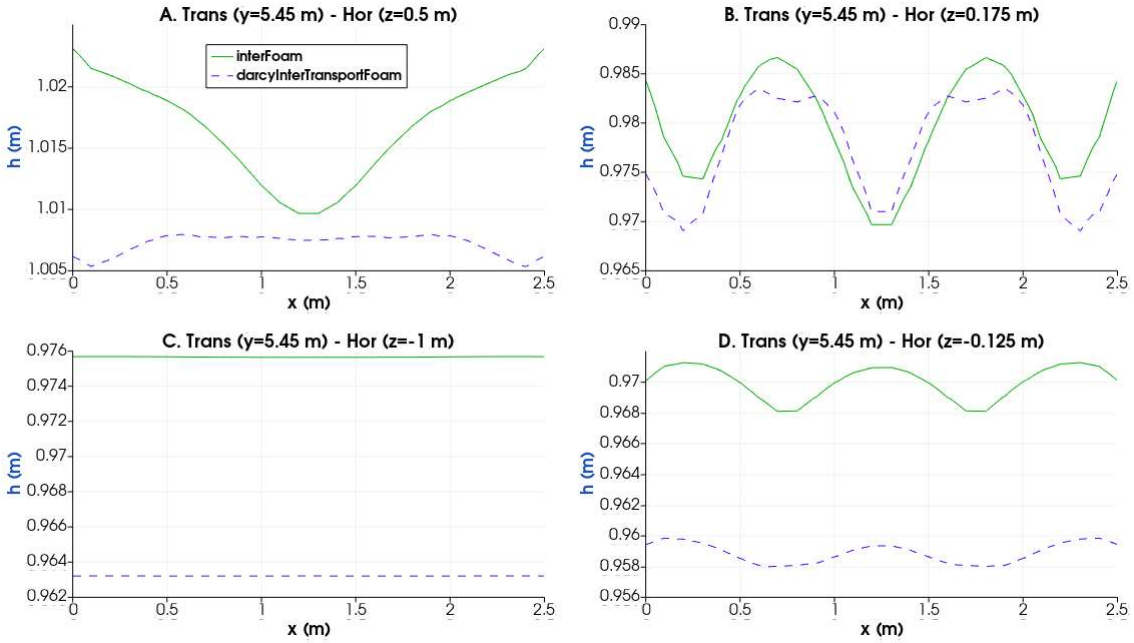



**Figure C.21.** Hydraulic head (h) along the intersections of the various horizontal slices with the transversal slice at y=5.45 m.

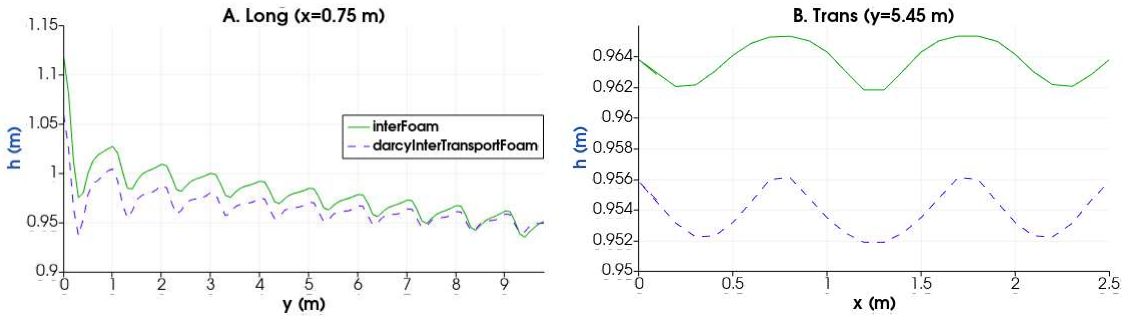



Figure C.22. Hydraulic head (h) along the intersections of the riverbed with the longitudinal slice at x=0.75 m (A) and the transversal slice at y=5.45 m (B).

