Numerical modeling based on spline basis functions : application to groundwater flow modeling in karst aquifers and advection dominated problems

Malenica, Luka

Doctoral thesis / Disertacija

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:
University of Split, Faculty of Civil Engineering, Architecture and Geodesy / Sveučilište u Splitu, Fakultet građevinarstva, arhitekture i geodezije

https://doi.org/10.31534/DocT.047.MalL

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:123:222929

UNIVERSITY OF SPLIT

FACULTY OF CIVIL ENGINEERING, ARCHITECTURE AND GEODESY

Luka Malenica, mag.ing.aedif.

Numerical modeling based on spline basis functions: Application to groundwater flow modeling in karst aquifers and advection dominated problems

Doctoral dissertation

Split, 2019.

Luka Malenica, mag.ing.aedif.
Serial number: 047

This doctoral dissertation has been submitted for evaluation to Faculty of Civil Engineering, Architecture and Geodesy of University of Split, for the purpose of obtaining the doctoral degree in the field of Technical sciences, scientific field of Civil engineering.

Supervisor: Prof. Hrvoje Gotovac, PhD

The committee for evaluation:

Prof. Roko Andričević, PhD
Prof. Zoran Milas, PhD
Prof. Goran Lončar, PhD
Prof. Kristijan Posavec, PhD
Prof. Željko Tuković, PhD

The defense committee:

Prof. Roko Andričević, PhD

Prof. Goran Lončar, PhD

Prof. Željko Tuković, PhD

The doctoral dissertation has been defended on February 8, 2019.

Secretary:

Saša Delić, LL.M.

This dissertation contains:

174 pages

81 figures

11 tables

117 references

To my family

Acknowledgments

This thesis has been supported by the Croatian Science Foundation (in Croatian Hrvatska zaklada za znanost—HRZZ) through grant number: UIP-2013-11-8103, and Faculty of Civil Engineering, Architecture and Geodesy (FGAG) at University of Split.

I would like to thank my supervisor, prof. Hrvoje Gotovac for his support, guidance, and friendship. The amount of time and energy he invested in our work is really special. He was always at a disposal and ready to help, both in professional and personal sense. I would also like to thank all my professors, colleagues and friends for their help and support. I thank committee members for their useful comments and suggestions.

Special thanks goes to my family, especially my parents Silvana and Ivica, wife Kasia, and our kids Marita and Šimun who were my greatest inspiration.

UDC 532.5:[517.5:519.87](043.3)
556.3:551.435.8](043.3)

Luka Malenica, mag.ing.aedif.

Numerical modeling based on spline basis functions: Application to groundwater flow modeling in karst aquifers and advection dominated problems

Abstract:

The main objective of this thesis is to utilize the powerful approximation properties of spline basis functions for numerical solutions of engineering problems that arise in the field of fluid mechanics. Special types of spline functions, the so-called Fup basis functions, are used as representative members of the spline family. However, the techniques developed in this work are quite general with respect to the choice of different spline functions.

The application of this work is twofold. The first practical goal is the development of a novel numerical model for groundwater flow in karst aquifers. The concept of isogeometric analysis (IGA) is presented as a unified framework for multiscale representation of the geometry, material heterogeneity and solution. Moreover, this fundamentally higher-order approach enables the description of all fields as continuous and smooth functions by using a linear combination of spline basis functions. Since classical IGA uses the Galerkin and collocation approach, in this thesis, a third concept, in the form of control volume isogeometric analysis (CV-IGA), is developed and set as the foundation for the development of a karst flow numerical model. A discrete-continuum (hybrid) approach is used, in which a three-dimensional laminar matrix flow is coupled with a one-dimensional turbulent conduit flow. The model is capable of describing variably saturated conditions in both flow domains. Since realistic verification of karst flow models is an extremely difficult task, the particular contribution of this work is the construction of a specially designed 3D physical model (dimensions: 5.66 x 2.95 x 2.00 m) to verify the developed numerical model under controlled laboratory conditions.

As a second application, this thesis presents the development of a full space-time adaptive collocation algorithm with particular application to advection-dominated problems. Since these problems are usually characterized by numerical instabilities, the novel adaptive algorithm accurately resolves small-scale features while controlling the numerical error and

spurious numerical oscillations without need for any special stabilization technique. The previously developed spatial adaptive strategy dynamically changes the computational grid at each global time step, while the novel adaptive temporal strategy uses different local time steps for different collocation points based on the estimation of the temporal discretization error. Thus, in parts of the domain where temporal changes are demanding, the algorithm uses smaller local time steps, while in other parts, larger local time steps can be used without affecting the overall solution accuracy and stability. In contrast to existing local time stepping methods, the developed method is applicable to implicit discretization and resolves all temporal scales independently of the spatial scales. The efficiency and accuracy of the full space-time adaptive algorithm is verified with some classic 1D and 2D advection-diffusion benchmark test cases.

Keywords: spline basis functions, control volume isogeometric analysis, coupled numerical karst flow model, physical karst flow model, multiscale modeling, adaptive numerical modeling, local time stepping, advection-dominated problems

UDK 532.5:[517.5:519.87](043.3)
556.3:551.435.8](043.3)

Luka Malenica, mag.ing.aedif.

Numeričko modeliranje bazirano na spline baznim funkcijama: Primjena na modeliranju tečenja u krškim vodonosnicima i advektivno dominantnim problemima

Sažetak:

Glavni cilj ove disertacije je iskoristiti moćna aproksimacijska svojstva spline baznih funkcija za numeričko modeliranje inženjerskih problema u području mehanike fluida. Posebni tipovi spline baznih funkcija, takozvane Fup bazne funkcije, su izabrane kao reprezentativni član spline funkcija. Međutim, tehnike razvijene u ovom radu su poprilično generalne s obzirom na izbor različitih spline funkcija.

Primjena u ovom radu je dvostruka. Prvi praktični cilj je razvoj numeričkog modela za modeliranje tečenja u krškim vodonosnicima. Koncept izogeometijske analize (IGA) je predstavljen kao ujedinjeni pristup za višeskalno modeliranje geometrije, heterogenosti materijala i rješenja. Osim toga, pristup je fundamentalno višeg reda, dok linearna kombinacija spline baznih funkcija omogućuje opis različitih polja kao kontinuiranih i glatkih funkcija. Kako klasična IGA generalno koristi Galerkin-ov ili kolokacijski pristup, u ovoj tezi razvijen je treći pristup u obliku izogeometrijske analize kontrolnog volumena (CV-IGA), te postavljen kao temelj za razvoj numeričkog modela za tečenje u kršu. Primijenjen je diskretno-kontinuumski (hibridni) pristup u kojem je trodimenzionalno laminarno tečenje u krškoj matrici spregnuto s jednodimenzionalnim turbulentnim tečenjem u kanalima. Model je u stanju opisati varijabilno zasićene uvjete u obje domene. Budući da je verifikacija krškog model iznimno težak zadatak, poseban doprinos ovog rada je konstrukcija posebno dizajniranog 3D fizikalnog modela (dimenzije: 5.66 x 2.95 x 2.00 m) u svrhu validacije razvijenog numeričkog modela u kontroliranim laboratorijskim uvjetima.

Kao drugu aplikaciju, ovaj rad predstavlja razvoj potpunog prostorno-vremenskog adaptivnog kolokacijskog algoritma s posebnom primjenom na advektivno-dominantnim problemima. Budući da su takvi problemi obično karakterizirani numeričkim nestabilnostima, novi adaptivni algoritam precizno opisuje lokalizirane strukture rješenja, kontrolirajući numeričku pogrešku i oscilacije bez posebnih tehnika stabilizacije. Prethodno razvijena

prostorno adaptivna strategija dinamički mijenja broj i raspored kolokacijskih točaka u svakom globalnom vremenskom koraku, dok nova adaptivna vremenska strategija koristi različite lokalne vremenske korake za različite kolokacijske točke na temelju procjene pogreške vremenske diskretizacijske. Dakle, u dijelovima domene gdje su vremenske promjene zahtjevne algoritam koristi manje lokalne vremenske korake, dok u drugim dijelovima koristiti veće vremenske korake bez utjecaja na ukupnu točnost i stabilnost numeričkog rješenja. Za razliku od postojećih metoda koje koriste tehnike lokalnih vremenskih koraka, razvijena metoda primjenjiva je kod implicitne diskretizacije, te pronalazi sve vremenske skale neovisno o prostornim skalama. Učinkovitost i točnost cjelovitog prostorno-vremenskog adaptivnog algoritma potvrđena je klasičnim 1D i 2D advektivno-difuzijskim testnim primjerima.

Ključne riječi: spline bazne funkcije, izogeometrijska formulacija kontrolnih volumena, spregnuti numerički model tečenja u kršu, fizikalni model tečenja u kršu, višeskalno modeliranje, adaptivne numeričke tehnike, lokalni vremenski koraci, advektivno dominantni problemi

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview and motivation

Numerical analysis is a widely accepted tool for the simulation of many complex engineering problems. In recent decades, many different numerical approaches have been suggested. Generally, each approach possesses specific strengths and weaknesses, and no single numerical method is the best for all applications. However, some of the most popular and versatile discretization techniques are the finite element method (FEM) and finite volume method (FVM). Methods such as the finite difference method (FDM), spectral element method (SEM), boundary element method (BEM), discrete element method (DEM), together with various collocation, meshfree and other hybrid approaches are usually practical only for limited classes of problems.

Differences among numerical methods can be found in their formulations, discretization approaches (for both domain and equations), interpolation profiles (or basis functions), order of accuracy, treatment of boundary conditions, and many other characteristics. In this work, the main emphasis will be on using the advantages of by spline functions as the basis for the development of novel numerical models in the field of fluid mechanics.

Regarding the interpolation profiles for the discretization of the governing partial differential equations (PDEs), two main approaches have historically dominated the field of computational mechanics. The first and oldest forms the basis for the finite difference method. The starting point of FDM is to cover the solution domain with a (generally uniform) grid. At each grid point, the differential form of the PDE is discretized by approximating the derivatives with finite difference expressions. These expressions are generally derived by using Taylor series expansion or polynomial fitting through a certain number of specific neighboring grid points. Since each grid point produces one discretized (algebraic) equation with multiple unknowns, all equations must be combined in the system and solved simultaneously. The

final results of the FDM are the solution values at the grid points. However, there is no explicit reference for how a solution behaves between grid points. In this way, the FDM can be thought of as akin to a laboratory experiment, where measurements are obtained at some discrete locations (Patankar [1]). Besides the fact that the conservation properties of the governing equations are generally not preserved, the main disadvantage limiting the FDM from widespread application is its restriction to simple geometries.

The finite volume method (FVM) can be considered to be a natural improvement of the FDM. Generally, the solution domain is subdivided into an finite number of non-overlapping control volumes (CVs), and the conservative form of the governing PDEs is integrated over each CV. The volume integrals over the CVs are converted to surface integrals via the divergence theorem, and interpolation is used to express the values at the CV boundaries in terms of nodal (CV-center) values. The finite differences expressions are used to discretize the derivatives, while suitable numerical quadrature is used for integration. Again, the final results are the solution values at the nodal points. Since different interpolation profiles can be used for different terms arising in the governing equations, there is no explicit reference for the solution variation between nodal points. The FVM is conservative by its construction, and its main advantages are the direct physical meaning of the discretized equations and its formulation for complex geometries. The disadvantage of the FVM is that methods higher than second order are difficult to construct [2]. This is mostly because the FVM requires three levels of approximation, namely, interpolation, differentiation and integration,and approximation profiles of higher order are difficult to construct on unstructured meshes.

The FDM and FVM do not use interpolation profiles in terms of a defined interpolation space (such as trial basis functions space); rather, they use local approximation profiles. This local interpolation profiles are not necessarily the same for all terms in the governing equations. This approach permits complete freedom; however (as mentioned before), the solution is not uniquely defined throughout the domain, except for discrete nodal (grid) points. Moreover, these interpolation profiles are generally one-dimensional and the consequence of using such profiles is usually reduced accuracy for multidimensional problems. This is particularly true when first-order interpolation is used on meshes oblique to flow gradients [1].

The second widely used approach to interpolation profiles is the one

adopted in the finite element method (FEM), as well as in most variational-based formulations. The trial vector space of basis (shape) functions is defined, and a linear combination of these functions is used for both interpolation and differentiation during the discretization process. In this way, the final solution is uniquely defined throughout the whole domain. The most widely used basis functions among the FEM community are Lagrangian polynomials. The same functions are commonly used to construct test (weighted) function spaces through the Galerkin formulation and for geometry mapping through the isoparametric concept. The multidimensional basis functions are simply constructed by using the tensor product of one-dimensional basis functions.

The main strengths of the FEM are the ability to handle arbitrary geometries, the construction of higher-order approximations on unstructured grids, and the strong mathematical background created during years of research (e.g., Bathe [3] and Zienkiewicz *et al.* [4]). Historically, the classical (Galerkin) FEM has dominated the fields of soil and structural mechanics due to the excellent approximation properties for elliptic or parabolic problems characterized by self-adjoint (symmetric) operators. However, the presence of an inevitable convective (advective) term in the fluid governing equations (generally Navier-Stokes equations) subjects the FEM to numerical instabilities (e.g., Courant or Peclet number). Different stabilizations have been proposed (e.g., Zienkiewicz *et al.* [5]), however the same effects are more naturally achieved via the FDM and FVM (e.g., upwinding).

The numerical solutions produced by FEM are continuous and smooth inside a particular element. However, application of classical Lagrangian basis functions ensures only C^0 continuity on the element boundaries, even in the case when higher-order basis functions are used. Moreover, the gap between the finite element analysis (FEA) for the solution description on the one hand and computer-aided design (CAD) for the geometry description on the other hand has long been evident. Mesh generation still presents a substantial bottleneck to the effective use of FEA for complex engineering problems. It is estimated that approximately 80% of the overall analysis time is devoted to mesh generation in the automotive, aerospace, and ship building industries [6]. Moreover, once a mesh is constructed, each mesh refinement usually requires communication with the CAD system, which is probably the main reason techniques such as adaptive mesh refinement (AMR) are still primarily of academic interest.

This gap between FEA (as well as overall numerical analysis) and CAD

systems is due mostly to differences in the used interpolation (basis) functions. Whereas classical polynomials have dominated the field of numerical analysis, spline-based basis functions (e.g., B-splines, NURBS, T-splines, etc.) play a crucial role in the field of computational geometry. In addition to use in CAD systems, they are extensively used for data fitting and computer graphics, where their powerful approximation properties have been demonstrated.

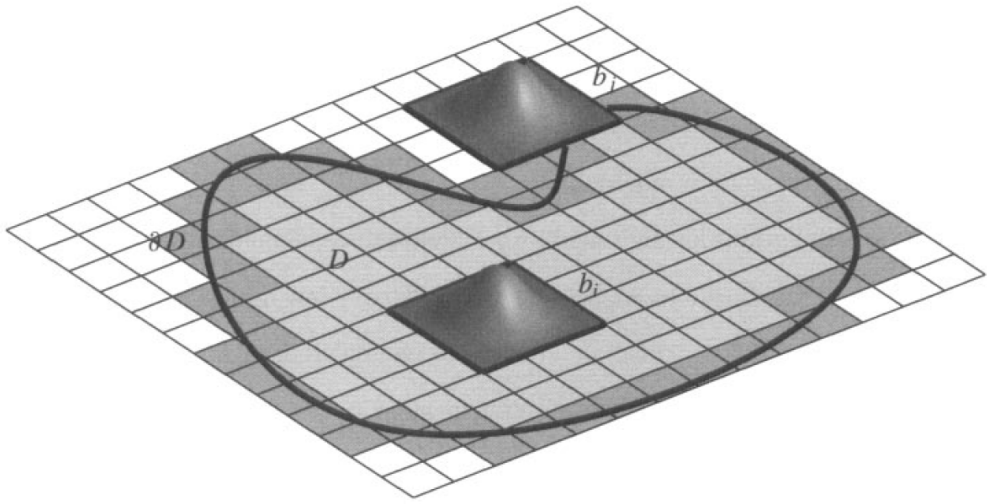


FIGURE 1.1: Grid cells of biquadratic B-splines with support intersecting a bounded domain (taken from Höllig [7]).

Since geometric modeling and numerical simulations are closely linked in computational engineering, the use of splines as finite element basis functions suggests itself [7]. The basic idea is to combine the advantages of B-spline approximations on regular grids and standard mesh-based finite elements (Figure 1.1). However, the early efforts have not been successful for two main reasons:

1.) The satisfaction of the Dirichlet boundary conditions is not straightforward due to the non-interpolatory nature of basis functions. If a linear combination of B-splines is required to vanish on the boundary ∂D of domain D, then, in general, all coefficients of the linear combination interacting ∂D must be zero. This characteristic results in very poor approximation for problems with Dirichlet boundary conditions.

2.) The B-spline basis is not uniformly stable for general geometries. As shown in Figure 1.1, the basis may contain functions with very small support inside domain D, which can lead to an ill-conditioned matrix and cause slow convergence of iterative methods.

One approach for overcoming both of the mentioned difficulties is given in Höllig [7] by construction of so-called weighted extended B-splines (web-splines). This (basically meshless) approach avoids mesh generation by using a regular mesh background (Figure 1.1), while web-splines reduce the stiffness matrix condition number and stabilize the numerical methodology. However, true popularity of spline functions for numerical analysis was achieved by the introduction of the concept of isogeometric analysis (Hughes *et al.* [6] and Cottrell *et al.* [8]). The main idea of isogeometric analysis (IGA) is to overcome the gap between FEA and CAD by using the same type of spline basis functions for both systems. The root of IGA is the isoparametric concept (widely used in classical FEM), where the basis functions used to approximate the solution fields are also used to represent the geometry. The IGA turns this idea around and selects a basis used to describe the geometry in CAD systems as the basis for numerical approximation of unknown fields. The crucial difference is that, contrary to classical FEA, where the geometry is only approximated, IGA enables the geometry to be represented exactly (in the CAD sense).

Moreover, in addition to the clear advantages for describing the geometry, the spline basis functions also possess many interesting properties that are quite desirable for approximating solution fields. The main advantages can be summarized as follows:

- Exact representation of CAD geometry
- The approach is fundamentally high order
- High continuity of numerical solutions
- Increased accuracy and robustness per-degree-of-freedom when compared with same order FEM
- Demanding mesh generation is avoided since splines operate on uniform grids
- Refinement can be performed without interaction with the CAD system
- Hierarchical spline functions enable adaptive refinement
- Spline functions of all orders have strictly positive values, which increases the numerical stability

1.2 Objectives

The main objective of this thesis is to demonstrate the capabilities of spline basis functions through the development of novel numerical methods with specific application to problems arising in the field of fluid mechanics. Emphasis is given to spline approximation properties, and only simple geometries are considered. In addition to classical B-splines, this work introduces lesses known Fup basis functions (Rvačev V.L. [9] and Gotovac and Kozulic [10]) as representative members of the spline family. The Fup basis functions belong to the class of atomic basis functions and can be regarded as infinitely differentiable (perfect) B-splines.

The first application of this work is the creation of novel 3D numerical model for the simulation of groundwater flow in karst aquifers. Prior to the development of the numerical karst flow model, this thesis analyzes different numerical formulations with spline-based basis functions through the framework of isogeometric analysis (IGA). Beside of the analysis of existing Galerkin and collocation IGA formulations, the novel numerical approach, called control volume isogeometric analysis (CV-IGA), is developed and set as the foundation for the development of the karst flow model. The emphasis is placed on the potential of IGA for multiscale description of the geometry, soil properties and solution fields as smooth and continuous functions through a fundamentally higher-order approach. The particular contribution of this thesis is the construction of a large 3D physical karst flow model. Since the verification and validation of complex numerical karst flow models is practically very difficult under realistic catchment conditions, this work presents a unique experimental setup that is used to verify the developed numerical model under controlled laboratory conditions.

As a second application based on Fup functions, this work improves a previously developed adaptive Fup collocation method (AFCM; [11]) via the development of a novel local time stepping procedure. The main idea of the method is to dynamically adapt the computational grid during the simulation so that the algorithm uses more collocation points (i.e., higher resolutions) only in regions where the solution changes are demanding (e.g., localized step gradients or discontinuities). In addition to existing (spatial) adaptive grid refinement capability, the novel algorithm resolves fine temporal solution scales by using smaller local time steps only in regions of the domain where the temporal solution changes are intensive. Special care is

devoted to the implicit temporal discretization, particularly because most existing space-time adaptive algorithms are constructed for only explicit methods. Moreover, a previous version of the AFCM algorithm used Fup basis functions for interpolation and grid adaptation, while derivatives occurring in PDEs were discretized by the classical finite-difference operator. Thus, this work presents the construction of a novel Fup collocation operator that can be used for discretization of derivatives as an alternative to the previously used finite-difference operator. The developed numerical procedure is particularly efficient for modeling advection-dominated problems, a demanding transport phenomenon that often arises in the field of fluid dynamics.

1.3 Outline

Chapter 2 provides the mathematical background of the spline basis functions used in this work. In addition to classical B-splines, the lesser known Fup basis functions are introduced.

In Chapter 3, the isogeometric analysis (IGA) approach is introduced. In addition to two classical IGA numerical formulations (Galerkin and collocation), a novel conservative and efficient control volume IGA formulation is developed. The concept of IGA is presented as a unified framework for multiscale description of the geometry, material heterogeneity and flow variables, where all approximated fields are represented as continuous and smooth functions. The problem of steady-state groundwater flow through highly heterogeneous porous media is used to perform convergence analysis of the three proposed formulations.

Chapter 4 addresses groundwater flow modeling in karst aquifers. The mathematical models that account for variably saturated conditions in karst aquifers are introduced. Discretization of the governing equations, as well as a coupled numerical algorithm, is discussed in detail. The 3D physical model is introduced, and additional information is provided in Appendix A. The experimental results, a comparison with the numerical results and a detailed discussion are presented at the end of this chapter.

Chapter 5 describes the development of the full space-time adaptive Fup collocation method. Discretization is described in detail, and the complex mathematical expressions for the development of the collocation operator can be found in Appendix B. A previously developed spatial adaptive strategy is briefly described, and the novel temporal adaptive strategy is introduced. The developed method is applied to advection-dominated problems,

and the numerical results confirm its efficiency.

Finally, Chapter 6 summarizes the most important findings of the thesis and provides suggestions for future research.

Chapter 2

Spline basis functions

This chapter provides the mathematical background for the two representative members of spline functions used in this work. The chapter starts with a description of B-splines, followed by a description of Fup functions. The Fup functions belong to the class of atomic functions and can be regarded as infinitely differentiable B-splines. Generally, atomic functions are defined as solutions of differential-functional equations (e.g., [10]–[13]). However, in this work, the Fup functions are constructed with the aid of convolution theorem, a procedure that clearly demonstrates their close relationship to B-splines.