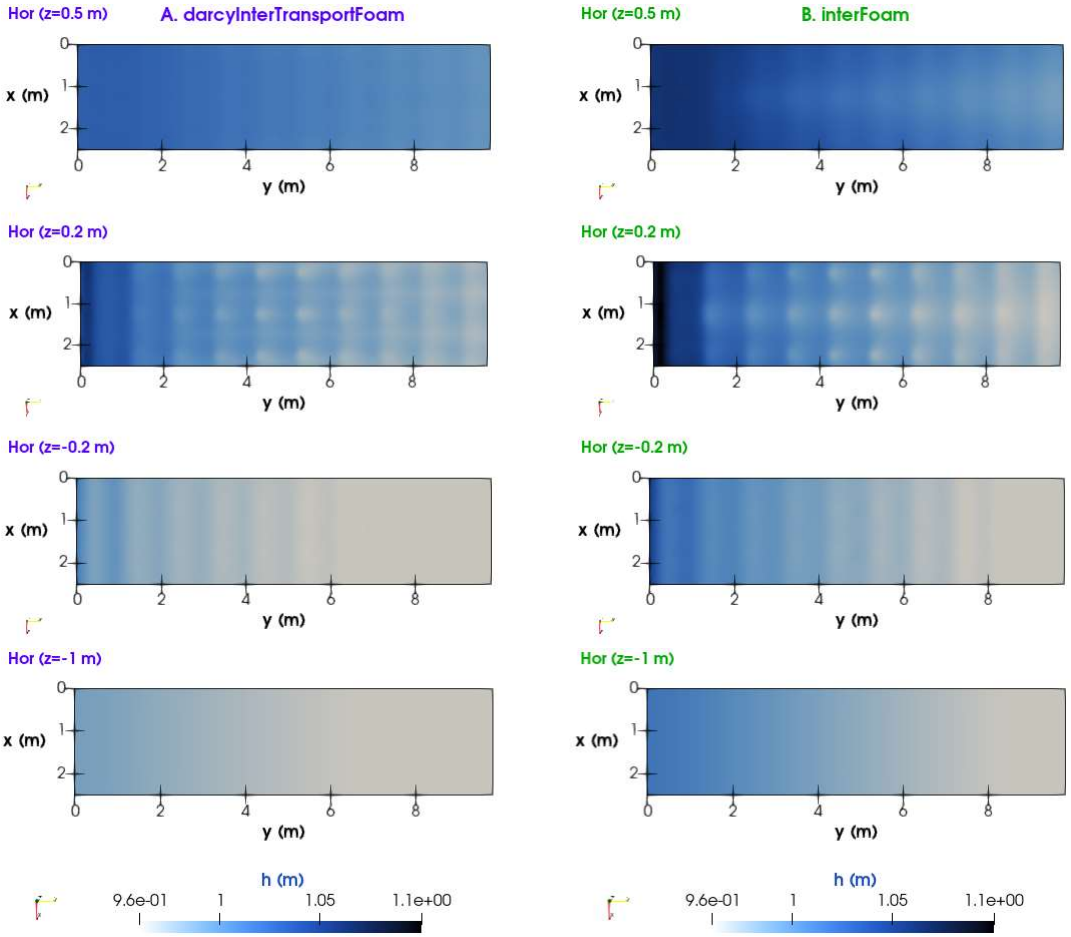



Figure C.23. Hydraulic head field (h) across the horizontal slices at z=0.5 m, z=0.2 m, z=-0.2 m and z=-1 m.