2.1 B-spline basis functions

The simplest example of an algebraic B-spline basis function is the B-spline of the zero-th order $B_0(\xi)$:

$$B_0(\xi) = \begin{cases} 1 & \xi \in [-1/2, 1/2] \\ 0 & elsewhere \end{cases} \tag{2.1}$$

whose Fourier transform can be obtained in the following manner:

$$f_0(t) = \int_{-\infty}^{+\infty} B_0(\xi) \cdot e^{it\xi}\, d\xi = \int_{-1/2}^{+1/2} 1 \cdot cos(t \cdot \xi)\ dt = \frac{sin(t/2)}{t/2} \tag{2.2}$$

B-splines of the n-th order for the uniform distribution of knots according to the law - $\xi_k = k - (n+1)/2, k = 0, 1, ..., n+1$, can be presented as:

$$B_n(\xi) = \frac{1}{n!} \sum_{k=0}^{n+1} (-1)^k \cdot C_{n+1}^k \cdot \left(\xi + \frac{n+1}{2} - k\right)_+^n \tag{2.3}$$

where C_n^k are binomial coefficients:

$$C_n^k = \binom{n}{k} = \frac{(n)!}{(n-k)\,!\cdot k!} \tag{2.4}$$

For example, B-splines up to the third order can be presented according to (2.3) as follows:

$$\begin{aligned}
B_0(\xi) &= (\xi + 1/2)_+^0 - (\xi - 1/2)_+^0 \\
B_1(\xi) &= (\xi + 1)_+^1 - 2(\xi)_+^1 + (\xi - 1)_+^1 \\
B_2(\xi) &= \left[(2\xi + 3)_+^2 - 3(2\xi + 1)_+^2 + 3(2\xi - 1)_+^2 - (2\xi - 3)_+^2\right]/8 \\
B_3(\xi) &= \left[(\xi + 2)_+^3 - 4(\xi + 1)_+^3 + 6(\xi)_+^3 - 4(\xi - 1)_+^3 + (\xi - 2)_+^3\right]/6
\end{aligned} \tag{2.5}$$

Figure 2.1 shows that the compact support of $B_n(\xi)$ consists of $(n+1)$ unit characteristic intervals and $(n+2)$ knots, while on each interval $[\xi_k, \xi_{k+1}]$ $B_n(\xi)$ is presented by the local polynomial of the n-th order. For instance, $B_3(\xi)$ has four characteristic intervals and five knots. Furthermore, increasing the B-spline order increases the length of its compact support, and for the limit as $n \to \infty$, the length goes to infinity. The coordinate ξ_T is called the vertex and serves as the origin for the shifting of the basis functions along the ξ axis. Figure 2.2 presents the cubic B-spline $B_3(\xi)$ with the first three derivatives.

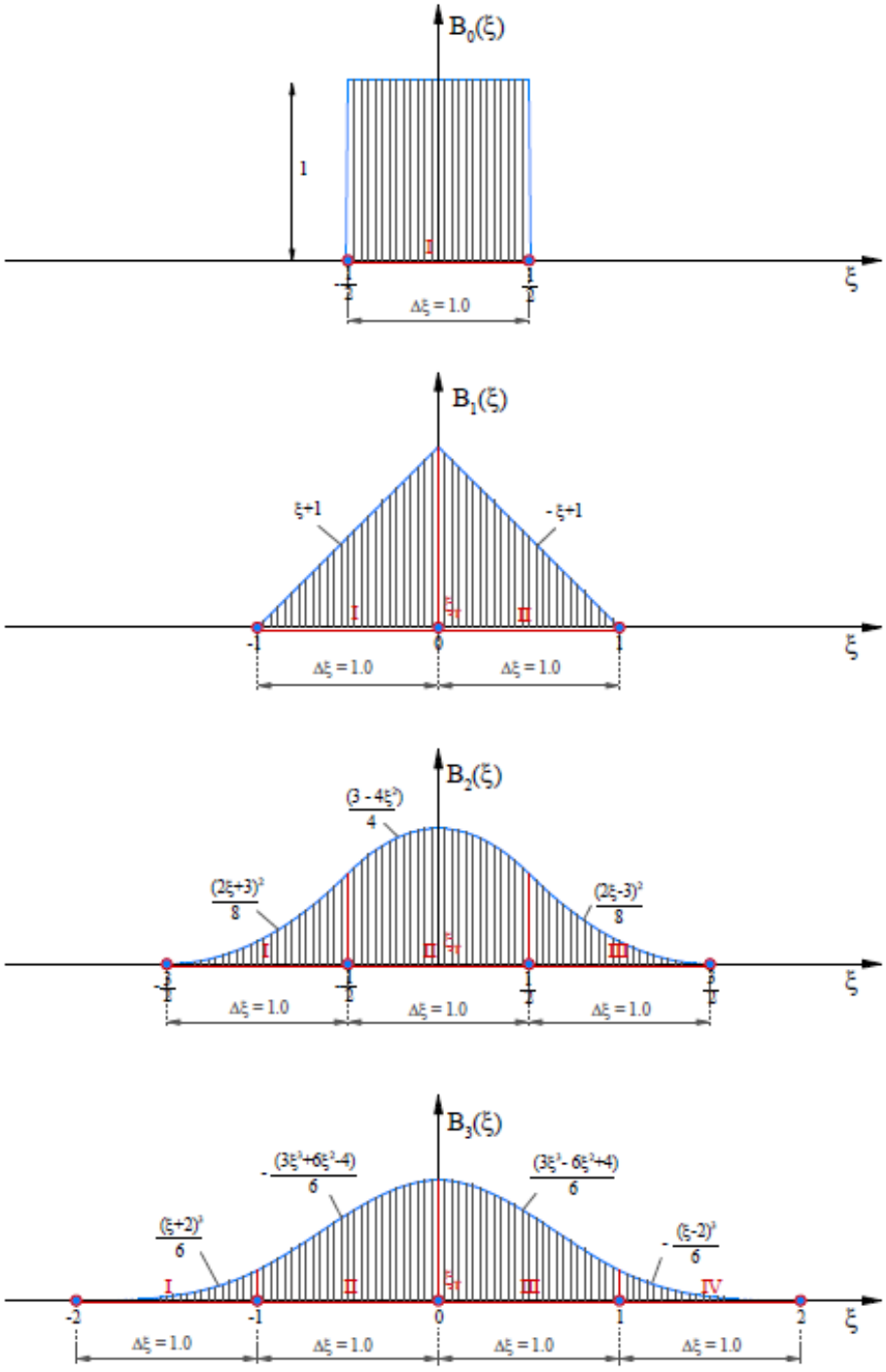


FIGURE 2.1: B-splines: $B_n(\xi)$, $n = 0, 1, 2, 3$.

Figures 2.1 - 2.3 presents the connection between B-splines and their derivatives. The first derivative of $B_3(\xi)$ can be presented as linear combination of contracted and shifted $B_2(\xi)$. Moreover, the second derivative of $B_3(\xi)$ can be presented as a linear combination of contracted and shifted $B_1(\xi)$, and so on. Finally, each $i-$th derivative of $B_n(\xi)$ is a linear combination of the contracted and shifted $B_{n-i}(\xi)$.

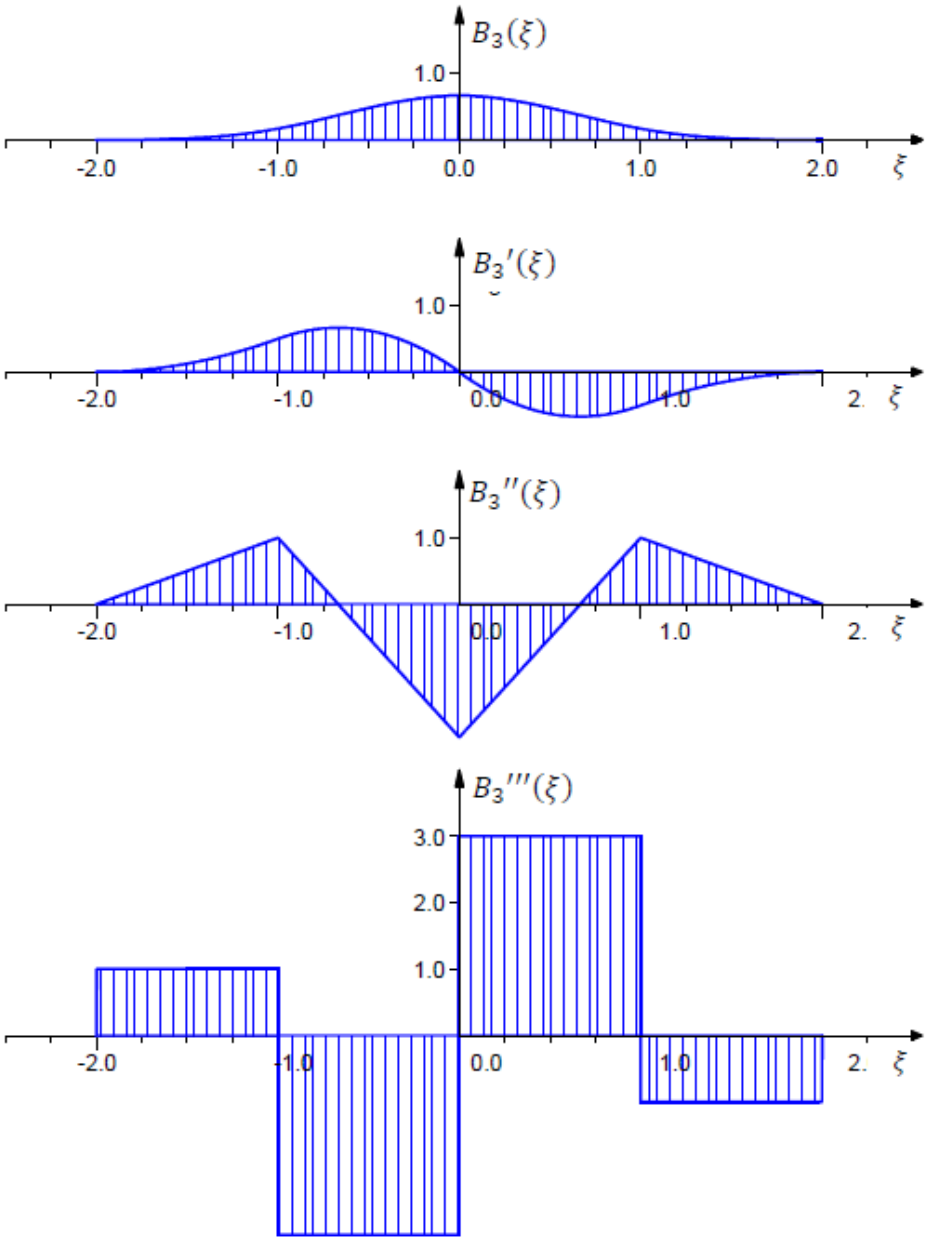


FIGURE 2.2: $B_3(\xi)$ with its first three derivatives.

According to the mentioned properties, $B_n(\xi)$ can be presented by convolution in the following form:

$$B_n(\xi) = \int_{-\infty}^{\infty} B_{n-1}(\xi - t)\, B_0(t)\ dt \tag{2.6}$$

or:

$$B_n(\xi) = B_{n-1}(\xi) * B_0(\xi) = \underbrace{B_0(\xi) * \ldots * B_0(\xi)}_{(n+1)\ \ times} \tag{2.7}$$

where n is the order of the B-spline and $B_0(\xi)$ is given by (2.1). The convolution theorem states that the Fourier transform (FT) of $B_n(\xi)$ can be expressed as a product of (*n+1*) particular FT's of $B_0(\xi)$ according to (2.7):

$$f_n(t) = \left(\frac{sint/2}{t/2}\right)^{n+1} \tag{2.8}$$

Equation (2.7) implies that the support of $B_n(\xi)$ is the union of the $(n+1)$ characteristic intervals $\Delta\xi$. Generation of B-splines according to (2.7) and the convolution theorem is shown in Figure 2.3.

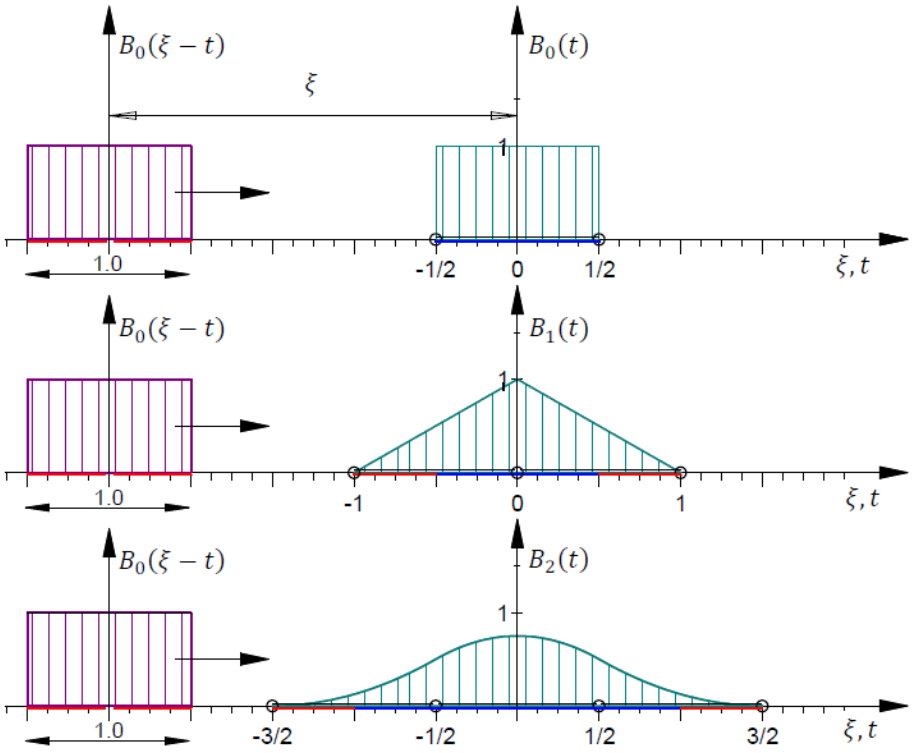


FIGURE 2.3: Generation of $B_2(\xi)$ using the convolution theorem.

Properties of the B-splines can be summarized as follows:

1. They have compact support where they have strictly positive non-zero values; elsewhere, they are zero, implying localized approximation properties.

2. A linear combination of shifted B-splines by a characteristic interval describes a unit constant function ("partition of unity").

3. A linear combination of shifted B_n-splines by a characteristic interval describes algebraic polynomials up to the n-th order.

4. B_n-splines can be presented by a linear combination of the shifted B-splines of the same order but using two-times-smaller support. This implies that B-splines support multiresolution analysis and efficient adaptive h, p and k numerical procedures (e.g., [14]–[18]).

2.2 Fup basis functions

Fup basis functions belong to the class of atomic basis functions (e.g., [9], [10]) and span vector space of algebraic polynomials, while their properties are closely related to the B-splines, as will be explained in the sequel. The simplest and basic atomic basis function is $up(\xi)$, which can be generated by the following convolution procedure:

$$up\left(\xi\right) = B_0\left(\xi\right) * B_0\left(2\xi\right) * \cdots * B_0\left(2^k\xi\right) * \cdots * B_0\left(2^\infty\xi\right) \tag{2.9}$$

$$F_0\left(t\right) = \prod_{j=1}^{\infty} \frac{\sin(t/2^j)}{t/2^j} \tag{2.10}$$

The function $up\left(\xi\right)$ is obtained by an infinite number of convolutions of the contracted $B_0\left(\xi\right)$ with compact support 2^{-k} and vertex value 2^k (see Eq. 2.9), as shown in Figure 2.4 and the Fourier transform (2.10). The compact support of $up\left(\xi\right)$ is the union of an infinite number of finite intervals according to (2.9); still, its compact support is finite:

$$h_{up} = \sum_{k=0}^{\infty} \frac{1}{2^k} = 2 \quad \rightarrow \quad \text{supp } up\left(\xi\right) = [-1,\ 1] \tag{2.11}$$

Rvačev V.L. [9] proved that the length of the compact support (2.11) can be presented as a distance metric of the set of binary-rational points 2^{-k}, while all other points as $\pm 1/3,\ \pm 4/7,\ \pm\sqrt{2}/2,\ \pm\pi/8$ contain zero metric length. The convolution procedure (2.9) causes $up(\xi)$ to contain all polynomial orders by parts of its compact support. The function $up(\xi)$ can be regarded as a perfect spline due to its infinite number of continuous and non-zero derivatives, but it still is not an analytic function at any point of its support. Moreover, its finiteness is higher than that of B-splines. The basic atomic basis function $up\left(\xi\right)$ retains the good localized property of B-splines but also possesses the property of universality, as algebraic or trigonometric polynomials. Universality means that adding new basis functions in some approximation can only improve or at least not change the previous approximation

which means that the new approximation contains old approximation as a vector subspace.

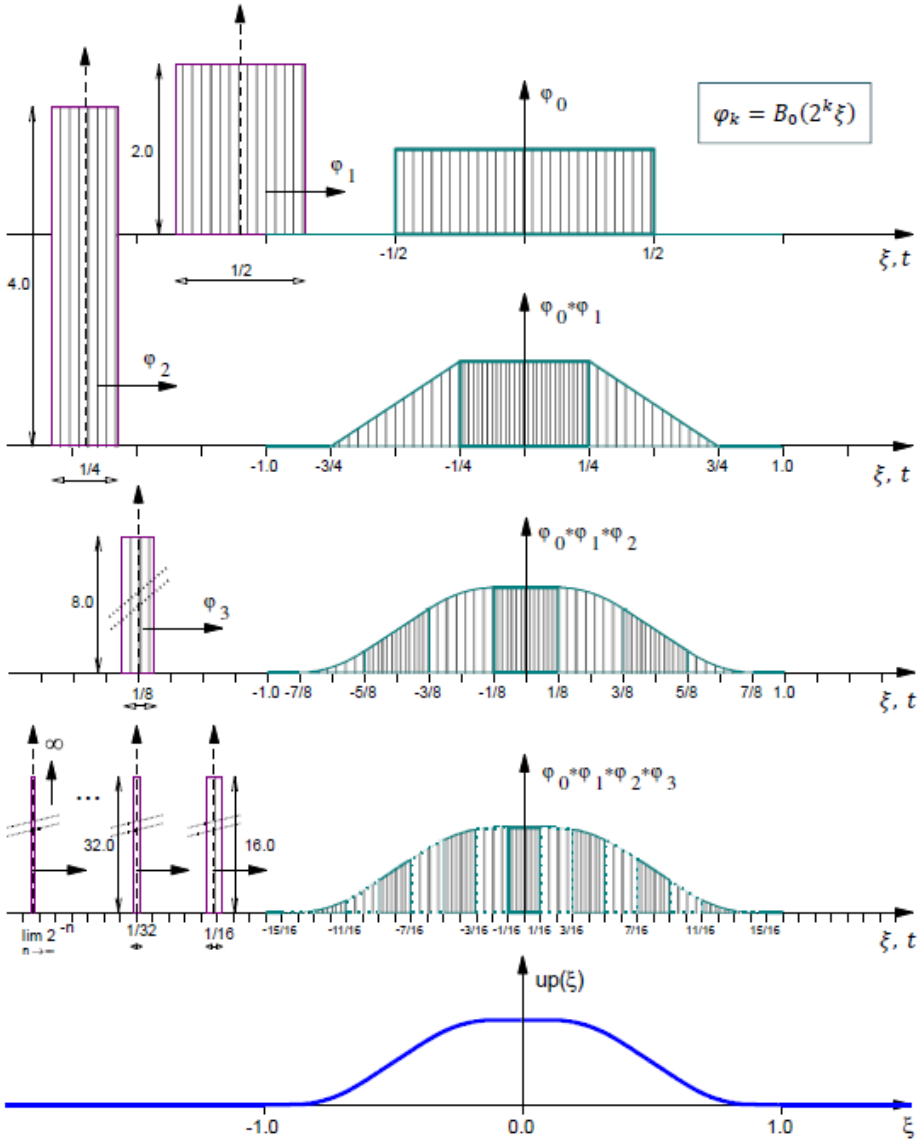


FIGURE 2.4: Generation of the function $up(\xi)$ with the aid of convolution theorem.

Let us define binary-rational points as:

$$\xi_{br} = -1 + k \cdot 2^{-m}, \quad m \in N, \quad k = 1, \ldots, 2^{m+1} \tag{2.12}$$

where the values of $up(\xi)$ and its derivatives can be found exactly in the form of rational numbers. At all other points of the compact support calculation of $up\,(\xi)$ can be done only approximately, but up to the computer accuracy. Although the calculation of $up\,(\xi)$ can use FT (2.10), Gotovac and

Kozulic [10] found a more appropriate expression for the calculation of $up(\xi)$ values in binary-rational points:

$$up(\xi_{br}) = \frac{2^{-m(m+1)/2}}{m!} \sum_{j=1}^{k} \delta_j \sum_{\ell=0}^{[m/2]} C_m^{2\ell} \cdot (2(k-j)+1)^{m-2\ell} \cdot a_{2\ell} \tag{2.13}$$

where δ_j are coefficients that have sign according to the following recursive formulas:

$$\delta_{2k-1} = \delta_k, \quad \delta_{2k} = -\delta_k, \quad k \in N, \quad \delta_j = 1 \tag{2.14}$$

$C_m^{2\ell}$ are binomial coefficients, expression $[m/2]$ represents the highest integer value, and $a_{2\ell}$ are even moments of $up(\xi)$ defined by the following recursive formulas:

$$a_{2k} = \frac{(2k)!}{2^{2k}-1} \sum_{\ell=1}^{k} \frac{a_{2k-2\ell}}{(2k-2\ell)!(2\ell+1)!}, \qquad k \in N \;\; ; \;\; a_0 = 1 \tag{2.15}$$

For the calculation of $up(\xi)$ values at arbitrary points, Gotovac and Kozulic [10] suggested a special series based on Taylor series of the $up(\xi)$ function at the binary-rational points ξ_{br} (because it is then a polynomial of the m-th order). Values of the even function $up(\xi)$ in arbitrary point $\xi \in [0\ ,\ 1]$ can be presented as follows:

$$up\,(\xi) = 1 - up(\xi - 1) = 1 - \sum_{k=1}^{\infty} \; (-1)^{1+p_1+\cdots+p_k} \; p_k \sum_{j=0}^{k} C_{jk} \cdot {\Delta_k}^j \tag{2.16}$$

where the coefficients C_{jk} are rational numbers containing values of $up(\xi)$ at the binary-rational points $\xi_k = -1 + 1/2^m$ (Gotovac and Kozulic [10]):

$$C_{jk} = \frac{1}{j!} 2^{j(j+1)/2} up(-1 + 2^{-(k-j)}) \; ; \quad j = 0\ ,\ 1\ ,\ \ldots\ ,\ k \;\; ; \;\; k = 1\ ,\ 2\ ,\ \ldots\ ,\ \infty \tag{2.17}$$

Factor Δ_k in (2.16) presents the difference between the real value of coordinate ξ and its binary presentation with k bytes, where $p_1 \ldots\ p_k$ are the digits 0 or 1:

$$\Delta_k = \xi - \sum_{i=1}^{k} p_i \cdot \frac{1}{2^i} \tag{2.18}$$

This means that the obtained accuracy of the coordinate ξ reflects on the accuracy of $up(\xi)$ values at arbitrary points, which in turn depends on the computer accuracy. For a chosen m, the calculation error of $up(\xi)$ values at the arbitrary points ξ (equal to the residual of series (2.16) when k= 1 , ... , m), does not exceed the value of $up\ (-1+2^{-m})$. In this work, we use 2^{16} binary-rational points (m=16), which means that the calculation error is of the order of $up\ \left(-1+2^{-16}\right)$ =$0.117 \cdot 10^{-51}$, which is significantly smaller than the computer accuracy. In practice, for all numerical calculations, it is sufficient to use (2.13) and 2^{16} binary-rational points (m=16) because its density enables interpolation of the value at any arbitrary point up to the computer accuracy.

For an exact description of polynomials up to the n-th order on the interval $\Delta\xi_n = 2^{-n}$, it is necessary to use 2^{n+1} basis functions obtained by shifting $up(\xi)$ for$\Delta\xi_n$. Such a relatively large number of basis functions imply poor approximation properties of $up\,(\xi)$. This is the main reason why application of $up(\xi)$ in numerical analysis for practical purposes is quite limited. $Fup_n(\xi)$ are another class of atomic basis functions, also belonging to the polynomial types of basis functions, which require only (n+2) basis functions to exactly describe polynomials up to the n-th order on interval $\Delta\xi_n = 2^{-n}$. For instance, for the development of a 4-th order polynomial, only 6 or (n+2) $Fup_4(\xi)$ are needed in comparison to 32 $up(\xi)$ basis functions. The compact support of $Fup_n(\xi)$ contains n+2 characteristic intervals $\Delta\xi_n = 2^{-n}$:

$$supp\ \ Fup_n(\xi) = \left[-\,(n+2)\cdot 2^{-n-1},\ (n+2)\cdot 2^{-n-1}\right] \tag{2.19}$$

For $n = 0$, the following holds:

$$Fup_0(\xi) = up(\xi) \tag{2.20}$$

Function $Fup_n(\xi)$ can be obtained by a convolution procedure using the contracted B_n and up basis function:

$$Fup_n(\xi) = B_n(2^n\ \xi) * up\ (2^{n+1}\ \xi) \tag{2.21}$$

This means that $Fup_n(\xi)$ is closely related to $B_n\ (\xi)$ and that they together

share all the mentioned properties. However, $Fup_n(\xi)$ has better approximation properties than $B_n(\xi)$ due to the convolution with the up function containing all orders of polynomials by parts and infinite continuity. Alternately, they share the same convergence properties because it is directly linked by the polynomial order which can be exactly described by these functions. Additionally, the better approximation properties of $Fup_n(\xi)$ are paid by one more characteristic interval for the same n-th order of basis functions. Atomic basis functions have a "deeper" mathematical background, and they are generally solutions of differential-functional equations, which for $Fup_n(\xi)$ take the following form:

$$Fup_n^{'}(\xi) = 2\sum_{k=0}^{n+2}\left(C_n^k - C_n^{k-2}\right)\cdot Fup_n\left(2\xi - \frac{k}{2^n} + \frac{n+2}{2^{n+1}}\right) \tag{2.22}$$

where C_n^k are binomial coefficients. The relationship between $Fup_n(\xi)$ and the shifted $Fup_{n+1}(\xi)$ can be presented in the general form:

$$Fup_n(\xi) = \frac{1}{2^{n+1}}\sum_{k=0}^{n+1} C_{n+1}^k \cdot Fup_{n+1}\left(\xi - \frac{k}{2^{n+1}} + \frac{n+1}{2^{n+2}}\right) \tag{2.23}$$

Equations (2.22-2.23) present the atomic structure of these basis functions because a function and its derivatives are decomposed by a linear combination of these same functions (Rvačev V.L. [9]). $Fup_n(\xi)$ can be calculated by a linear combination of $up(\xi)$ mutually shifted by the characteristic interval 2^{-n}:

$$Fup_n(\xi) = \sum_{k=0}^{\infty} C_k(n)\cdot up\left(\xi - 1 - \frac{k}{2^n} + \frac{n+2}{2^{n+1}}\right) \tag{2.24}$$

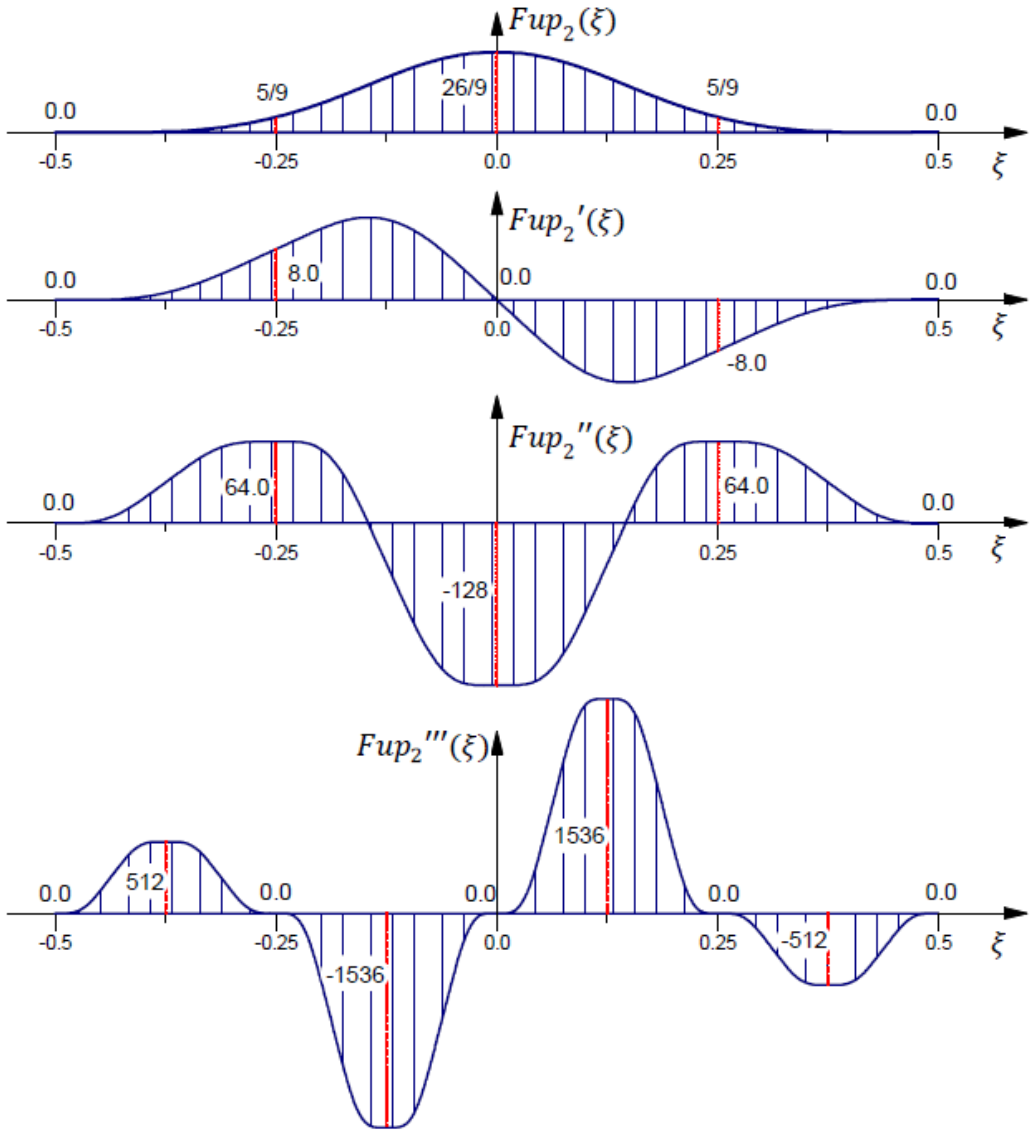


FIGURE 2.5: Function $Fup_2(\xi)$ with its first three derivatives.

The zero coefficient in (2.24) is:

$$C_0(n) = 2^{C_{n+1}^2} = 2^{n(n+1)/2} \tag{2.25}$$

Other coefficients are calculated in the form $C_k(n) = C_0(n) \cdot C_k'(n)$, where the coefficients $C_k'(n)$ are obtained using the following recursive formulas:

$$\begin{aligned} C_0'(n) &= 1 \\ C_k'(n) &= (-1)^k C_{n+1}^k - \sum_{j=1}^{min\{k;\,2^{n+1}-1\}} C_{k-j}'(n) \cdot \delta_{j+1} \end{aligned} \tag{2.26}$$

Derivatives of $Fup_n(\xi)$ are calculated from (2.22) and (2.24). Figure 2.5 presents $Fup_2(\xi)$ with its first three derivatives, which has the same number of characteristic intervals as $B_3(\xi)$. However, the third derivative has no discontinuities due to the enhanced continuity of $Fup_2(\xi)$.

Chapter 3

Isogeometric analysis (IGA) concept

This chapter serves as the preliminary derivation of a discretization technique used for groundwater flow modeling in karst, which is the main objective of Chapter 4. A short introduction to classical isogeometric analysis (IGA) is presented, followed by a description of three different numerical formulations used for spatial discretization of a simple diffusion problem. In addition to two the classical IGA formulations (Galerkin and collocation), a novel control volume IGA formulation is developed. The concept of IGA is presented as a unified framework for a multiscale description of the geometry, material heterogeneity and solution fields. The problem of steady-state groundwater flow through saturated heterogeneous porous medium is used to perform a comparison between different numerical formulations, and the novel control volume formulation is selected as the best candidate for the development of the karst flow model.

3.1 General overview of the IGA framework

Spline methods have attracted scientific attention since their discovery in the 1950s. However, the research has developed in two separate directions, one for CAD and geometric descriptions, and another for numerical analysis for solving the (initial-) boundary value problems. Hughes *et al.* [6] linked these two directions so that all variables, geometry coordinates and solutions are described by a linear combination of spline basis functions. For example, Figure 3.1 presents the isogeometric analysis (IGA) scheme of how one 2D subdomain or patch (defined as a general four-sided object) is transformed from the parameter space (virtual domain) to the physical space (real domain) using the following spline representation:

$$x(\xi,\eta) = \sum_{j=1} x_j \varphi_j(\xi,\eta) \;\; ; \;\; y(\xi,\eta) = \sum_{j=1} y_j \varphi_j(\xi,\eta) \tag{3.1}$$

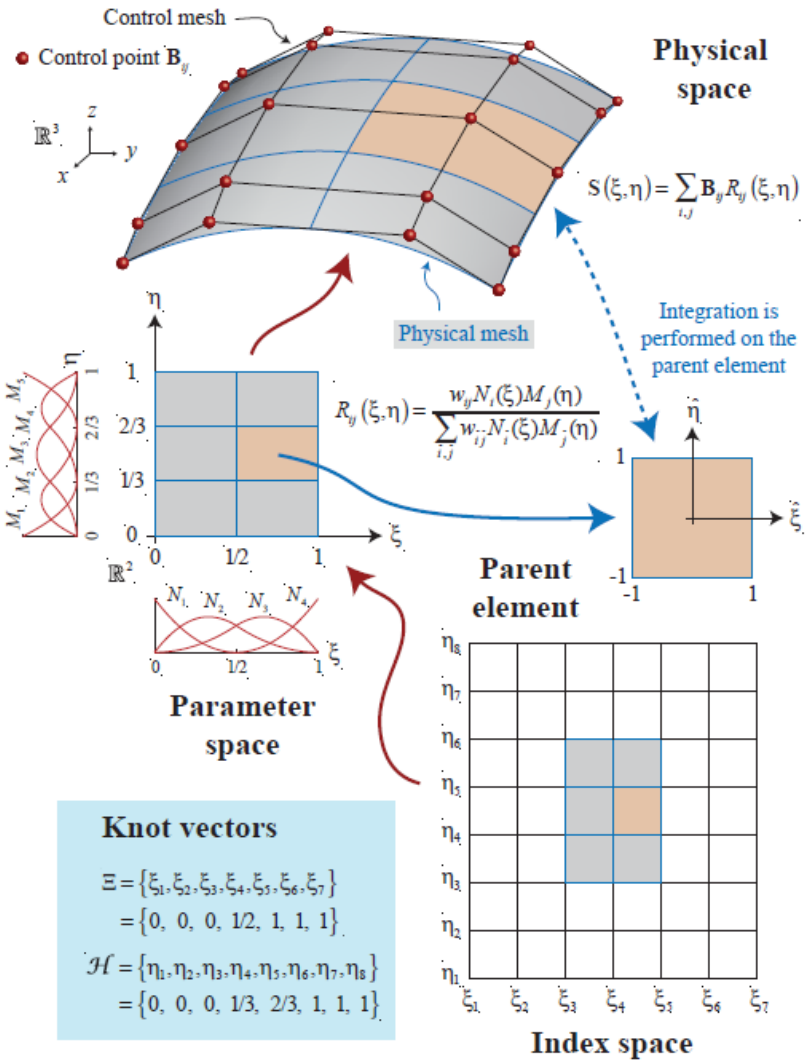


FIGURE 3.1: Schematic illustration of isogeometric analysis (IGA): physical space with control mesh and control points, parameter space with spline basis functions and related parent elements, knot vectors, and index space (taken from Cottrell *et al.* [8]).

where x_j and y_j are the coordinates of the control points $\mathbf{B}(x_j, y_j)$ in the physical space, while ξ and η are the coordinates in the parameter space. The control points define the control mesh, which enables the designer to create a wide range of desired objects, for instance, in the car or aviation industry. The key elements of (3.1) are spline basis functions (φ_j). In classic IGA, B-splines and NURBS are widely recognized as suitable spline basis functions with the following main properties: a) compact support, implying localized

approximation properties; b) strictly positive values, implying positive definiteness; c) partition of unity; d) continuity up to the desired order; and e) changeable length of compact support, implying multiresolution or multiscale representation. Multidimensional spline basis functions are constructed via the tensor product of the basis functions from each direction [8]. Expression (3.1) shows that IGA operates only with basis functions in the virtual regular domain since transformations from the real to virtual domain, and vice versa, are defined by the Jacobian, and its inverse, as in classic FEM. The main difference is that the FEM performs transformations for each element, while IGA considers the transformation of each patch, which can be thought of as a macro-element or a subdomain [8].

The numerical solution is also described in the virtual domain by an independent set of spline basis functions:

$$u\left(\xi,\eta\right)=\sum_{j=1}\alpha_j\varphi_j\left(\xi,\eta\right) \tag{3.2}$$

Notably, the number and order of the basis functions may not be the same in (3.1) and (3.2).

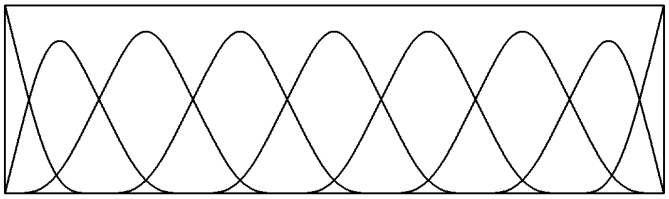


FIGURE 3.2: $Fup_1(x)$ basis function trial space.

In this work, only simple geometries are considered. Thus, for simplicity, the numerical solution will be expressed directly in the physical space:

$$u(\mathbf{x})=\sum_{j=1}\alpha_j\varphi_j(\mathbf{x})=\alpha_j\varphi_j(\mathbf{x}) \tag{3.3}$$

where Einstein summation notation $\sum_j \alpha_j\varphi_j(\mathbf{x}) = \alpha_j\varphi_j(\mathbf{x})$ will be used in the rest of this work. Once spline coefficients α_j are known, the m-th derivative of the numerical solution is easily calculated by simply differentiating the basis functions, as shown in the following expression:

$$\nabla^m u(\mathbf{x})=\alpha_j\nabla^m\varphi_j(\mathbf{x}) \tag{3.4}$$

The trial function space of $Fup_1(x)$ basis functions is shown in Figure 3.2, and multidimensional Fup basis functions are obtained as tensor products of

the one-dimensional basis functions. Figure 3.3 shows the two-dimensional $Fup_1(x, y)$ basis function and one of its partial derivatives.

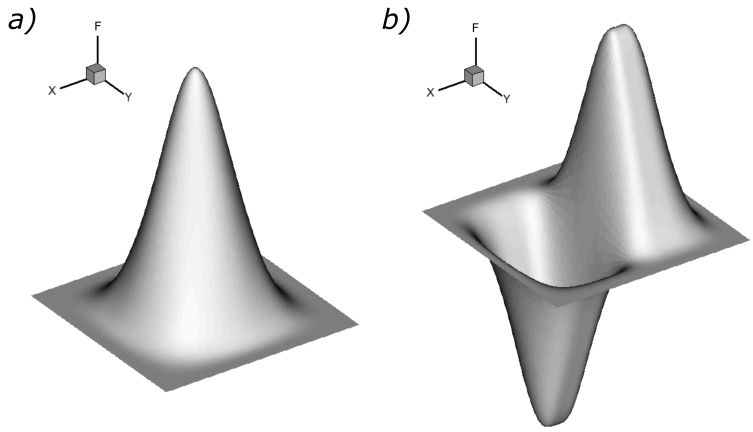


FIGURE 3.3: 2D Fup basis function $F = Fup_1(x, y)$: a)F; b)$\frac{\partial F}{\partial x}$.

In the following, specific information regarding the basis function properties, such as convergence order and number of nonzero elements, will be stated only for Fup basis functions. Generally, the convergence order of both basis functions is determined by their polynomial orders n and is the same for the B_n and Fup_n basis functions. The numbers of nonzero elements are not the same because the numbers of characteristic intervals are different (see Chapter 2). Thus, the number of nonzero elements for B_n is the same as that for Fup_{n-1}.

3.2 Three IGA formulations

The discretization process will be presented by considering a simple steady-state diffusion equation in the form:

$$\nabla \cdot (D(\mathbf{x})\nabla u(\mathbf{x})) = 0 \quad \text{in} \quad \Omega \tag{3.5}$$

with appropriate boundary conditions:

$$u(\mathbf{x}) = u_D(\mathbf{x}) \quad \text{on} \quad \Gamma_D \tag{3.6}$$

$$(D(\mathbf{x})\nabla u(\mathbf{x})) \cdot \mathbf{n} = q_N(\mathbf{x}) \quad \text{on} \quad \Gamma_N \tag{3.7}$$

where $u(\mathbf{x})$ is the (scalar) dependent variable, $D(\mathbf{x})$ is a general diffusion function, $\mathbf{n}$ is the outward normal vector, Ω is the domain of interest, and

Γ_D and Γ_N are the domain boundaries under the Dirichlet and Neumann boundary conditions, respectively.

A general approach that can be used to derive the different numerical formulations is the method of weighted residuals. The idea is to integrate differential equation (3.5) over the domain of interest and multiply it by a finite number of weighting (test) functions $w_i(\mathbf{x})$:

$$\int_{\Omega} w_i(\mathbf{x}) \nabla \cdot (D(\mathbf{x}) \nabla u(\mathbf{x})) d\Omega = 0 \tag{3.8}$$

The number of test functions is generally the same as the number of basis functions, so the number of equations is equal to the number of unknowns. One way to derive the three formulations presented in this work is to use different weighting functions, as will be presented in the following.

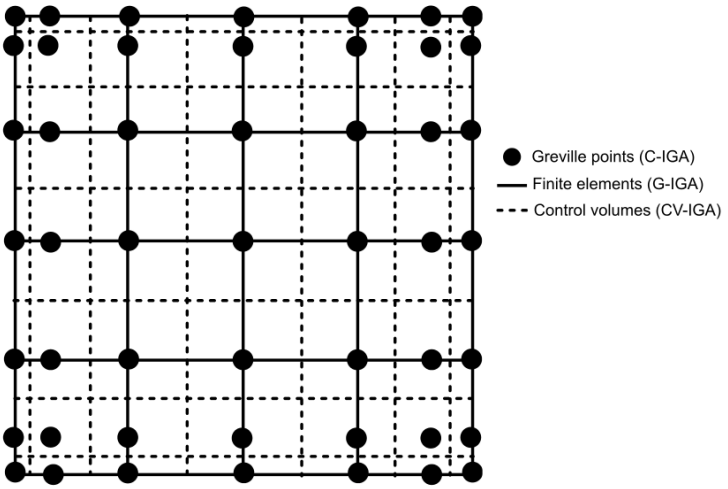


FIGURE 3.4: Discretization of 2D domain with three different IGA forumations.

3.2.1 Galerkin (G-IGA)

The classical Galerkin formulation is obtained by using the same functions to construct both the trial (basis) and test (weight) function spaces. In Galerkin-IGA, the spline basis functions (φ_j) used for the solution and geometry description are also used as the weighting functions. Thus, setting $w_i(\mathbf{x}) = \varphi_i(\mathbf{x})$ and substituting into equation (3.8) produces:

$$\int_{\Omega} \varphi_i(\mathbf{x}) \nabla \cdot (D(\mathbf{x}) \nabla u(\mathbf{x})) d\Omega = 0 \tag{3.9}$$

By applying integration by parts and divergence (Gauss-Ostrogradsky) theorem, the weak form of (3.9) is obtained:

$$\int_{\Gamma_N} \varphi_i(\mathbf{x})(D(\mathbf{x})\nabla u(\mathbf{x})) \cdot \mathbf{n} d\Gamma - \int_{\Omega} \nabla \varphi_i(\mathbf{x}) \cdot (D(\mathbf{x})\nabla u(\mathbf{x})) d\Omega = 0 \quad (3.10)$$

In the first term, the integration is performed over Γ_N instead of whole Γ. This is the standard approach also used in classical FEM, where the test function space is restricted and expected to vanish on Γ_D (e.g., [7], [8]). However, boundary conditions must be imposed to obtain a well-posed problem. The Neumann boundary conditions are satisfied in the classical weak sense by simply replacing $(D\nabla u) \cdot \mathbf{n}$ in (3.10) with the defined boundary condition function q_N (3.7) to obtain:

$$\int_{\Gamma_N} \varphi_i(\mathbf{x}) q_N(\mathbf{x}) d\Gamma - \int_{\Omega} \nabla \varphi_i(\mathbf{x}) \cdot (D(\mathbf{x})\nabla u(\mathbf{x})) d\Omega = 0 \quad (3.11)$$

In the classical FEM, the Dirichlet boundary conditions are satisfied by imposing solution values in the finite element nodes (strong imposition). While the classical FEM uses Lagrangian basis functions, which are collocated in finite element nodes, the higher-order spline basis functions in IGA do not possess such property because there are always few nonzero basis functions at each point on the Dirichlet boundary (exceptions are 1D problems). In this work, the Dirichlet boundary conditions for G-IGA are implemented so that spline approximation u exactly satisfies Dirichlet boundary conditions u_D at the appropriate boundary points.