## C.2.4. Solute Concentration

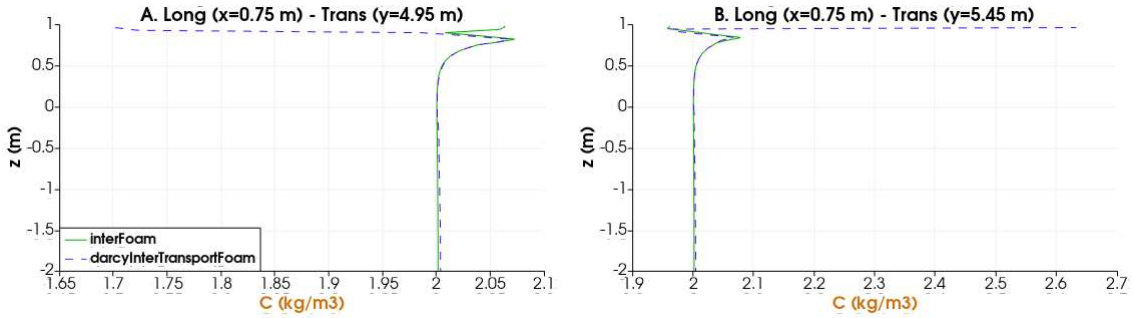



Figure C.24. Solute concentration (C) along the intersections of the longitudinal slice at x=0.75 m and the transversal slices at y=4.95 m and y=5.45 m.

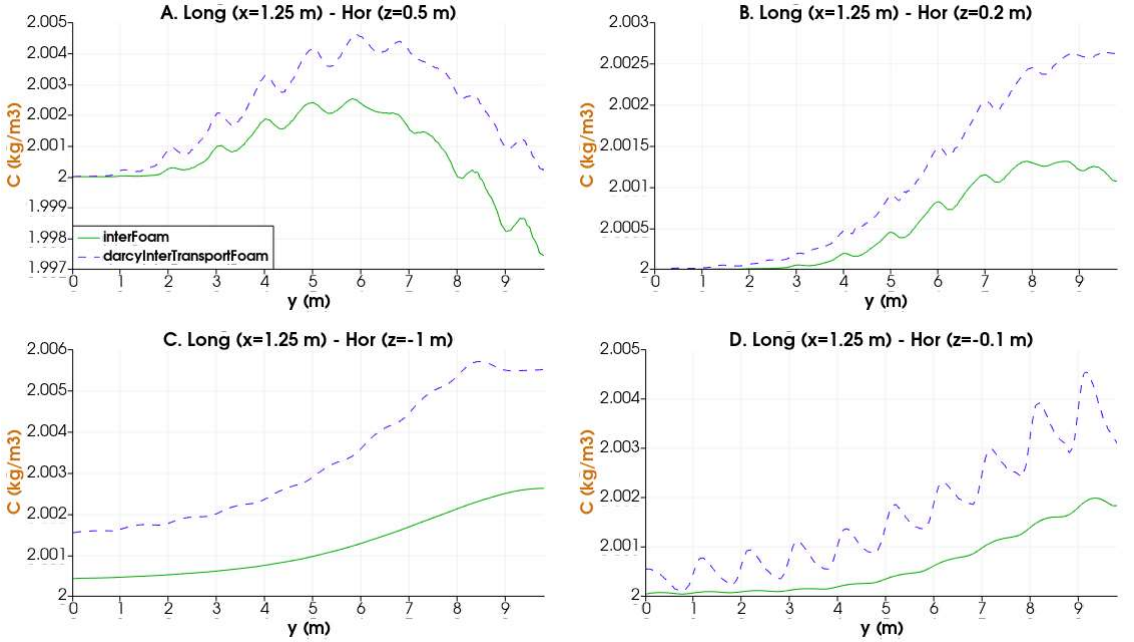



Figure C.25. Solute concentration (C) along the intersections of the various horizontal slices with the longitudinal slice at x=1.25 m.

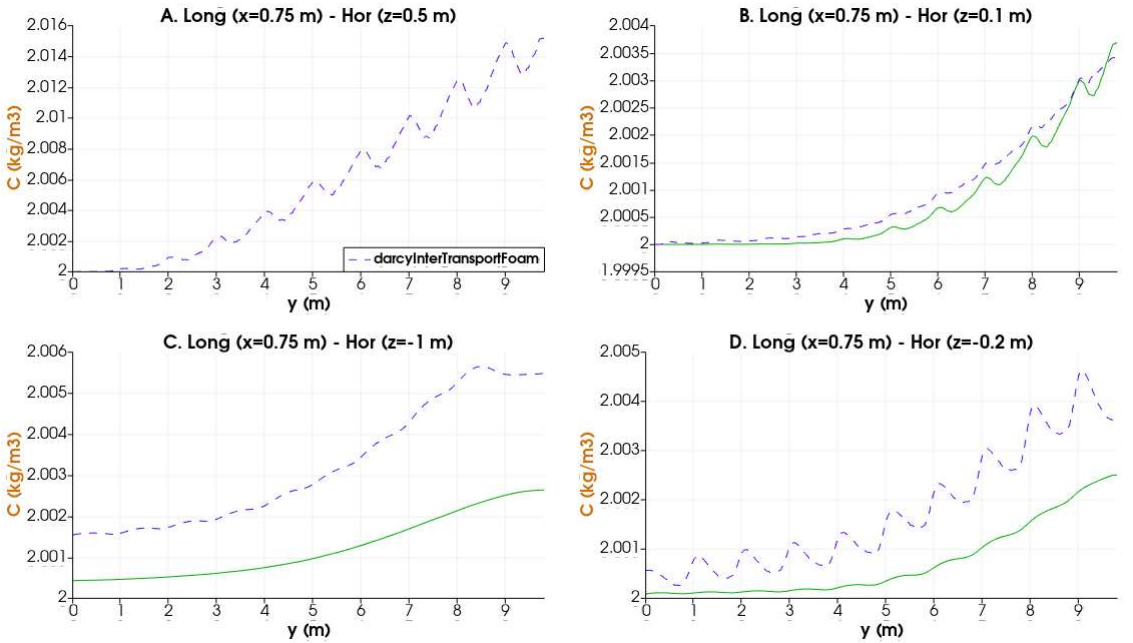



Figure C.26. Solute concentration (C) along the intersections of the various horizontal slices with the longitudinal slice at x=0.75 m.

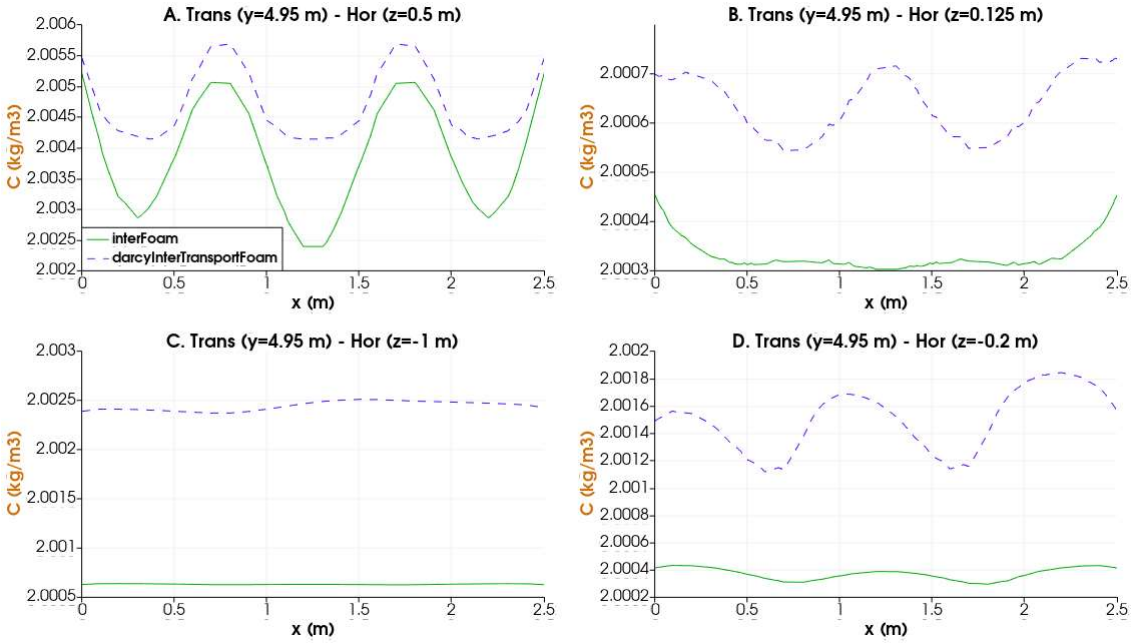



Figure C.27. Solute concentration (C) along the intersections of the various horizontal slices with the transversal slice at y=4.95 m.