Finally, by expressing solution $u(\mathbf{x})$ as a linear combination of spline basis functions ($u(\mathbf{x}) = \alpha_j \varphi_j(\mathbf{x})$), the weak form (3.11) and Dirichlet boundary equation (3.6) can be presented in their final discrete forms:

$$\int_{\Gamma_N} \varphi_i(\mathbf{x}) q_N(\mathbf{x}) d\Gamma - \alpha_j \int_{\Omega} \nabla \varphi_i(\mathbf{x}) \cdot (D(\mathbf{x})\nabla \varphi_j(\mathbf{x})) d\Omega = 0, \ \mathbf{x} \in (\Omega \cup \Gamma_N) \quad (3.12)$$

$$\alpha_j \varphi_j(\mathbf{x}_B) = u_D(\mathbf{x}_B), \ \mathbf{x}_B \in \Gamma_D \quad (3.13)$$

The maximum number of nonzero basis functions (φ_j) for a given discretized equation (related to the i-th test function φ_i) is determined by the polynomial order (n) of the basis functions. For the Fup_n basis functions,

this number is defined by $(2n+3)^{dim}$, where dim denotes the dimensionality of the problem. The implementation details of IGA generally follow the architecture used in the FEM [8]. Thus, the assembly of the system is performed by looping over elements (see Figure 3.4) instead of looping through all the global shape functions. The Gaussian quadrature is typically used to perform numerical integration over each element. The differences from the FEM are that the control points are not necessarily located at the element corners (see Figure 3.1) and the fact that the control variables (coefficients of the linear combination α_j) cannot be interpreted as nodal values due to the non-interpolatory nature of spline basis functions.

3.2.2 Collocation (C-IGA)

The collocation approach is conducted by using the Dirac delta function as the weighting function, i.e., by setting $w_i(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_i)$, which produces:

$$\int_{\Omega} \delta(\mathbf{x} - \mathbf{x}_i) \nabla \cdot (D(\mathbf{x}) \nabla u(\mathbf{x})) d\Omega = 0 \tag{3.14}$$

Due to the properties of the Dirac delta function, the demanding integration procedure is eliminated. Thus, the outcome of this approach is the strong differential form, which satisfies the governing PDE (3.5) at each i-th internal collocation point:

$$\nabla \cdot (D(\mathbf{x}_i) \nabla u(\mathbf{x}_i) = 0 \tag{3.15}$$

Whereas the weak Galerkin form (3.10) required only C^1 continuity, the strong collocation formulation (3.15) requires at least C^2 continuity of the basis functions. This is evident by differentiating equation (3.15) to obtain:

$$\nabla D(\mathbf{x}_i) \cdot \nabla u(\mathbf{x}_i) + D(\mathbf{x}_i) \nabla^2 u(\mathbf{x}_i) = 0 \tag{3.16}$$

Thus, in the general case (with an exception for $D(\mathbf{x}) = const$), the strong collocation form of diffusion equations (3.15) becomes the advection-diffusion equations (3.16). Moreover, this change requires calculation diffusion gradients, which is possible by using spline functions for the description of the diffusion function, as will be described in section 3.3. In the case of high material heterogeneities (represented by the diffusion function, as will be presented in the following), the diffusion gradients are expected to be significantly larger than the diffusion values. Therefore, problem (3.16) becomes an advection-dominated problem, a problem that is usually subject to

additional numerical challenges in terms of numerical stability and accuracy. Despite these shortcomings, Schillinger *et al.* [19] proved that C-IGA can be competitive with G-IGA in terms of computational cost for a given accuracy, especially for higher-order spline basis functions. This is mostly due to the fact that a collocation formulation avoids costly numerical integration and the fact that the number of nonzero elements is significantly reduced compared with the Galerkin formulation. The number of nonzero elements for the discretized i-th equation is $(n+2)^{dim}$ for odd order and $(n+1)^{dim}$ for even order basis functions, where n is the order of the Fup_n basis functions.

The collocation formulation satisfies the boundary conditions in the strong sense, which means that both the Dirichlet and Neumann conditions are satisfied exactly. The discretized diffusion equations (3.5) and corresponding boundary conditions (3.6-3.7) in the collocation formulation are given by:

$$\nabla D(\mathbf{x}_i) \cdot \alpha_j \nabla \varphi_j(\mathbf{x}_i) + D(\mathbf{x}_i)\alpha_j \nabla^2 \varphi_j(\mathbf{x}_i) = 0,\ \mathbf{x}_i \in \Omega \tag{3.17}$$

$$\alpha_j \varphi_j(\mathbf{x}_i) = u_D(\mathbf{x}_i),\ \mathbf{x}_i \in \Gamma_D \tag{3.18}$$

$$(D(\mathbf{x}_i)\alpha_j \nabla \varphi_j(\mathbf{x}_i)) \cdot \mathbf{n} = q_N(\mathbf{x}_i),\ \mathbf{x}_i \in \Gamma_N \tag{3.19}$$

The choice of the collocation points is one of the most important components for the success of the collocation method. The selection has direct influence on the accuracy and stability. In recent years, considerable progress has been made on this topic, especially within C-IGA, and various sets of collocation points, such as the Greville abscissae, knot maxima and Demko points, have been proposed. The Greville abscissae have been widely adopted as the default choice because of their simple definition and because they have been shown to be stable in most cases (e.g., [19], [20]). The Greville abscissa in each spatial direction can be easily computed from a knot vector $G = \{g_1, g_2, \ldots, g_{k+n+1}\}$ as:

$$\hat{g}_i = \frac{g_{i+1} + \ldots + g_{i+n}}{n},\ i = 1, \ldots, k \tag{3.20}$$

where k is the number of basis functions for a particular direction. In this work, the Greville points are used as collocation points. For 2D domain the Greville points are shown in the Figure 3.4.

3.2.3 Control volume (CV-IGA)

The control volume formulation is obtained by subdividing computational domain Ω into a set of nonoverlapping control (finite) volumes Ω_i, such that $\Omega = \cup_i \Omega_i$. By setting the weighting function:

$$w_i = \begin{cases} 1 & \mathbf{x} \in \Omega_i \\ 0 & \mathbf{x} \notin \Omega_i \end{cases} \tag{3.21}$$

and substituting into (3.8), the direct integral form of the governing equation is recovered:

$$\int_{\Omega_i} \nabla \cdot (D(\mathbf{x}) \nabla u(\mathbf{x})) d\Omega = 0 \tag{3.22}$$

Note that integration is performed only over the i-th control volume (CV) due to the properties of the test functions (3.21). Finally, the conservative control volume formulation is derived by applying the divergence theorem, which produces:

$$\int_{\Gamma_i} (D(\mathbf{x}) \nabla u(\mathbf{x})) \cdot \mathbf{n} d\Gamma = 0 \tag{3.23}$$

where Γ_i is the boundary of the i-th CV.

In this work, the control volumes are defined in a manner such that each CV edge lies in the middle between two Greville points (see Figure 3.4). This means that the numbers of basis functions and CVs are the same, and there are no additional equations for the boundary conditions. Thus, the CVs with faces coinciding with the domain boundary require special treatment. The equations for these CVs have to be modified to incorporate the boundary condition contributions, and the weak imposition is used for both the Dirichlet and Neumann boundaries. The implementation will be demonstrated by considering the conservative form (3.23).

Dirichlet boundaries

The weak imposition of the Dirichlet boundary conditions can be obtained by the addition of an extra term to (3.23):

$$\int_{\Gamma_i} (D(\mathbf{x}) \nabla u(\mathbf{x})) \cdot \mathbf{n} d\Gamma + \sigma \int_{\Gamma_{D_i}} (u(\mathbf{x}) - u_D(\mathbf{x})) d\Gamma = 0 \tag{3.24}$$

where Γ_{D_i} represents the part of the i-th CV that belongs to the Dirichlet boundary, i.e., $\Gamma_{D_i} = \Gamma_D \cap \Omega_i$, and σ is the penalty coefficient, which is normally set to 1. Using a large value for the penalty coefficient will produce a similar effect as if a boundary condition was strongly imposed. The ability to satisfy the Dirichlet boundary condition approximately can be a significant advantage if it allows for greater accuracy in the interior of the domain and, in some cases, can help eliminate some of the spurious oscillations encountered with traditional strongly imposed conditions [8].

Neumann boundaries

Equation (3.23) for a boundary CV with Neumann boundary conditions can be written in the form:

$$\int_{\Gamma_i^*} (D(\mathbf{x})\nabla u(\mathbf{x})) \cdot \mathbf{n} d\Gamma + \int_{\Gamma_{N_i}} q_N(\mathbf{x}) d\Gamma = 0 \tag{3.25}$$

where $\Gamma_i^* = \Gamma_i \backslash \Gamma_{N_i}$, $\Gamma_{N_i} = \Gamma_N \cap \Gamma_i$, and q_N is the Neumann boundary condition function according to (3.7). The result of weak imposition is that the boundary conditions are never enforced exactly. However, the accuracy is excepted to improve, together with the solution in the interior of the domain, as the mesh is refined. This approach is known to be superior when highly nonlinear boundary conditions must be imposed.

Finally, after expressing solution u as a linear combination of spline basis functions ($u = \alpha_j \varphi_j$), the general discretized control volume equation, which accounts for possible boundary condition modifications, can be written in the following form:

$$\int_{\Gamma_i^*} (D(\mathbf{x})\alpha_j \nabla \varphi_j(\mathbf{x})) \cdot \mathbf{n} d\Gamma + \int_{\Gamma_{N_i}} q_N(\mathbf{x}) d\Gamma + \sigma \int_{\Gamma_{D_i}} (\alpha_j \varphi_j(\mathbf{x}) - u_D(\mathbf{x})) d\Gamma = 0 \tag{3.26}$$

An attractive feature of the control volume formulation is conservation. Thus, the conservation is exactly satisfied over any CV (local conservation), as well as over the whole computational domain (global conservation). In such case, even the coarse-mesh solution exhibits an exact integral balance (Patankar [1]). Moreover, control volume formulation (3.23) requires only integration over CV boundaries Γ_i, while the Galerkin formulation (see eq. 3.11) requires full integration over the part of domain defined by the particular test function. Thus, CV-IGA generally requires cheaper numerical

integrations since the boundary has lower dimensionality than its domain. Moreover, the number of nonzero basis functions for each discretized equation is $(n+2)^{dim}$ for odd order and $(n+3)^{dim}$ for even order basis functions, which means that the cost for the solution of the system of equations is generally lower than that for Galerkin-IGA.

3.3 Multiscale representation of heterogeneity and flow variables

As described in the previous sections, the IGA offers an interesting possibility to describe the geometry and solution variables as continuous and smooth numerical representations by using linear combinations of spline basis functions. In this section, an additional possibility is introduced because the spline basis can be used to describe different fields (such as material heterogeneity given as a general diffusion function $D(\mathbf{x})$). This process can be especially beneficial when information is given as discrete data values and the numerical procedure seeks accurate and stable interpolations. Moreover, as presented in the previous section, the collocation formulation generally requires derivatives of such fields, which are not always easy to calculate, even in the case when these fields are known as (complex) analytical functions (which is rarely the case). Thus, the spline representation is presented as an efficient tool for the description of different fields as smooth and continuous functions.

Moreover, an additional advantage of using such spline representations is the ease of constructing a multiscale approach. As already mentioned for the geometry and solution, the numbers of basis functions used for describing different fields do not need to be the same. Thus, the geometry, material properties, and solution fields are naturally represented in the multiscale sense by using different numerical resolutions.

The construction of the such spline representation will be described by considering the hydraulic conductivity field, which is important input data for realistic groundwater flow modeling through heterogeneous porous media. The groundwater flow equation is the same type as the considered diffusion equation (3.5) and is given in the form:

$$\nabla \cdot (K(\mathbf{x}) \nabla h(\mathbf{x})) = 0 \tag{3.27}$$

where h is the hydraulic (piezometric) head (L) and K is the hydraulic conductivity function (L/T). The spline representation of such a field is given in standard spline form:

$$K(\mathbf{x}) = \beta_k \phi_k(\mathbf{x}) \tag{3.28}$$

The coefficients β_k can be calculated by using different formulations, and ϕ_k are the standard spline basis functions. In this work, the control volume formulation is presented. Again, the domain is subdivided into a set of nonoverlapping CVs, and an integral equation of each CV has the form:

$$\beta_k \int_{\Omega_i} \phi_k(\mathbf{x}) d\Omega = \int_{\Omega_i} K(\mathbf{x}) d\Omega \tag{3.29}$$

The spline projection of the field K is obtained by writing equations for all the CVs and solving system of equations for unknown spline coefficients β_k.

Figure 3.5 presents an example of spline description for a given material heterogeneity using different numerical resolutions. The field is generated by Hydrogen code [21] at the fourth resolution level as a set of discrete cell average values (Figure 3.5f), while Figures 3.5a-e represent the spline approximation of the field by using different numerical resolutions.

The coarsest zero level (Figure 3.5a) describes only the global conductivity patterns, while each higher level (Figure 3.5b-e) captures finer heterogeneity scales. Contrary to the classical numerical approaches (such as the FEM or FVM), where the heterogeneity field is discontinuous and described by using the same numerical resolution as that used for the discretization of PDEs, this work presents a different approach that is closely related to the physical definition and in situ measurements.

The heterogeneity of porous media can be defined from the pore scale to a larger macro scale. A fundamental concept of the representative elementary volume (REV) is that it is the smallest volume (consisting of a certain number of pores and grains) that can be represented by the average value in terms of a continuous function. The REV is a scale larger than the pore scale but significantly smaller than the macro scale. Each larger scale includes some effects related only to that scale and changes the representative average value at some specific location. By contrast, our equipment always requires a certain scale, and the obtained measurements represent an average value on that scale. Moreover, different measurements give different average values on different scales.

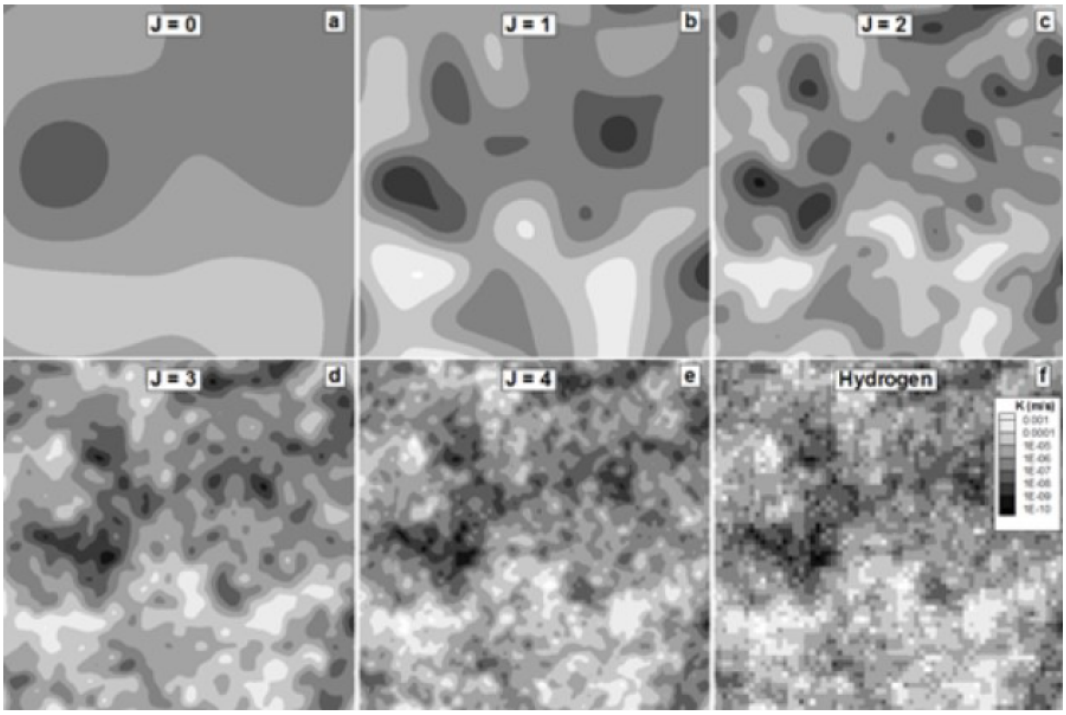


FIGURE 3.5: Multiscale description of the material heterogeneity (hydraulic conductivity field) generated by Hydrogen [21] at the fourth resolution level as a set of discrete cell average values.

Thus, in this work, spline basis functions are used to represent the heterogeneity field as a smooth and continuous function. The important advantage of this approach is the representation of the groundwater velocity field. Usually, the velocity is expressed by Darcy's law:

$$q(\mathbf{x}) = -K(\mathbf{x})\nabla h(\mathbf{x}) \tag{3.30}$$

Classical FEA produces a discontinuous velocity field due to the C^0 property of the numerical solution, even in the case of a homogeneous aquifer. By using the smooth spline approximation for the hydraulic head, the derivatives are continuous. However, for a heterogeneous conductivity field represented as a piecewise constant value over each element (classical approach in the FEM or FVM), the velocity field is discontinuous due to discontinuities in the conductivity field. By using spline basis functions to describe both the conductivity ($K(\mathbf{x}) = \beta_k \phi_k(\mathbf{x})$) and hydraulic head ($h(\mathbf{x}) = \alpha_j \varphi_j(\mathbf{x})$) fields, the velocity is defined as a smooth and continuous function at any point in the domain:

$$q(\mathbf{x}) = -\beta_k \phi_k(\mathbf{x}) \alpha_j \nabla \varphi_j(\mathbf{x}) \tag{3.31}$$

This approach eliminates "unphysical" velocity discontinuities and avoids the additional numerical errors and inaccurate velocity field needed for transport simulations. Finally, the numerical resolutions for the descriptions of the geometry, heterogeneity and solution are independent, and solution refinement can be performed without affecting the geometry or material heterogeneity.

3.4 Numerical examples

To verify the methodology, a classic 2D benchmark of the flow in a saturated multi-Gaussian heterogeneous porous media under mean uniform flow conditions is considered (see, e.g., [22]–[24]). Figure 3.6 presents such a realization with two different heterogeneity ($Y = lnK$) log-variances: $Var(lnK) = 1$ (Figure 3.6a) and $Var(lnK) = 8$ (Figure 3.6b). Hydraulic conductivity is generated on the fourth level (256×128 cells) with four control volumes per correlation length. The boundary conditions create mean uniform flow conditions in the following way: the top and bottom are no-flow Neumann conditions, and the left and right boundaries have Dirichlet boundary conditions with head equal to 10 (m) and 0 (m), respectively.

Figure 3.6 presents the log-conductivity field as the filled background, as well as the hydraulic head (red contours) and velocity solution (black polylines), also at the fourth level for both lnK variances. The solution is obtained by CV-IGA and the Fup_2 basis functions. The homogeneous media yields a uniform flow solution with a linear distribution of the head and horizontal streamlines. Heterogeneity causes deviation of both families of curves. The weakly heterogeneous porous media exhibits small heterogeneity variations, implying small velocity variations and a relatively uniform flow field (Figure 3.6a). The highly heterogeneous solution is characterized by preferential flow channels, which occupy a small portion of the domain but transfer most of the total flow discharge (Figure 3.6b). The flow field tends to connect higher conductivity zones to spend minimum flow energy. This Figure clearly shows how complexities of the velocity field arise with variation in heterogeneity, introducing large head and velocity gradients.

The head can be considered for all levels, independent of the conductivity resolution, from the chosen zero level (in this case, the zero level is chosen to be 16×8 cells) to the maximum user-defined level (2048×1024 cells), using an independent linear combination of Fup_2 basis functions. The zero-coarse level usually describes only the global flow patterns, while each higher level

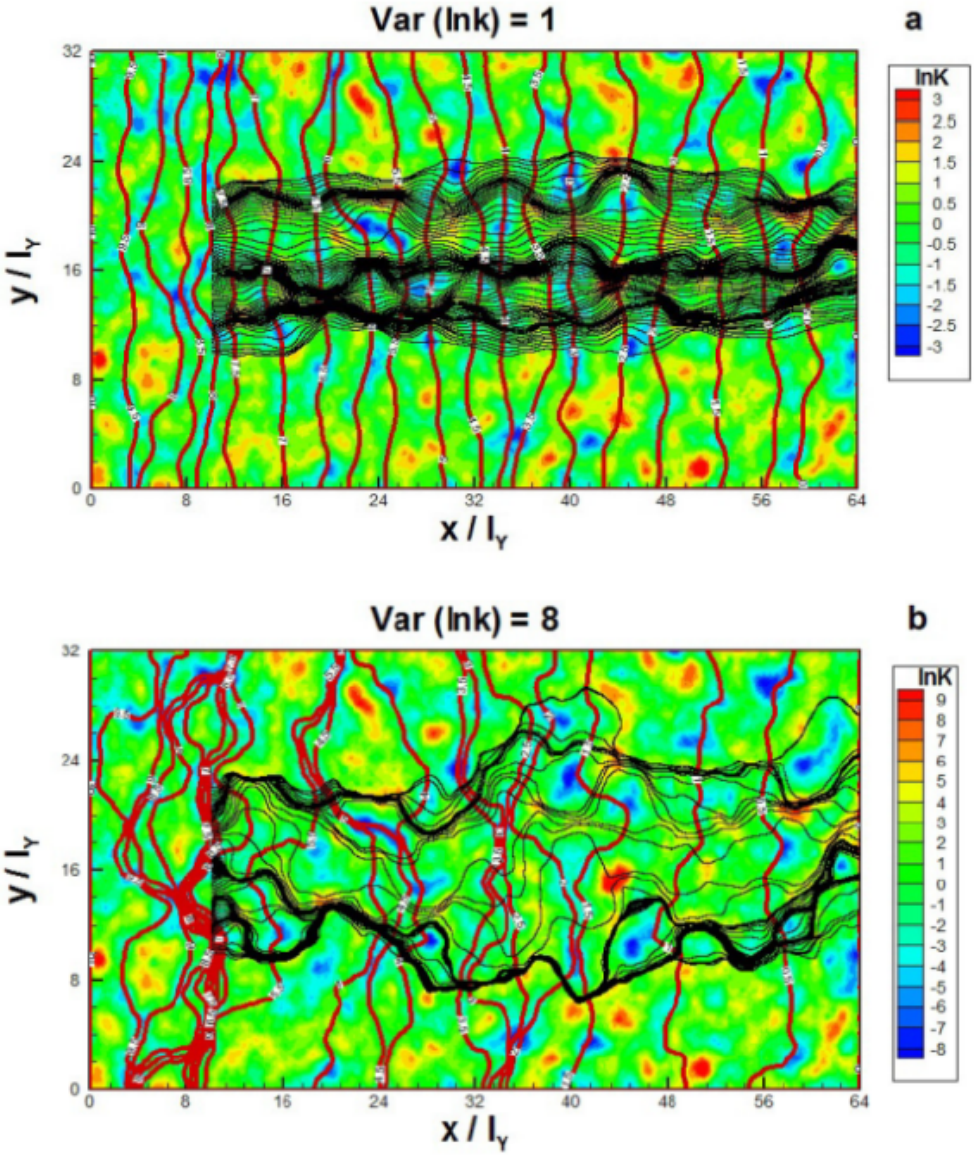


FIGURE 3.6: CV-IGA solution for flow in heterogeneous porous media defined by isotropic exponential covariance and mean uniform flow conditions for a) $Var(lnK) = 1$, b) $Var(lnK) = 8$. The head solution is represented by red contours, streamlines are represented by black polylines, and the conductivity field is represented by the filled background.

includes more detailed velocity patterns, completely resolving the influence of all the heterogeneity scales.

Figure 3.7 presents a head solution at different resolution levels (for log-conductivity at the fourth level presented in Figure 3.6b). For the high lnK variance field generated at the fourth level, the head solution at the second level (Figure 3.7a) is in relatively close agreement with that of the fourth level. However, inside the control volumes at the second level, there are zones with

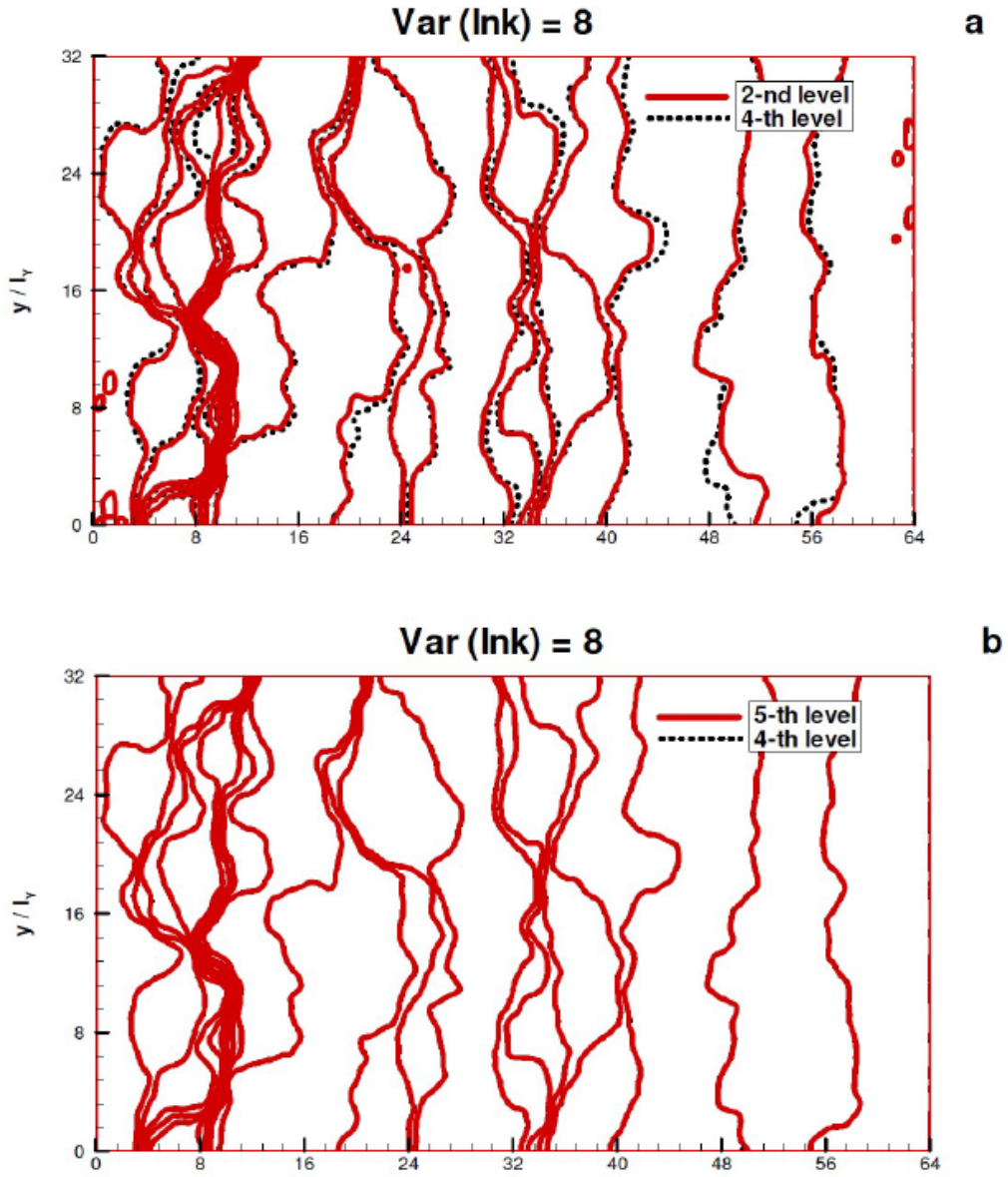


FIGURE 3.7: CV-IGA solution of the head field at three different resolution levels for conductivity defined at the fourth level with an $Var(lnK) = 8$ (shown in the Figure 3.6b).

stagnant streamlines (artificial sink/source zones), implying that the numerical resolution is not sufficiently fine to avoid oscillatory behavior for highly heterogeneous cases. The control volume formulation exactly satisfies the mass balance through the cell boundaries at each resolution level, but the large heterogeneity variations at the fourth level cause a large hydraulic gradient (head derivatives) involving numerical oscillations at the second level. This is not the case for weak and mild heterogeneity, where even coarser levels can produce accurate and stable head and velocity fields. However, for high heterogeneity, it is not possible to obtain an accurate velocity field

on the coarser levels. Figure 3.7b confirms that the fourth level produces an accurate head solution because its comparison to the fifth level shows a negligible difference. The head difference between the fourth and fifth levels is less than 0.1 mm (head is bounded in the interval [0, 10 m]), which, when compared with the larger differences obtained when comparing consecutive coarser levels, means that the solution converges monotonically toward a grid-independent solution.

The accuracy and convergence analysis are studied in greater detail in Figures 3.8 and 3.9 for the head and velocity solutions, respectively. A one-dimensional analogy of the presented two-dimensional example (for higher heterogeneity variances $Var(lnK) = 8$ shown in Figure 3.6b) is used to compare three different IGA formulations (Galerkin: G-IGA, Collocation: C-IGA, and Control volume: CV-IGA) and four different orders of B-spline and Fup basis functions ($p = 1, 2, 3, 4$). The Figures demonstrate the dependence of grid spacing Δx with respect to the L_2 error norm, and the slopes of the simulated lines represent the convergence rate (p) of each method.

A similar analysis is performed many times for the Poisson equation for G-IGA and C-IGA (among others Schillinger *et al.* [19]). In the case of a homogeneous aquifer, groundwater flow problem (3.27) is equal to the Poisson equation. G-IGA yields the optimal convergence rate for the Poisson problem, which is equal to $p = n + 1$ if n is the order of the basis functions. Figure 3.8a-d confirms that this result holds, even for the high heterogeneity case $Var(lnK) = 8$ and higher-order basis functions up to the 4-th order. C-IGA yields suboptimal convergence rates of $p = n$ for even and $p = n - 1$ for odd basis functions. C-IGA is less expensive and more efficient than G-IGA; however, it suffers from three basic drawbacks: 1) the absence of an optimal convergence rate and, therefore, reduced accuracy, 2) the groundwater flow (diffusion) problem is regarded as an advection-diffusion equation through the strong formulation (see section 3.2) involving higher-order derivatives and possible numerical instabilities for high heterogeneity cases, and 3) there is no local and global mass balance due to the collocation nature of the algorithm. Gomez and Lorenzis [25] proved that Greville collocation points are not optimal convergence points; instead, they found superconvergent collocation points, which yield an optimal convergence rate for odd basis functions ($p = n + 1$). Still, for even basis functions, the convergence rate is not optimal ($p = n$) because the superconvergent points coincide with the Greville points in this case. The reason for this "phenomenon" is still not well understood.

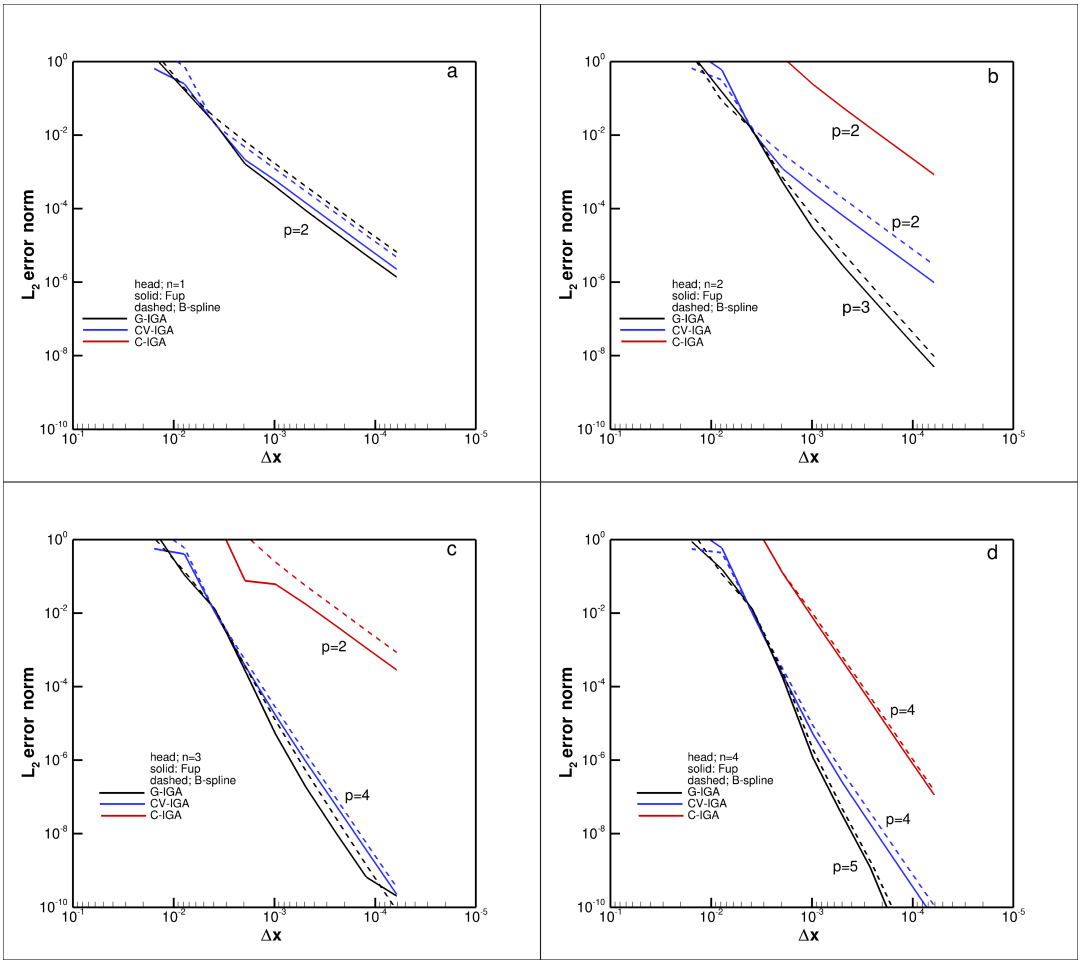


FIGURE 3.8: Convergence analysis of the head field for four different orders of basis functions and conductivity realization defined at the fourth level with the $Var(lnK) = 8$.

CV-IGA lies between G-IGA and C-IGA with respect to computational efficiency and accuracy, as shown in Figure 3.8. Figure 3.8 shows that the convergence rate for CV-IGA is optimal ($p = n + 1$) for odd and suboptimal ($p = n$) for even basis functions, the same as C-IGA with superconvergent collocation points. Again, the "mystery" of even basis functions is also present for CV-IGA. Figure 3.8a shows that for first-order basis functions, the convergence rate is optimal for both G-IGA and CV-IGA, and there are no collocation results because C-IGA requires at least C^2 continuity or continuous second-order derivatives that are satisfied with B_3 spline basis functions. The Fup_1 basis functions possess such continuity; however, there are also no collocation results because the second derivatives of the Fup_1 basis functions are zero at the Greville points.

Figure 3.8b-d presents the convergence results for higher-order basis functions. Due to the convergence order reduction, C-IGA and CV-IGA are

less accurate than G-IGA for even order basis functions; however, CV-IGA is approximately two orders of magnitude more accurate than C-IGA. For odd order basis functions, CV-IGA has the optimal convergence rate with accuracy very close to that of G-IGA. Even though detailed efficiency analysis was not performed, it is important to emphasize that CV-IGA is a less expensive numerical procedure due to the reduced number of nonzero matrix entries ($2n+3$ for G-IGA vs. $n+2$ for CV-IGA per discretized equation) and cheaper numerical integrations (integration over element of dimension dim for G-IGA vs. integration over the CV boundary of dimension $dim-1$ for CV-IGA).

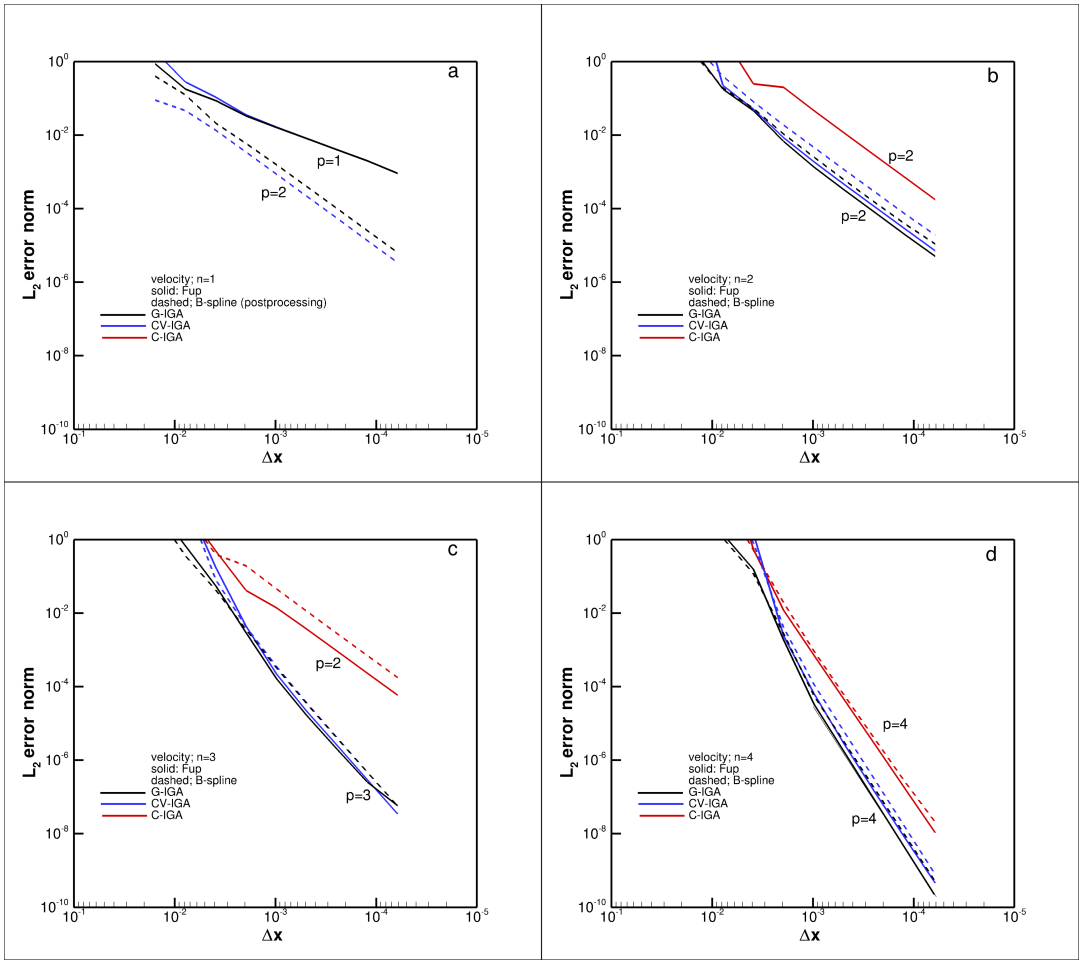


FIGURE 3.9: Convergence analysis of the velocity field for four different orders of basis functions and conductivity realization defined at the fourth level with the $Var(lnK) = 8$.

Figure 3.9 presents the convergence and accuracy analysis for the velocity calculated from Darcy's law (3.30). Generally, C-IGA yields the optimal convergence rate $p = n$ for even basis functions and suboptimal convergence

rate $p = n - 1$ for odd basis functions. CV-IGA yields the same optimal convergence rate as G-IGA for all orders of basis functions and has very similar results to G-IGA. G-IGA and CV-IGA generally produce much more accurate velocity results than C-IGA. The velocity results with the B_1 spline for G-IGA and CV-IGA are obtained by post-processing, which is the usual approach in the FEM. In that case, an enhanced convergence rate for velocity is obtained, but the processing requires additional CPU time in comparison to the classic IGA due to the additional velocity approximation.

Finally, CV-IGA represents the best candidate for the velocity approximation of groundwater flow modeling in heterogeneous porous media for the following reasons: 1) optimal velocity convergence rate and accuracy very close to that of G-IGA at a substantially lower computational cost, 2) enhanced accuracy and stability when compared to C-IGA, and 3) robust numerical solution that satisfies the local and global mass balance.

3.5 Conclusions

In this chapter, the IGA framework with the classical Galerkin (G-IGA) and collocation (C-IGA) formulations is presented, and the novel control volume IGA formulation (CV-IGA) is introduced. Groundwater flow modeling in saturated heterogeneous porous media is considered as a preliminary research before addressing the more complex problem of groundwater flow in karst aquifers, which is the aim of the next chapter.

The IGA is presented as a unified multiscale framework in the sense that it can describe the geometry, material heterogeneity and solution, while obtaining a velocity field without the usual artificial discontinuities. The three formulations presented in this work have similarities but also important differences:

- G-IGA has an optimal convergence rate for head $(n + 1)$ and velocity (n) if n is order of the basis functions. However, G-IGA is the most expensive IGA concept due to numerical integration over $\Omega \in R^{dim}$, as well as the large number of nonzero matrix members for each equation $(2n + 3)^{dim}$, where dim represents the dimensionality of the problem.

- C-IGA has a suboptimal convergence rate for head for even (n) and odd $(n - 1)$ basis functions. Moreover, C-IGA has a suboptimal convergence rate for velocity for odd basis functions $(n - 1)$ but an optimal convergence rate for even (n) basis functions. C-IGA is considerably

less expensive than G-IGA due to the absence of numerical integration, as well as a substantially smaller number of nonzero matrix members ($(n+2)^{dim}$ for odd and $(n+1)^{dim}$ for even order basis functions). The main drawbacks of C-IGA are the absence of local and global mass balance and reduced stability in the case of high heterogeneity since a simple diffusion problem becomes an advection-dominated advection-diffusion problem in the strong collocation formulation.

- CV-IGA has a suboptimal convergence rate for head (n) for even basis functions. However, CV-IGA has an optimal convergence rate for head ($n+1$) for odd basis functions, as well as an optimal convergence rate for velocity (n) for all orders of basis functions, with an accuracy very close to that of G-IGA. CV-IGA lies between C-IGA and G-IGA with respect to computational cost, convergence and accuracy. The number of nonzero matrix members for each equation is $(n+2)^{dim}$ for odd and $(n+3)^{dim}$ for even basis functions. Additionally, CV-IGA requires numerical integration over $\Gamma_i = \partial\Omega_i \in R^{dim-1}$ control volume boundaries, which is much less expensive than that required by G-IGA. CV-IGA exclusively has a local/global mass balance property, in contrast to the two other IGA concepts.

The mentioned numerical properties for all formulations are summarized in Tables 3.1 and 3.2.

TABLE 3.1: Convergence properties of different formulations.

Formulation	**Even Fup$_n$**	**Odd Fup$_n$**
G-IGA	$n+1$	$n+1$
C-IGA	n	$n-1$
CV-IGA	n	$n+1$

TABLE 3.2: Number of nonzero elements in each row of the global stiffness matrix.

Formulation	**Even Fup$_n$**	**Odd Fup$_n$**
G-IGA	$(2n+3)^{dim}$	$(2n+3)^{dim}$
C-IGA	$(n+1)^{dim}$	$(n+2)^{dim}$
CV-IGA	$(n+3)^{dim}$	$(n+2)^{dim}$

Fup basis functions and B-splines of the same order have the same convergence rates for all three concepts and all mentioned cases. Fup basis functions

always obtain better accuracy than B-splines of the same order. However, Fup basis functions are more expensive than B-splines because they include one more characteristic interval. The choice of basis function depends on the problem, its numerical formulation and the required continuity.

Finally, the integral form of the equations and the direct physical meanings of the discretized equations, together with the noted computational advantages, make CV-IGA the most powerful candidate for groundwater flow modeling in karst aquifers, which is the main objective of the following chapter.

Chapter 4

Groundwater flow modeling in karst aquifers

In this chapter, a novel numerical and physical model for groundwater flow in karst aquifers is presented. After a general overview of the modeling approaches in karst hydrogeology, a mathematical model, which accounts for variably saturated conditions in both flow domains (matrix and conduits), is defined. The control volume formulation and Fup_1 basis functions are used as the foundation for the development of the numerical model. Discretization of the governing equations and a coupled 3D-1D numerical algorithm are discussed in detail. A 3D physical model is described, and a comparison of the numerical and experimental results is presented. The chapter ends with a discussion of the presented results and conclusions. The methodology and the scientific contributions of this chapter have been published in Malenica *et al.* [26].

4.1 Introduction

Approximately 20-25% of the global population depends largely or entirely on groundwater obtained from karst aquifers [27]. For proper water management and protection, it is important to understand and be able to predict groundwater flow in karst. Numerical modeling is an efficient method for describing many physical phenomena and is frequently used in traditional hydrogeology. However, the existence of a highly permeable conduit network embedded in a less permeable rock matrix results in a highly heterogeneous permeability distribution and makes karst different from other aquifers [28]. Figure 4.1 shows the conceptual model of a karst system according to Goldscheider and Drew [29]. Moreover, this duality in the permeability results in different flow conditions and makes karst particularly difficult to model. The conduit system occupies only a small portion of the total

aquifer but has a major impact on the hydraulic behavior of the karst system [30]. Normally, most of the recharge water is collected and transmitted toward the spring by the conduit network [27]. The karst conduits are characterized by fast, mostly turbulent flow, which can be free surface (partially saturated conduit) or pressurized (fully saturated conduit). The remaining part of the aquifer is characterized by a system of small fissures and pores and is herein referred as the karst matrix (generally, one can speak about a triple-porosity system: conduits, fractures and porous matrix; however, for flow modeling purpose, the latter two are usually combined and referred to as the matrix). The matrix is important for water storage and attenuation of contaminants. In contrast with conduits, the flow in the matrix is generally laminar (seepage) but can be both saturated and unsaturated. The complexity of the karst aquifers is further increased by the existence of many special karst forms (such as caves, sinkholes, and epikarst) and the fact that there is interaction between conduit and matrix systems. During drought season, the pressure in conduits is generally lower than the pressure in the matrix, and conduits are recharged by groundwater stored in the matrix. In contrast, during storm flow intervals, the pressure in the conduits increases faster than in surrounding matrix, and a substantial pressure difference between the matrix and conduit occurs. The flow in conduits normally becomes pressurized, and hydraulic head inversion occurs, i.e., the conduit network starts to recharge the matrix storage.