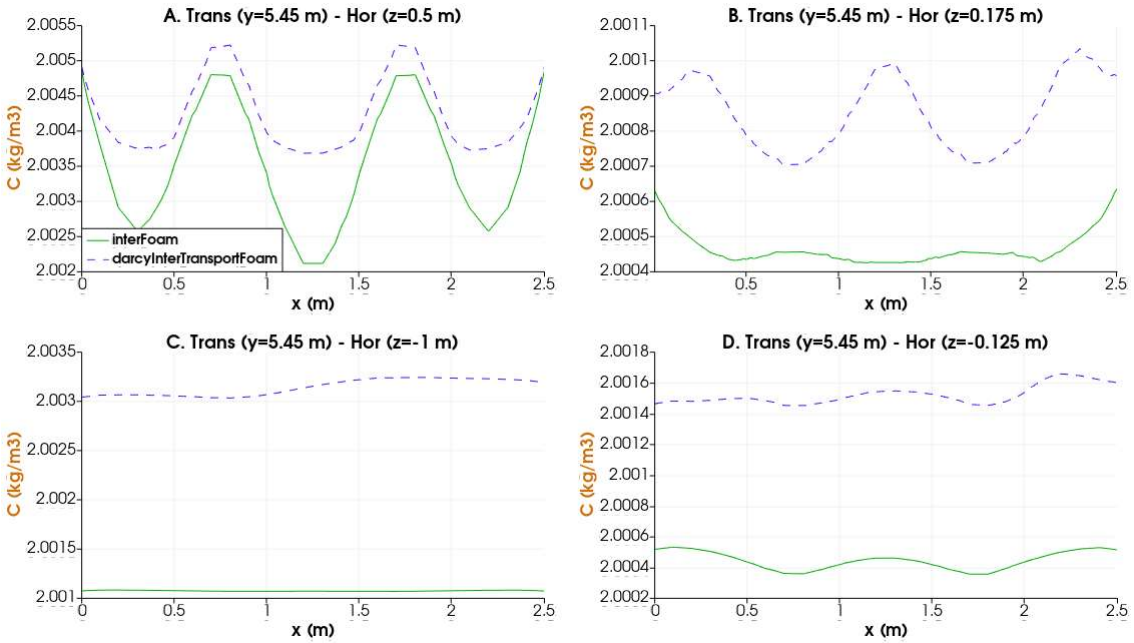



Figure C.28. Solute concentration (C) along the intersections of the various horizontal slices with the transversal slice at y=5.45 m.

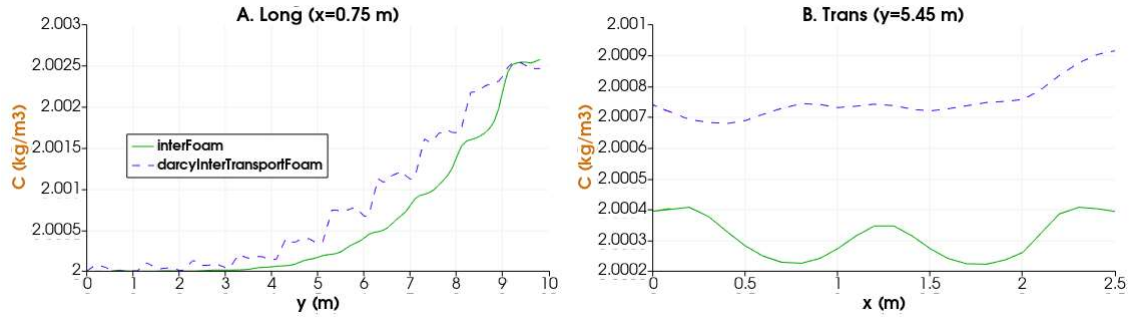



Figure C.29. Solute concentration (C) along the intersections of the riverbed with the longitudinal slice at x=0.75 m (A) and the transversal slice at y=5.45 m (B).

# References

[1] OpenCFD Ltd. (2023, December). *OpenFOAM® v2312 User Guide*. Retrieved December 25, 2024, from https://doc.openfoam.com/2312/