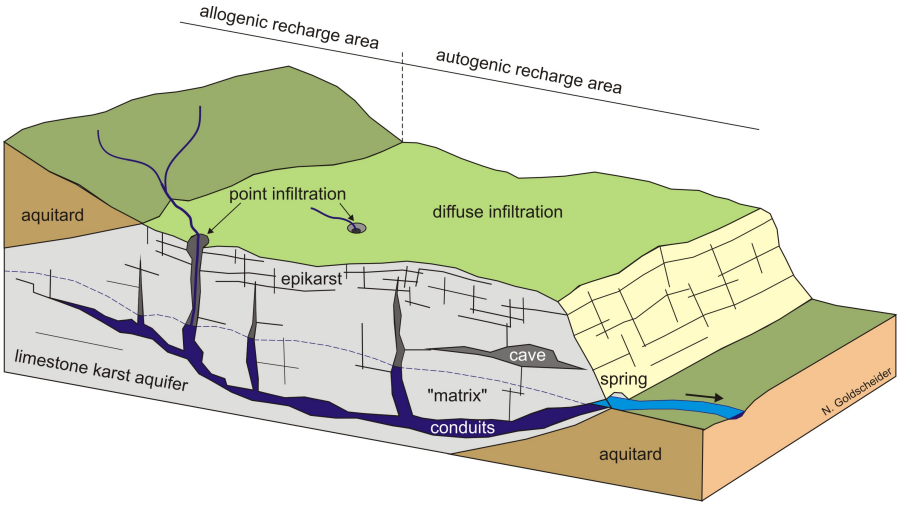


FIGURE 4.1: The conceptual model for a karst aquifer according to Goldscheider and Drew [29].

In recent years, many approaches for modeling groundwater flow in karst have been suggested (e.g.,[31]–[33]). Generally, they are divided into two main groups [30]: spatially lumped (hydrological) and spatially distributed

(hydraulic) models. In the following, only a short overview of the mentioned approaches is given; a more detailed descriptions can be find in [34], [35]. Spatially lumped models transform recharge events (input) into spring hydrographs (output). They are often called global models since they consider the system as a single unit and simulate the global hydraulic behavior of the entire system. These models are relatively simple to use and do not require detailed knowledge about the system; however, they do not simulate actual physical processes and cannot account for spatial variability within an aquifer. To account for spatial variations in the involved parameters (e.g., permeability or recharge distribution) and flow variables (e.g., pressure or velocity), spatially distributed (physical-based) models are needed. The physical processes of flow are described via partial differential equations (PDEs), and numerical (discretization) techniques are used to solve initial-boundary value problems (IBVPs). The solution of a particular IBVP is given by the flow variables, which have both spatial and temporal variability.

Among the different approaches suggested within the spatially distributed models (e.g., Kovács and Sauter [34] and Hartmann *et al.* [35]), the focus of this work is the so-called (coupled) discrete-continuum (DC) or hybrid approach. In this approach, karst conduits are discretized as one-dimensional elements, and governing equations similar to the one for hydraulic networks are used to describe the (usually pressurized) flow in the conduit network. The low-permeability matrix is discretized in three dimensions, and Darcy flow is generally assumed to be valid. The coupling between two domains is established via a first-order exchange term governed by the conduit-matrix head difference [36] or by assuming continuous heads [37]. The first DC model was presented by Király [38], and example applications of DC models include [36], [37], [39]–[47], among others. The most popular and widely used DC model is the Conduit Flow Process model [48] integrated into MODFLOW-2005 [49]. Conduit Flow Process (CFP) is an open-source finite difference solver that has been documented for wider use and developed to account for dual-porosity nature, such as that observed in karst aquifers. The flow in the karst matrix is based on the classical MODFLOW approach, whereas the turbulent conduit flow is based on the Darcy–Weisbach equation. Partially filled conduits are simulated as fully saturated pipes by using corrected diameter . However, most of the MODFLOW-CFP applications were performed under fully saturated conditions. The main reason for considering only fully saturated conditions is the robustness of the numerical simulations. It is well known that high

nonlinearity of soil properties in the unsaturated zone can produce severe convergence problems [50]. Similarly, the transition between a free surface and pressurized flow can affect the numerical behavior in terms of robustness and correctness [44]. Thus, it can be stated that MODFLOW-CFP has been designed and used as a compromise between practical applicability and numerical stability. Further, the two advanced karst flow models were presented by Rooij [51] and Rooij *et al.* [52]. First, in his thesis [51], the turbulent conduit flow was coupled with laminar matrix flow via a finite element model that accounted for variable saturation in both flow domains. The coupling was imposed by head continuity on the matrix-conduit interface. Later, a novel model in which surface flow was added to the previously mentioned subsurface system was developed [52]. The subsurface equations remained the same, but the conduit-matrix coupling was based on Peaceman's well index. Finite-difference discretization was used, and certain numerical problems mentioned in [51] were avoided. Both models were applied to hypothetical karst systems.

The DC approach can be generally considered as the most advanced among practically used models. While still being computationally acceptable, it accounts for a heterogeneity and duality of karst and can be used to better understand complicated hydrodynamic processes as well as for testing different conceptual models. However, in addition to the complexity of developing reliable numerical models, this approach requires large input data, which usually limits its practical application [34]. The shape and position of the conduit network and a reasonable distribution of hydraulic parameters and recharge are necessary to obtain realistic results. Moreover, the verification of such complex numerical models is quite difficult, but still, it is important before its practical usage. Since analytical solutions are not available (except for some trivial cases) and collected field data are generally insufficient for true verification, laboratory experiments are one possible logical choice. In the literature, there seems to be only a few attempts to construct laboratory analogues of karst. Models such as those presented by Öllős and Németh [53] and Wu and Hunkeler [54] were constructed to analyze flows dominated by conduit network. Faulkner *et al.* [55] has demonstrated a laboratory model that is capable of simulating coupled conduit-matrix conditions. It has been used to analyze both groundwater flow and solute transport, and their numerical model has been verified with measured results. However, the overall dimensions (0.6 x 0.26 x 0.02 m) of their model make it essentially two-dimensional and applicable only for laminar flow in conduits. Recently,

the Castro [56] presented the 3D sandbox with overall dimensions 1.044 m x 0.419 m x 0.434 m. The sandbox was filled with homogeneous porous material and perforated steel pipe with external diameter 0.019 m was used to simulate karst conduit. The steady-state experiments under saturated flow conditions were performed, and collected data was used for the validation of MODFLOW-CFP.

The purpose of this chapter is twofold. First aim is to develop novel computational model based on Fup basis functions and control volume formulation. The model is of the discrete-continuum type and accounts for variably saturated conditions in both flow domains. Since groundwater flow modeling in karst is quite complex, the advancement in a pure mathematical and numerical sense is important for future improvement of modeling capabilities in karst hydrogeology. Moreover, in such a complex area, critical consideration of the used methods and obtained results is necessary. Furthermore, as a second objective, a three-dimensional physical model was constructed. Although it is not an exact representation of a realistic karst system, the presented unique and quite complex physical karst flow model can be used for verification and validation of different mathematical and numerical models. The most important features of karst systems can be captured, such as laminar groundwater flow through both unsaturated and saturated heterogeneous porous media, turbulent free surface and/or pressurized flow through karst conduits and groundwater exchange between two systems. Different boundary conditions and recharge characteristics, in addition to many other possibilities, enable the testing of different flow conditions. The pressure distribution in a karst matrix and discharge from both systems are continuously measured during experiments. Details regarding to the novel physical model and additional experimental results are shown in Appendix A.

4.2 Mathematical Model

Generally, the same physical laws (i.e., conservation of mass and momentum) govern both matrix and conduit flows. However, for practical purposes, they often have to be simplified.

Thus, the matrix system is modeled as a continuum using the Darcy law and the representative elementary volume (REV) concept [57]. Additionally, in many practical applications, the variably saturated flow equation [58], [59] (often referred to as Richards' equation) is capable to accurately simulate flow through both saturated and unsaturated zone.

Flow in the conduits is generally described with the system of 3D Navier-Stokes equations. However, solution of the full 3D equations is very demanding, and the possible free surface flow condition necessitates special approaches. Additionally, the flow in conduits is usually turbulent, and the scale of interest is large [60], [61]. All these reasons, together with the fact that the geometry of the karst conduits is very complex and, in most cases, unknown, lead to simplified mathematical models. Thus, as in usual approaches for hydraulic network and open-channel flow modeling [62], [63], the flow in karst conduits is treated as one-dimensional.

Since flow equations of the two systems (i.e., matrix and conduit) are governed by different partial differential equations (and dimensionality), mathematical models have to be coupled. This section describes mathematical models for matrix flow, conduit flow and conduit-matrix interface.

In the following, uppercase (H, Ψ) and lowercase (h, ψ) will be used to distinct between matrix and conduit hydraulic and pressure heads, respectively.

4.2.1 Variably saturated flow in karst matrix

The mathematical model for flow in the karst matrix is based on the governing equation for variably saturated subsurface flow in a porous medium (eq., [48], [58], [64]):

$$S_s \frac{\theta}{\eta} \frac{\partial H}{\partial t} + \frac{\partial \theta}{\partial t} = \nabla \cdot \Big(k_r(\theta) \mathbf{K_s} \cdot \nabla H \Big) + q_{Ms} \tag{4.1}$$

where S_s is the specific storage coefficient (L^{-1}), θ is the volumetric water content (-), η is the porosity (-), H is the hydraulic (piezometric) head (L), k_r is the relative permeability (-), $\mathbf{K_s}$ is the saturated hydraulic conductivity tensor ($L\ T^{-1}$), and q_{Ms} is the volumetric flux per unit volume representing sources and (or) sinks (T^{-1}). Let $[0, t_{end}]$ be the time interval of interest, Ω be a bounded domain, and Γ_D and Γ_N be domain boundaries with Dirichlet and Neumann boundary conditions, respectively. The appropriate initial and boundary conditions are given by

$$H = H_0 \quad \text{on} \quad \Omega, \quad t = 0 \tag{4.2}$$

$$H = H_D \quad \text{on} \quad \Gamma_D, \quad t \in [0, t_{end}] \tag{4.3}$$

$$-(k_r(\theta) \mathbf{K_s} \cdot \nabla H) \cdot \mathbf{n} = q_N \quad \text{on} \quad \Gamma_N, \quad t \in [0, t_{end}] \tag{4.4}$$

where **n** (-) is the outward normal vector and H_0, H_D and q_N are prescribed initial and boundary condition functions. To solve system (4.1)-(4.4), constitutive relationships for describing the unsaturated flow and storage properties must be defined. In this work, the bimodal constrained form of the van Genuchten-Mualem model ([65], [66]) is used.

The water content is expressed through the effective saturation Θ_e (-):

$$\Theta_e(\Psi) = \frac{\theta - \theta_r}{\theta_s - \theta_r} \tag{4.5}$$

where Ψ is the pressure head ($\Psi = H - z$) (L), z is the elevation head (L), θ_r is the residual volumetric water content, and θ_s is the saturated volumetric water content (often assumed to be equal to the porosity). Generally, the saturation and volumetric water content are related by the porosity, i.e., ($\theta = \Theta \cdot \eta$).

The relationship between the water content and pressure head (retention function) is given by

$$\Theta_e(\Psi) = \begin{cases} \sum_{j=1}^{2} w_j \Theta_{e,j} & \Psi < 0 \\ 1 & \Psi \geq 0 \end{cases} \tag{4.6}$$

where each mode of the curve is given by [67]

$$\Theta_{e,j}(\Psi) = [1 + (|\alpha_j \Psi|)^{n_j}]^{-m_j} \tag{4.7}$$

The specific moisture capacity is given as a function of the effective saturation:

$$C(\Theta_e) = \frac{\partial \theta}{\partial \Psi} = -\sum_{j=1}^{2} w_j \frac{\eta m_j \alpha_j (1 - \Theta_r)}{1 - m_j} \Theta_r^{1/m_j} (1 - \Theta_{e,j}^{1/m_j})^{m_j} \tag{4.8}$$

The relative permeability is given as a function of the effective saturation:

$$k_r(\Theta_e) = \left[\sum_{j=1}^{2} w_j \Theta_{e,j} \right]^{\tau} \left[\frac{\sum_{j=1}^{2} w_j \alpha_j \left[1 - (1 - \Theta_{e,j}^{1/m_j})^{m_j} \right]}{\sum_{j=1}^{2} w_j \alpha_j} \right]^2 \tag{4.9}$$

where α_j is the inverse of the air entry pressure head (L^{-1}), n_j is the pore size distribution index ($n_j > 1$) (-), m_j is the parameter defined as

$m_j = 1 - 1/n_j$ (-), and τ is an additional fitting parameter (-), called the tortuosity parameter, which is often set to $\tau = 0.5$. This model is a weighted superposition of two van Genuchten functions, where the weighting factors w_j are subject to $0 \leq w_j \leq 1$ and $\sum w_j = 1$. This function is much more suitable to fit data that do not perfectly follow the van Genuchten unimodal shape and will be used in this work to describe soil properties of physical model. Note that the bimodal model also covers the commonly used unimodal van Genuchten model by simply setting $w_1 = 1$ and $w_2 = 0$.

4.2.2 Variably saturated flow in karst conduits

The one-dimensional flow in karst conduits in the l-direction is given by

$$C(h)\frac{\partial h}{\partial t} = \frac{\partial}{\partial l}\left(\frac{K_C(h)}{\sqrt{|\frac{\partial h}{\partial l}|}}\frac{\partial h}{\partial l}\right) + q_{Cs} \tag{4.10}$$

where C is the capacity term (L), h is the hydraulic (piezometric) head (L), l is the spatial longitudinal coordinate (L), K_C is the conveyance factor ($L^3\ T^{-1}$), and q_{Cs} is a sink/source term ($L^2\ T^{-1}$). Appropriate initial and boundary conditions are given by

$$h = h_0 \quad \text{on} \quad \Omega, \quad t = 0 \tag{4.11}$$

$$h = h_D \quad \text{on} \quad \Gamma_D, \quad t \in [0, t_{end}] \tag{4.12}$$

$$-\frac{K_C(h)}{\sqrt{|\frac{\partial h}{\partial l}|}}\frac{\partial h}{\partial l} = Q_N \quad \text{on} \quad \Gamma_N, \quad t \in [0, t_{end}] \tag{4.13}$$

where h_0, h_D and Q_N are prescribed initial and boundary condition functions. The nonlinear diffusion equation (4.10) is usually called the noninertia wave equation and can be derived by neglecting inertia terms in the general momentum equation and combining with the mass conservation equation [51]. The noninertia wave equation can be used to describe both free surface and pressurized flows by modifying the capacity term (Rooij *et al.* [52]):

$$C = \begin{cases} W & \text{for free surface flow} \\ \rho_w g A_c \kappa_w & \text{for pressurized flow} \end{cases} \tag{4.14}$$

where W is the top width of the flow (L), ρ_w is the density of the water ($M\ L^{-3}$), g is the acceleration of the gravity ($L\ T^{-2}$), A_c is the cross-section flow

area (L^2), and κ_w is the compressibility of water ($L\ T^2\ M^{-1}$). The conveyance factor K_C is given by the Manning equation [68]:

$$K_C = \frac{1}{n_M} R^{2/3} \cdot A_c \tag{4.15}$$

where n_M is Manning's roughness coefficient ($L^{-1/3}\ T$) and R is the hydraulic radius (L), defined as $R = A_c / p_w$, where p_w is the wetted perimeter (L). The variables W, A_c, and p_w in the general case depend on the geometry of the conduit cross-section and water depth ψ (L). For a circular conduit with radius r_c (L), they are given by

$$W = 2\psi \sqrt{\frac{2r_c}{\psi} - 1} \tag{4.16}$$

$$A_c = \begin{cases} r_c^2 \arccos\left(1 - \frac{\psi}{r_c}\right) + \psi(\psi - r_c)\sqrt{\frac{2r_c}{\psi} - 1} & \text{for free surface flow} \\ \pi r_c^2 & \text{for pressurized flow} \end{cases} \tag{4.17}$$

$$p_w = \begin{cases} 2r_c \arccos\left(1 - \frac{\psi}{r_c}\right) & \text{for free surface flow} \\ 2\pi r_c & \text{for pressurized flow} \end{cases} \tag{4.18}$$

4.2.3 Exchange of water between the matrix and conduit flow

The exchange of water is established by defining a linear flux relation on the interface between different flow systems (e.g., Liedl *et al.* [36]):

$$q_{ex} = \alpha_{ex}(H - h) \tag{4.19}$$

where H and h are, as defined before, hydraulic heads in the matrix and conduit, and α_{ex} is the exchange coefficient (T^{-1}), which is usually determined via calibration. From a mathematical point of view, q_{ex} is expressed as a scalar value while being aware that its direction coincides with the normal direction of the conduit-matrix interface. Expression (4.19) will be used to couple the matrix and conduit flow systems. The incorporation of this term into both the matrix and conduit flow equations will be explained in more detail in next section.

4.3 Numerical Model

This section describes the numerical model used for discretization of the presented mathematical model. A novel numerical approach based on first order Fup_1 basis functions and control-volume formulation (presented in section 3.2.3) is used for spatial discretization. The temporal discretization is based on the well-known implicit (backward) Euler method, whereas the Picard technique is used for linearization of the governing equations. The coupling of the matrix and the conduit flow is based on the iterative (segregate) coupling approach. In this approach, each system has its own solver, and an iterative procedure is performed until convergence of both systems is not achieved. The solution from the latest iteration of one system is used to calculate the exchange fluxes in the other systems. Additionally, when needed, superscripts *M* and *C* will be used to distinguish matrix and conduit variables.

4.3.1 Discretization of the governing equation for the matrix flow

The governing equation for variably saturated flow in the matrix (4.1) is first discretized in time and linearized by following the approach suggested by Celia *et al.* [50]. Backward Euler temporal discretization of (4.1) produces

$$\frac{S_s}{\eta}\theta^{t+\Delta t}\frac{H^{t+\Delta t}-H^t}{\Delta t}+\frac{\theta^{t+\Delta t}-\theta^t}{\Delta t}=\nabla\cdot\left(k_r(\theta^{t+\Delta t})\mathbf{K_s}\cdot\nabla H^{t+\Delta t}\right)+q_{Ms}^{t+\Delta t} \tag{4.20}$$

Picard linearization can be applied to (4.20) to obtain

$$\frac{S_s}{\eta}\theta^{t+\Delta t,m}\frac{H^{t+\Delta t,m+1}-H^t}{\Delta t}+\frac{\theta^{t+\Delta t,m+1}-\theta^t}{\Delta t} =\nabla\cdot\left(k_r(\theta^{t+\Delta t,m})\mathbf{K_s}\cdot\nabla H^{t+\Delta t,m+1}\right)+q_{Ms}^{t+\Delta t,m} \tag{4.21}$$

where $m+1$ and m denote the current and previous iteration levels. As suggested by Celia *et al.* [50], the term $\theta^{t+\Delta t,m+1}$ can be expanded in a truncated Taylor series with respect to the hydraulic head H as

$$\theta^{t+\Delta t,m+1}=\theta^{t+\Delta t,m}+\left.\frac{d\theta}{dH}\right|^{t+\Delta t,m}\left(H^{t+\Delta t,m+1}-H^{t+\Delta t,m}\right)+O(\partial^2) \tag{4.22}$$

Neglecting all terms that of are an order higher than linear and incorporating (4.22) into (4.21) gives

$$\frac{S_s}{\eta}\theta^* \frac{H^{t+\Delta t} - H^t}{\Delta t} + \frac{\theta^* + C^*(H^{t+\Delta t} - H^*) - \theta^t}{\Delta t} = \nabla \cdot \left(k_r(\theta^*) \mathbf{K_s} \cdot \nabla H^{t+\Delta t} \right) + q^*_{Ms} \quad (4.23)$$

where $C = \frac{d\theta}{dH}$, whereas symbol $(*)$ is introduced to denote $(t + \Delta t, m)$, i.e., the previous iteration, and $(t + \Delta t)$ now denotes the current iteration, i.e., $(t + \Delta t, m + 1)$. This approach, which is called the modified Picard approximation and was introduced by Celia *et al.* [50], enables the discretize mixed-form equation (4.1) in a manner such that instead of two unknown variables (the water content and hydraulic head), the only remaining unknown variable is the hydraulic head. Contrary to the usually used pressure based formulation [50], this form retains the mass conservative properties of the numerical solution. The simple rearrangement of variables is performed to group all unknowns on the left side and all known variables on the right side:

$$H^{t+\Delta t}\left(\frac{S_s}{\eta}\theta^* + C^*\right) - \Delta t \, \nabla \cdot \left(k_r(\theta^*) \mathbf{K_s} \cdot \nabla H^{t+\Delta t} \right) = \frac{S_s}{\eta}\theta^* H^t + C^* H^* + \theta^t - \theta^* + \Delta t \, q^*_{Ms} \quad (4.24)$$

Next, spatial discretization is performed by first integrating semidiscrete differential equation (4.24) over a matrix finite control volume Ω_i^M:

$$\int_{\Omega_i^M} H^{t+\Delta t}\left(\frac{S_s}{\eta}\theta^* + C^*\right) d\Omega - \Delta t \int_{\Gamma_i^M} \left(k_r(\theta^*) \mathbf{K_s} \cdot \nabla H^{t+\Delta t} \right) \cdot \mathbf{n} \, d\Gamma = \int_{\Omega_i^M} \left(\frac{S_s}{\eta}\theta^* H^t + C^* H^* + \theta^t - \theta^* + \Delta t \, q^*_{Ms} \right) d\Omega \quad (4.25)$$

where the divergence theorem is applied on the second term of the left-hand side, as described in the section 3.2.3. Following (3.3), the hydraulic head can be expressed as

$$H(\mathbf{x}, t) = \alpha_j^M(t) \varphi_j^M(\mathbf{x}) \quad (4.26)$$

Since H is function of both space and time, the unknown Fup coefficients α_j are now time-dependent. Introducing (4.26) into (4.25) produces the full

discretized counterpart of equation (4.1) over Ω_i^M, written in the form

$$\alpha_j^{M,t+\Delta t} \cdot \left[S_{ij}^{M,*} + C_{ij}^{M,*} - K_{ij}^{M,*}\right] = \alpha_j^{M,t} \cdot S_{ij}^{M,*} + \alpha_j^* \cdot C_{ij}^{M,*} + V_i^{M,t} - V_i^{M,*} + W_i^{M,*} \quad (4.27)$$

where

$$S_{ij}^{M,*} = \frac{S_s}{\eta} \int_{\Omega_i^M} \theta^* \varphi_j d\Omega \quad (4.28)$$

$$C_{ij}^{M,*} = \int_{\Omega_i^M} C^* \varphi_j d\Omega \quad (4.29)$$

$$K_{ij}^{M,*} = \Delta t \int_{\Gamma_i^M} \left(k_r(\theta^*)\mathbf{K_s} \cdot \nabla \varphi_j\right) \cdot \mathbf{n} d\Gamma \quad (4.30)$$

$$V_i^{M,t} = \int_{\Omega_i^M} \theta^t d\Omega \quad (4.31)$$

$$V_i^{M,*} = \int_{\Omega_i^M} \theta^* d\Omega \quad (4.32)$$

$$W_i^{M,*} = \Delta t \int_{\Omega_i^M} q_{Ms}^* d\Omega \quad (4.33)$$

In the general case, coefficients (4.28)-(4.33) must be calculated via numerical integration. The analytically described function of initial conditions $H_0 = H_0(\mathbf{x})$ has to be projected to the space spanned by basis functions. In the standard numerical methods (e.g., FEM or FDM), this is done by setting nodal values $H_i^0 = H_0(\mathbf{x}_i)$. In the case of non-interpolatory basis functions, this is not possible, since degrees of freedom are not associated with nodal values of the solution [69]. Therefore, a control-volume projection of the form

$$\alpha_i^0 \int_{\Omega_i} \varphi_j d\Omega = \int_{\Omega_i} H_0 d\Omega \quad (4.34)$$

is defined for each CV to produce a system of equations whose solution (coefficients α_i^0) represents numerically specified initial conditions. Writing equations for all CVs and incorporating initial and boundary conditions (explained in section 3.2.3) produces a system of equations whose solution is the set of Fup coefficients $\alpha_i^{t+\Delta t}$, which define the matrix hydraulic head solution at time $t + \Delta t$ by (4.26). After the hydraulic head H is known, the water content θ can be obtained by using (4.5)-(4.7) Additionally, the numerical solution for Darcy velocity field $\mathbf{q}$ (L/T) is smooth and defined at any point of domain Ω by

$$\mathbf{q}(\mathbf{x}, t) = -k_r(\theta)\mathbf{K_s} \cdot \left(\alpha_j^M(t) \nabla \varphi_j^M(\mathbf{x})\right) \quad (4.35)$$

4.3.2 Discretization of the governing equation for conduit flow

Following the approach and notation used for matrix discretization, the equation for the conduit flow (4.10) is first discretized in time by the implicit Euler method and linearized via the Picard technique:

$$C^* \frac{h^{t+\Delta t} - h^t}{\Delta t} = \frac{\partial}{\partial l}\left(\frac{K_C^*}{\sqrt{|\frac{\partial h^*}{\partial l}|}} \frac{\partial h^{t+\Delta t}}{\partial l} \right) + q_{Cs}^* \tag{4.36}$$

Regrouping variables and integration over the one-dimensional conduit CV Ω_i^C produces

$$\int_{\Omega_i^C} h^{t+\Delta t} C^* d\Omega - \Delta t \int_{\Omega_i^C} \frac{\partial}{\partial l}\left(\frac{K_C^*}{\sqrt{|\frac{\partial h^*}{\partial l}|}} \frac{\partial h^{t+\Delta t}}{\partial l} \right) d\Omega = \int_{\Omega_i^C} \left(h^t C^* + \Delta t \, q_{Cs}^* \right) d\Omega \tag{4.37}$$

Introducing

$$h(x,t) = \alpha_j^C(t) \varphi_j^C(x) \tag{4.38}$$

into (4.37) produces

$$\alpha_j^{C,t+\Delta t} \cdot \left[C_{ij}^{C,*} - K_{ij}^{C,*} \right] = \alpha_j^{C,t} \cdot C_{ij}^{C,*} + W_i^{C,*} \tag{4.39}$$

where

$$C_{ij}^{C,*} = \int_{\Omega_i^C} C^* \varphi_j d\Omega \tag{4.40}$$

$$K_{ij}^{C,*} = \Delta t \int_{\Omega_i^C} \frac{\partial}{\partial l}\left(\frac{K_C^*}{\sqrt{|\frac{\partial h^*}{\partial l}|}} \frac{\partial \varphi_j}{\partial l} \right) d\Omega \tag{4.41}$$

$$= \Delta t \left(\frac{K_C^*}{\sqrt{|\frac{\partial h^*}{\partial l}|}} \frac{\partial \varphi_j}{\partial l} \right)\Bigg|_{\Gamma_{ie}^C} - \Delta t \left(\frac{K_C^*}{\sqrt{|\frac{\partial h^*}{\partial l}|}} \frac{\partial \varphi_j}{\partial l} \right)\Bigg|_{\Gamma_{iw}^C} \tag{4.42}$$

$$W_i^{C,*} = \Delta t \int_{\Omega_i^C} q_{Cs}^* d\Omega \tag{4.43}$$

where Γ_{ie}^C and Γ_{iw}^C denote the east and west boundaries of the i-th one-dimensional conduit control volume.

4.3.3 Numerical procedure for matrix-conduit coupling

The exchange of water between the matrix and conduit occurs at the interface between two flow domains. Since the control-volume formulation of the governing PDEs has a direct physical interpretation in terms of the underlying (mass) conservation laws, the exchange term can be conveniently expressed in terms of volumetric discharge as

$$Q_{ex} = \int_{\Gamma_{ex}} \alpha_{ex}(H - h) d\Gamma \tag{4.44}$$

where Γ_{ex} is the interface surface area (L^2). Coupling is established by incorporating term (4.44) in both equations through the source term. Since the calculated discharge leaves one domain and enters other, different signs are needed in different systems. The exchange term is fully explicit treated, i.e., the latest calculated matrix and conduit solutions are used to calculate the exchange term in both systems. Thus, the matrix source term (4.33) is modified to account for the conduit's contribution and now has the form

$$W_i^{M,*} = \Delta t \int_{\Omega_i^M} q_{Ms}^* d\Omega - \Delta t \int_{\Gamma_{iex}^M} \alpha_{ex}(H^* - h^*) d\Gamma \tag{4.45}$$

whereas the conduit source term (4.43) is modified to account for the matrix's contribution according to

$$W_i^{C,*} = \Delta t \int_{\Omega_i^C} q_{Cs}^* d\Omega + \Delta t \int_{\Gamma_{iex}^C} \alpha_{ex}(H^* - h^*) d\Gamma \tag{4.46}$$

Although the mathematical model for the conduit flow is one-dimensional, the exchange term is still calculated by considering a full three-dimensional conduit geometry. Additionally, due to the properties of the Fup basis functions, the matrix and conduit discretizations are not constrained by each other. Consequently, the CV boundaries of two flow domains and the vertices of different-dimensionality Fup basis functions do not need to match.

4.3.4 Overview of the numerical procedure

As already mentioned, the coupling is based on an iterative (segregate) approach. The overall solution procedure for the coupled model is summarized in Algorithm 1. Since both flow equations are nonlinear, there is a possibility of numerical instability. The usual techniques for increasing the robustness of the algorithm for equations (4.1) and (4.10) are mass lumping, upstream

Algorithm 1 The numerical algorithm

```
Initial guess = initial condition
H* = H^0
h* = h^0
Time step iterations
for TS = 1, N_TS do
   Matrix-conduit coupling iterations
   for IC = 1, N_IC do
      Matrix solver iterations
      for M = 1, N_M do
         Solve for H^{t+Δt}
         H* = H^{t+Δt}
      end for
      Conduit solver iterations
      for C = 1, N_C do
         Solve for h^{t+Δt}
         h* = h^{t+Δt}
      end for
   end for
   t = t + Δt
end for
```

weighting, positivity preserving [51] and underrelaxations [1]. In this work, all of these techniques are implemented and tested in some form; however, for the numerical examples used in this work, only explicit underrelaxation for solution of the nonlinear system of equations is used. Additionally, a simple adaptive time-stepping procedure based on counting the matrix-conduit coupling iterations is implemented. If the number of coupling iterations exceeds some user-predefined limit, the calculation is restarted at the beginning of the current time step, and a smaller time step is used.

4.4 Verification Examples

4.4.1 Matrix flow

Two-dimensional variably saturated infiltration data, presented by Vauclin *et al.* [70] and modeled by others (e.g., [48], [58]), are used to verify the matrix solver. The flow domain is a 6.00 m $\times$ 2.00 m box filled with homogeneous sandy soil. The initial conditions are a horizontal water table set at a height of 0.65 m from the bottom, and the water level on both the left and the right

faces is maintained at 0.65 m. A constant flux of q = 3.55 m/day was applied across the central 1.0 m of the soil surface, whereas the remaining surface was covered to prevent evaporation losses. Because of the experiment's symmetry, only half (the right part) of the domain (3.00 m x 2.00 m) is modeled. Boundary conditions are the specified flux on the left part (0.5 m) of the soil surface and the specified water level on the right face up to z=0.65 m. Other boundaries (including the left symmetry plane and right plane above the water table) are defined as no-flow boundary conditions. The soil properties used in the model are a saturated hydraulic conductivity of $K_S = 8.40$ m/day, a porosity of $\eta = \theta_s = 0.3$, and a residual saturation of $\theta_r = 0.01$. The unsaturated soil properties are fitted with a unimodal constrained Van Genucthen model with $\alpha = 3.3$ m^{-1} and $n = 4.1$. The specific storage S_s is neglected and set to zero.

The three-dimensional matrix solver is used to model this problem, whereas the results are shown as two-dimensional water table positions for selected times (Figure 4.2). Relatively coarse discretization is used ($\Delta x = \Delta z = 0.1$ m and $\Delta t = 10.0$ min); still, the model satisfactorily captures the water-table dynamics.

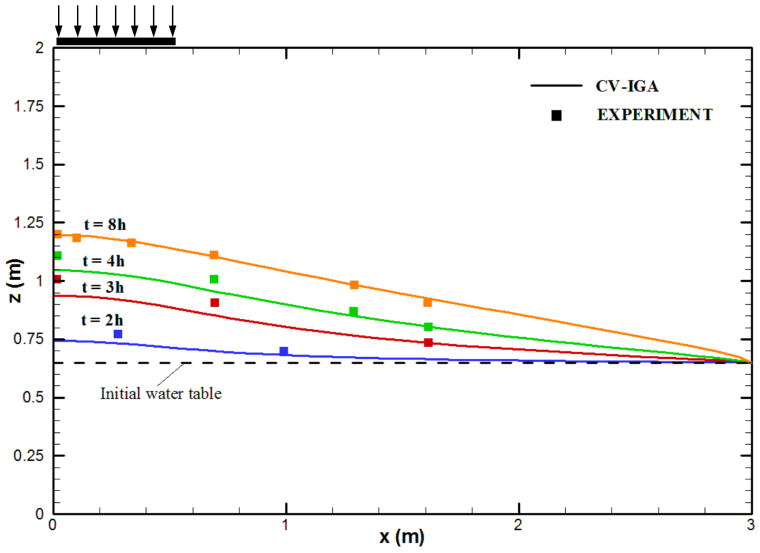


FIGURE 4.2: Water-table positions for infiltration study of [70].

4.4.2 Conduit flow

To verify the conduit solver, the second challenge test case performed by Rossman [71], which includes the transition between the free surface and pressurized flow is used. The problem consists of five conduit sections in

series (Figure 4.3), each being 1000 ft long. The first, third and fifth sections have diameters of 12 ft, whereas the second and fourth have diameters of 3 ft. The slope of the conduits is 0.05, and Manning's roughness coefficient is given by $n_M = 0.02$ m$^{1/3}$ s. Initially, the conduit is dry, and during the first 15 min, the inflow discharge is linearly increased from 0 to 50 ft^3/s. After 3 hours of simulation, the discharge is linearly decreased to zero in 15 min. The total simulation is 6 hours.

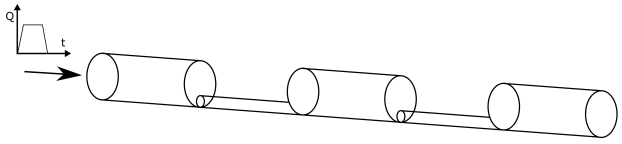


FIGURE 4.3: The Rossman [71] test example.

Each conduit section is discretized by 50 (one-dimensional) control volumes, and a time step of $\Delta t = 15$ s is used. The results are compared with those of the Storm Water Management Model (SWMM) presented by Rossman [71] and the DisCon model presented by Rooij *et al.* [52]. Figure 4.4 shows the flow rate in the middle of the first conduit section. The transition from a free surface to a pressurized flow is realized after approximately 1.5 hours. The results are in excellent agreement with the de Rooij model which is based on the same noninertia wave model (4.10). SWMM uses the full dynamic wave equation; still, good agreement of the results is observed.

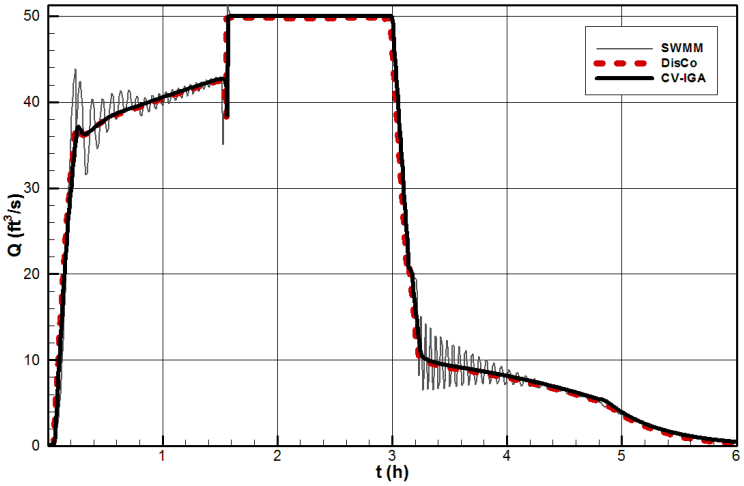


FIGURE 4.4: Hydrograph: comparison between SWMM [71], DisCo [52] and CV-IGA (present model).

4.5 Verification of the CV-IGA karst flow model with experimental data

In this section, the experimental setup is introduced, and the experimental results are compared with the numerical results. The details of the physical model and additional experimental results can be found in the Appendix A.

4.5.1 Experimental setup

FIGURE 4.5: Photography of physical model.

The physical model (Figure 4.5) is made of a concrete construction with overall dimensions 5.66 m $\times$ 2.95 m $\times$ 2.00 m. Two water reservoirs, as shown in Figure 4.6, are used to specify the water levels (boundary conditions) on the upstream and downstream ends of the model. Between reservoirs, the model is filled by heterogeneous porous material, which is used to simulate the karst matrix. Three type of materials are used: coarse quartz sand (CQS—67%), fine quartz sand (FQS—14%) and gravel (G—19%). The top 25 cm of the soil is filled by gravel to simulate epikarst effects. This layer prevents surface runoff and enables investigation of additional effects, such as fast infiltration and ponding on the interface with lower-permeability materials. In the middle of the model, there is a zone of fine quartz sand, which is used to produce a pressure drop and to decrease the overall matrix velocities. The rest is mostly filled by coarse quartz sand, but there are also particular zones with two other materials to magnify heterogeneity effects (Figure

4.7). Plastic perforated pipes are installed through the porous medium and are used to simulate karst conduits, whereas vertical branches are directly connected with conduits to simulate sinkholes. There are three independent conduit systems; however, during the experiment, only one conduit system is activated. The perforated parts of the pipes are surrounded by fine quartz sand and covered with geotextile to prevent perforations from clogging. On the top of the concrete construction, there are sprinklers and shower heads for rainfall simulation. The flow (computational) domain of matrix excludes walls and reservoirs, and the dimensions are $L_x = 4.0$ m, $L_y = 2.55$ m, and $L_z = 2.0$ m.

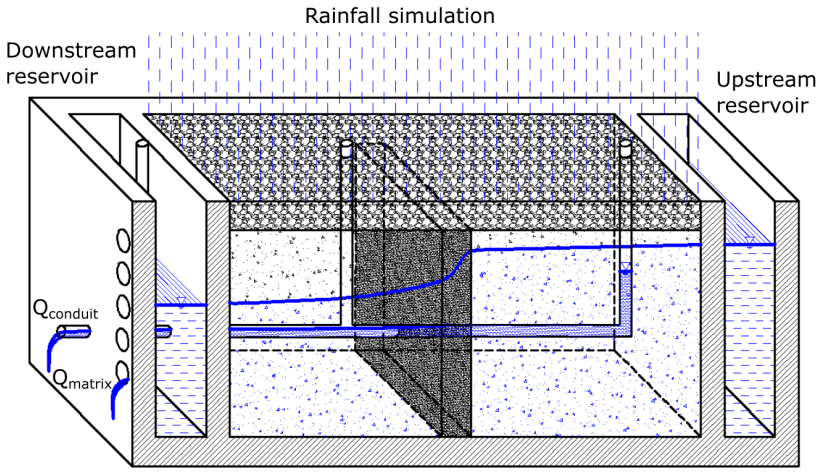


FIGURE 4.6: Scheme of the physical model

During the experiment, there is continuous measurement of discharges and pressures. The discharge is measured on the downstream ends of the model for both the matrix and conduit systems. The matrix discharge is regarded as the overall discharge from the porous medium that overflows from the downstream reservoir. The conduit discharge is measured on the outlet as sketched in Figure 4.6. In the comparison with realistic karst aquifers, the conduit outlet could be regarded as a main spring, whereas the matrix discharge can be considered as the total water that flows beside the main spring, regarding the porous matrix, fractures and small conduits. Measurement of the matrix discharge enables information that is generally unknown in practice. In addition to discharge, the pressure distribution is measured at 44 fixed points by pipes that are installed through the foundation slab and rise vertically. These pipes are perforated only on their ends (inside the porous medium) and are denoted as piezometers. Additionally, 25 fully perforated vertical pipes are installed starting from soil surface; they are denoted as boreholes. Since these pipes are fully perforated, borehole packers are used

to isolate part of pipe (approximately 10 cm) at the specific depth. A pressure sensor or water hose can be installed at any depth of borehole and used for pressure measurements or performing partially penetrating pumping tests, respectively.

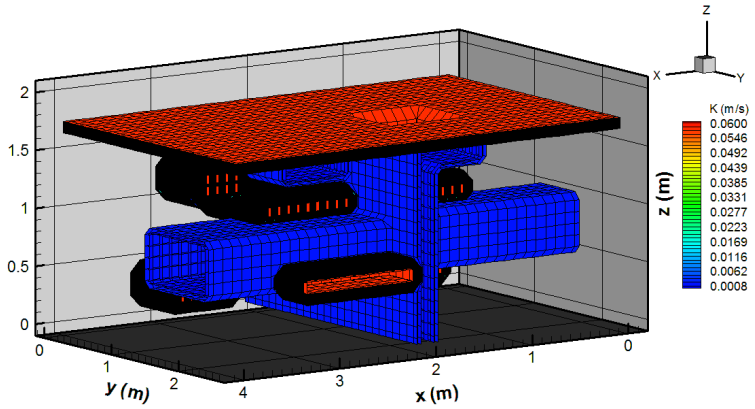


FIGURE 4.7: Heterogeneity of the porous medium.

The soil parameters for three different material types are presented in Table 4.1. The soil samples were sent to a specialized laboratory for measurements of the saturated and unsaturated soil parameters. However, because of the extremely high conductivity, the results for gravel (G) were not obtained. Therefore, Darcy experiments were performed for all three materials and, in addition to obtained values in laboratory (for CQS and FQS), used to determine the range of hydraulic conductivities for each material. Then, additional steady-state experiments (see Appendix A) were performed, and numerical calibration was used to define the final values of the saturated conductivities. The numerical calibration revealed that there exist moderate anisotropy of vertical conductivities, which is important to consider, especially because conduit-matrix interaction produces significant velocities in the vertical direction. All values adopted during the numerical calibration were inside the range of measured values. The laboratory measurements of unsaturated soil properties were available for CQS and FQS, and bimodal van Genuchten curves (presented in section 4.2.1) were used since classical (unimodal) van Genuchten curves were not able to fit data accurately. For G, there were no measurements, and the parameters were determined by guidelines from the literature and numerical calibration. The specific storage coefficient (S_s) values were taken from the literature.

As mentioned above, there are three different conduit systems installed in model, labeled C1, C2 and C3, that have differences in the shape, pipe diameter and number and size of perforations. If any of these conduits is

TABLE 4.1: Soil parameters for three different materials: coarse quartz sand (CQS), fine quartz sand (FQS) and gravel (G).

Parameter	CQS	FQS	G
K_H (m/s)	3.40×10^{-3}	2.00×10^{-4}	6.00×10^{-2}
K_V (m/s)	1.26×10^{-3}	8.00×10^{-5}	6.00×10^{-2}
S_s (m^{-1})	1.0×10^{-5}	1.0×10^{-4}	1.0×10^{-6}
α_1 (m^{-1})	18.00	3.00	10.00
α_2 (m^{-1})	0.50	2.00	-
n_1 (-)	3.2	9.2	2.9
n_2 (-)	1.3	2.2	-
w_1 (-)	0.62	0.67	1.0
w_2 (-)	0.38	0.33	0.0
τ (-)	2.96	2.295	0.5
η (-)	0.325	0.38	0.40
θ_r (-)	0.005	0.02	0.01

clogged at the outlet, its influence on the matrix flow is minor and therefore can be neglected; this was confirmed by the numerical model. Three types of PVC pipes were used, and the Manning coefficients are defined by manufacturer for conduits C1, C3 and unperforated part of C2 conduit. For the perforated part of C2 conduit, which is corrugated, there was not a specified value. Therefore, numerical calibration was needed; moreover, specific flow conditions (significant lateral inflow) inside the model prevented reliable empirical determination from being performed outside the model.

The physical karst flow model enables different experimental setups including different water levels in reservoirs, different rain distribution and intensity, pumping/extracting water from any of boreholes. Additionally, many different flow conditions can be achieved to test numerical models; for example, there are both saturated and unsaturated flow conditions in the matrix, flow in the conduits can be free surface or pressurized, and interaction between the matrix and conduit can be tested for different conditions. In the following, two test cases with comparisons between experimental and numerical results are presented.

4.5.2 Comparison between numerical and experimental results

In this section, the CV-IGA karst flow model is used to simulate experimental data obtained with the novel 3D physical model. The water levels in both the

upstream and downstream reservoirs are implemented as Dirichlet boundary conditions by specifying constant head values. Above the water levels in the reservoirs (up to the top of the soil), the no-flow (flux $q_N = 0$) Neumann boundary condition is used since there is no development of the seepage face. The same no-flow condition is used on remaining boundaries except on the top soil layer, where rain is specified by a nonzero flux value. Evaporation is neglected due to the short duration of the experiment. The conduit upstream boundary is defined as a no-flow condition, whereas the downstream boundary depends on the flow conditions.

Test case 1

In the first test case, a straight circular pipe with perforations uniformly distributed along the pipe (labeled as conduit C1) is considered. The position of the conduit is defined by the starting and ending coordinates of the pipe bottom, $L_1 = (0.36, 1.47, 0.75)$ and $L_3 = (4.97, 1.47, 0.70)$. The conduit diameter is 0.0155 m, and the Manning roughness coefficient is $n_M = 0.009$ m$^{1/3}$ s. The water levels in two reservoirs are kept fixed at $H_{upstream} = 1.555$ m and $H_{downstream} = 1.463$ m. Since for this test case, the conduit flow is pressurized, the Dirichlet boundary condition at outlet is defined by hydraulic head value which is equal to conduit top elevation (the atmospheric pressure at the outlet). Numerical results are obtained using matrix spatial discretization $\Delta x_M = \Delta z_M = 0.1$ m and $\Delta y_M = 0.102$ m, whereas the conduit discretization is set as $\Delta x_C = 0.02305$ m. The same time step $\Delta t = 1.0$ min was used for both matrix and conduit solvers, and the matrix-conduit exchange coefficient was set after calibration to $\alpha_{ex} = 0.18$ s^{-1}. Comparisons between the measured and calculated values for discharges and selected hydraulic heads are shown in Figures 4.8 and 4.9, respectively.

Before the measurement were started, the steady-state matrix flow (blue line) is established, while the conduit outlet is clogged (see Figure 4.8). $t = 0$ is defined as the moment at which conduit C1 is opened. Both the conduit (red line) and matrix responses are very fast due to high matrix conductivities. Basically, the conduit represents a highly permeable zone and drains groundwater from the surrounding porous medium. At $t = 20$ min, sprinkler rainfall of total discharge 10.17 L/min has started, whereas its influence on matrix discharge begins to be visible approximately 10 min later. This delay is mostly because of slow water infiltration through the unsaturated zone. Further, the matrix discharge is slowly raising and starts to approach a steady-state value after $t = 50$ min, whereas the conduit discharge remains

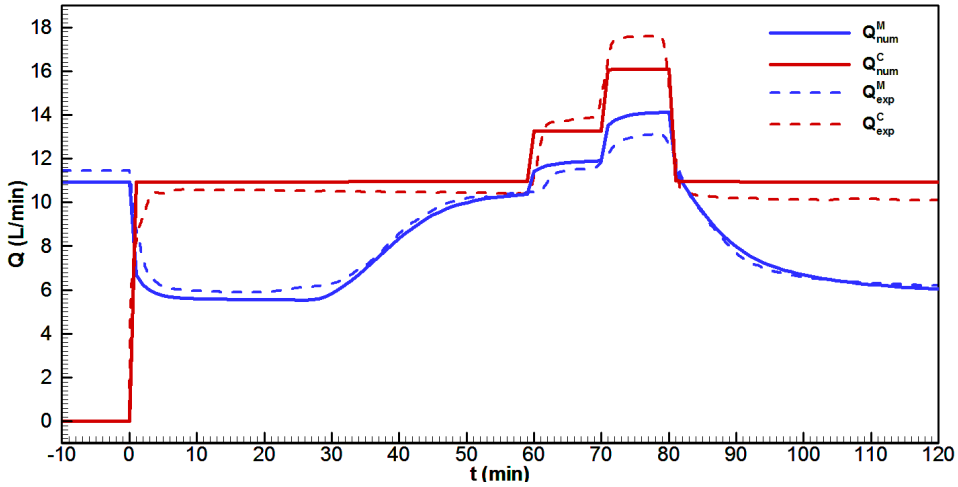


FIGURE 4.8: Hydrograph: comparison between experimental (dashed line) and numerical (solid line) results for matrix (M - blue line) and conduit discharge (C - red line), respectively.

practically the same. The reason for this is explained as follows. The conduit discharge can be increased by increasing matrix pressure in the vicinity of the conduit. However, due to high overall matrix permeability and moderate rainfall intensity, the increase in the matrix pressure around the conduit is not sufficiently high to be visible on conduit discharge. Thus, to achieve more significant variations of matrix-conduit exchange, the direct recharge (5.80 L/min) through conduit sinkhole (connected with upstream end of the conduit) is started at $t = 60$ min. The applied direct recharge produces increase in the conduit's pressure, and significant effects on both matrix and conduit flow are achieved. The numerical results reveal that increased conduit pressure produced by direct recharge has only reduced exchange discharge, and there is still no flow from conduit toward the matrix (head inversion). The total increase in conduit discharge is smaller than applied recharge because the matrix inflow is now reduced. Moreover, in the $t = 70$ min, the direct recharge is increased from 5.80 L/min to 14.65 L/min and further increase in conduit pressure now produces head inversion, but only in the upstream part of the model. In the $t = 80$ min both rainfall and direct recharge are turned off, while measurements are continued until end of experiment ($t = 120$ min).

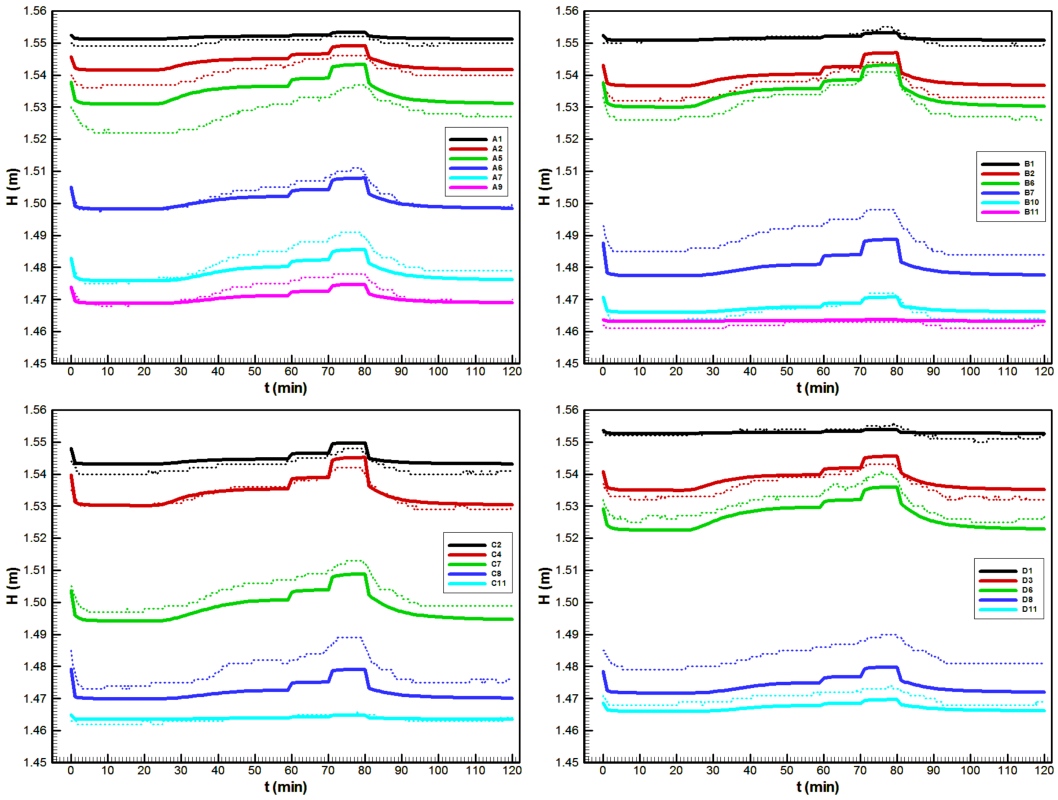


FIGURE 4.9: Hydraulic head: comparison between experimental (dashed line) and numerical (solid line) results.

The hydrograph comparison shows that numerical model follows the behavior of experimental results. The differences are mostly inside range of measurement error, which is estimated to be around 1.0 L/min. The numerical results are sharper then the experimental ones, which are smoothed by the measuring equipment.

The hydraulic head comparison in Figure 4.9 also demonstrates good agreement of the results. The precision of the measurements is mostly influenced by precision of photogrammetry and is approximately 1.5 cm. For the most results, this error is lower. Figure 4.10 shows the hydraulic head comparison for two piezometers that are closest to the conduit C1 and mostly influenced by matrix-conduit interaction. Again, the experimental results are slightly smoothed mostly because of time lag produced by filling/emptying the piezometers. These results clearly demonstrate that numerical model is capable to capture realistic flow dynamics of physical model.

Further, Figure 4.11 demonstrates conduit solutions in three representative time steps. This results confirm that head inversion can only happen from $t = 70 - 80$ min because for the remaining time, the hydraulic head in conduit is lower then minimum matrix head defined by the downstream

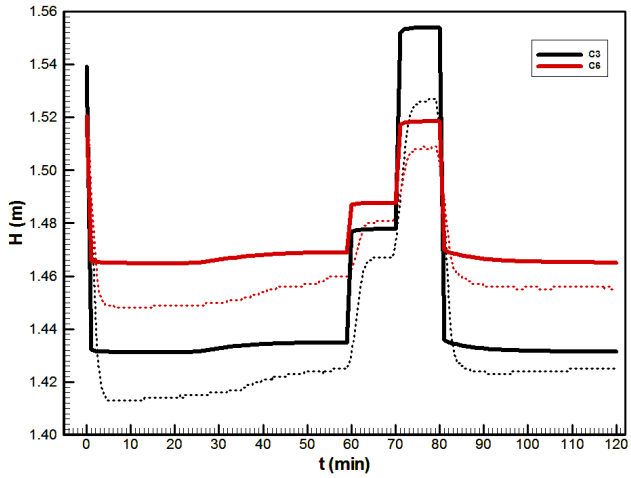


FIGURE 4.10: Hydraulic head: comparison between experimental (dashed line) and numerical (solid line) results for two most dynamic piezometers that are closest to the conduits.

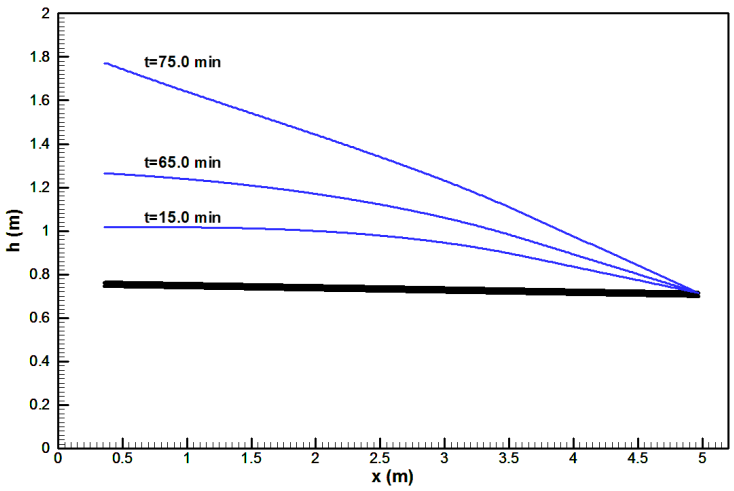


FIGURE 4.11: Conduit C1 (black lines) and the hydraulic head h (m) (blue lines) at different time steps.

reservoir ($H_{downstream} = 1.463$ m). Additionally, the calculated Reynolds number at conduit outlet was between 15000 and 22000 during simulation.

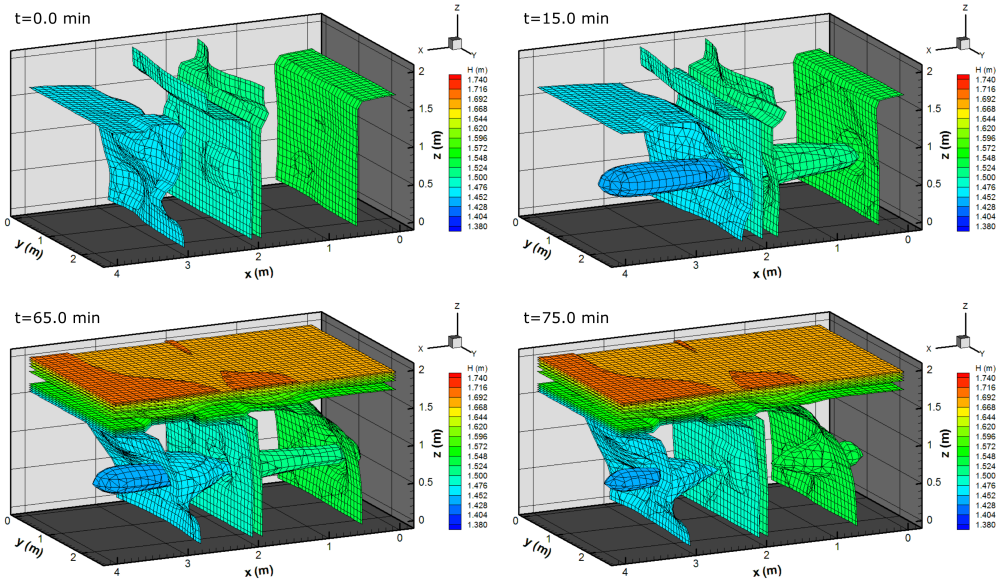


FIGURE 4.12: Contours of matrix hydraulic head H (m) at different time steps.

Finally, Figure 4.12 demonstrates matrix's hydraulic head solutions at different time steps. At $t = 0$ min (before conduit opening), the steady-state matrix solution is shown, whereas at $t = 15$ min, a high conduit influence on matrix flow is clearly visible. At $t = 65$ min, the matrix is influenced both by rainfall and direct recharge from the conduit, whereas at $t = 75$ min, the direct recharge is increased and head inversion is visible in the upstream part of the model.

Test case 2

For the second test case, a straight circular pipe with larger diameter that is perforated only at the upstream part of the model (C2 conduit) is used. The position of the conduit is defined by starting and ending coordinates of the pipe bottom $M_1 = (0.40, 1.64, 0.75)$ and $M_3 = (5.06, 1.64, 0.70)$, while the conduit is perforated only for $x \in [0.4, 2.1]$. The conduit diameter is 0.042 m, and the Manning roughness coefficients are $n_M = 0.044$ m$^{1/3}$ s and $n_M = 0.015$ m$^{1/3}$ s for perforated and unperforated parts, respectively. Water levels in two reservoirs are kept fixed at $H_{upstream} = 1.515$ m and $H_{downstream} = 1.351$ m. Since for this test case, there was free surface flow at the conduit outlet, the free outfall boundary condition [72] was used. This condition is implemented as a Dirchlet boundary condition, where the hydraulic head is calculated by using smaller value between critical and normal water depth. The same numerical discretization is used as in previous test case, while the

exchange coefficient was set after calibration $\alpha_{ex} = 0.23$ s^{-1}. Comparisons between the measured and calculated values for discharges and hydraulic heads are shown in Figures 4.13 and 4.14, respectively.

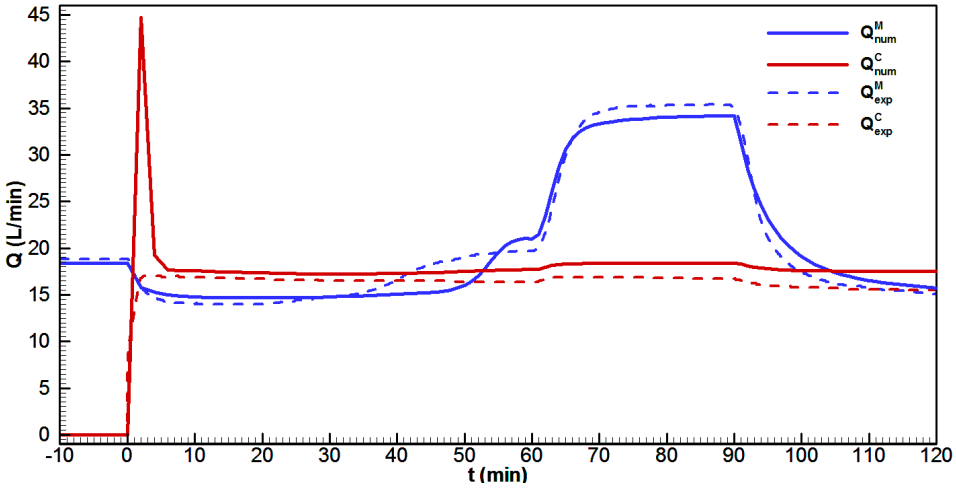


FIGURE 4.13: Hydrograph: comparison between experimental (dashed line) results and numerical (solid line) results for matrix (M - blue line) and conduit discharge (C - red line), respectively.

As for the previous test case, the steady state matrix flow (blue lines) is established before start of measurements. In $t = 0$, the conduit C2 is opened, and conduit discharge (red lines) is increased instantaneously. In the first time step, the numerical model shows much higher discharge value then one obtained by experiment. This value is the outcome of the initial conduit emptying and is perfectly meaningful. Because of larger pipe diameter, there is free surface (partially filled) flow at the downstream end; thus, the conduit needed to release excessive water, which was initially stored in the conduit. The measuring equipment used for experimental data was not able to capture this effect (see how discharges are measured in supplementary materials). Additionally, this behavior was not observed in the previous test case (C1 conduit) because of smaller pipe diameter and the fact that flow in the conduit remained pressurized.

At time $t = 10$ min, the water depth at outlet boundary was measured to be 19 mm, while the numerical model predicted a value of 22 mm. At $t = 20$ min, sprinkler rainfall of total discharge 12.00 L/min has started. At this time, the rainfall influence on the matrix discharge is observed later then indicated by the experimental data. The good agreement of rainfall infiltration obtained in the first test case suggests that main cause could be coarse quartz sand (CQS), which dominates in additional infiltration area that was

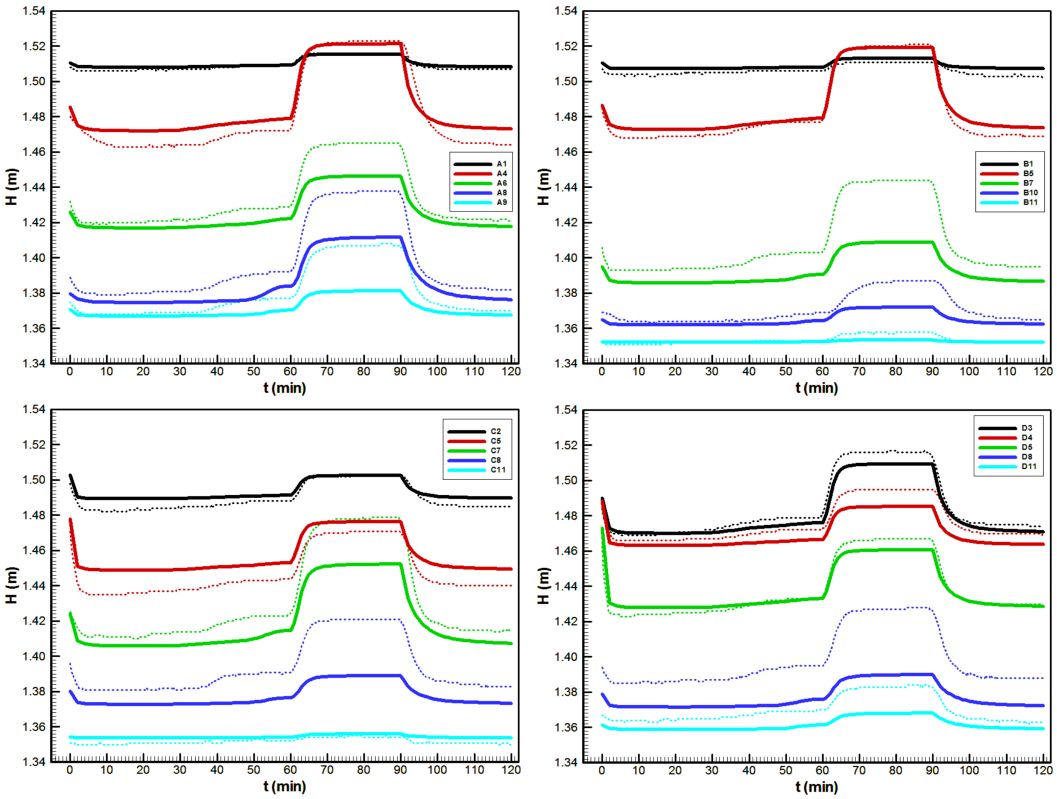


FIGURE 4.14: Hydraulic head: comparison between experimental (dashed line) and numerical (solid line) results.

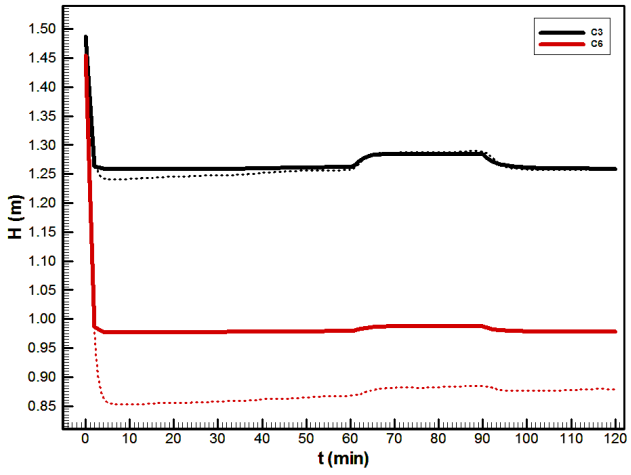


FIGURE 4.15: Hydraulic head: comparison between experimental (dashed line) and numerical (solid line) results for the two most dynamic piezometers that are closest to the conduits.

saturated in previous test case (the water table is lower in this case). Even though the water retention curve for this material was obtained in the laboratory, it is well known that the accurate measurement of soil parameters in the unsaturated zone is a very difficult task. For example, the soil hysteresis

or initial saturation deviations can significantly affect results due to the high nonlinearity of water retention curves.

Moreover, as in the previous case, the rainfall did not significantly affect conduit discharge; thus, at the time $t = 60$ min, the rainfall intensity is increased by turning on two additional shower heads. The total discharge of 42.30 L/min is distributed on square area defined by two corner points $R_1 = (1.20, 0.80, 2.00)$ and $R_2 = (2.10, 1.70, 2.00)$, while keeping the same sprinkler rainfall (12.00 L/min) distributed on overall top area. This high-intensity rain now produces a fast and significant increase in matrix discharge, but also, an increase in conduit discharge is observed. The maximum matrix discharge value was slightly underestimated by numerical model, which is followed by underestimated pressures in downstream part of model, meaning that concentrated rainfall influence is distributed something differently than was predicted by the numerical model. The imperfection of the measurements of rainfall distribution and models heterogeneity are probably the main reasons for these differences. The increase in conduit discharge is significantly smaller then in the matrix because high matrix permeability is able to collect most of the rainfall water and prevents a significant increase in matrix pressures. Both the sprinkler and shower heads rains are turned off in $t = 90$ min, while measurements are continued until the end of the experiment, $t = 120$ min.

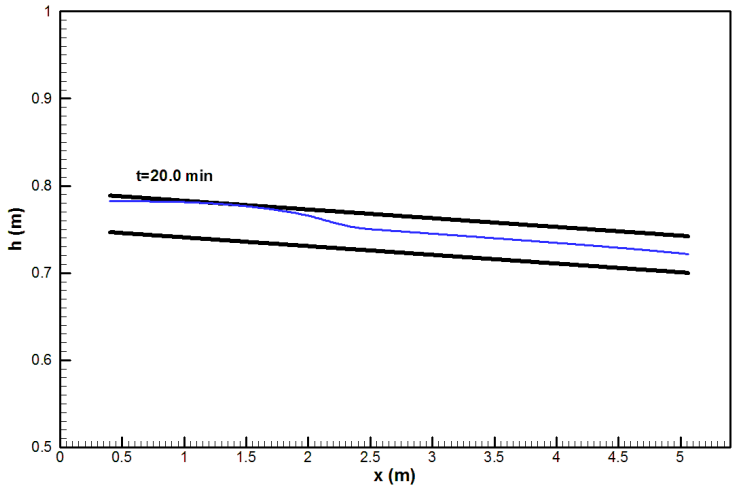


FIGURE 4.16: Conduit C2 (black lines) and the hydraulic head h (m) (blue line) at $t = 20.0$ min.

In this case, the head variations are more pronounced. The pressure drop introduced by opening the conduit is higher then in the previous case, and high-intensity rainfall produces an increase of the head up to 6 cm. The general behavior of hydraulic head is quite realistic, but as mentioned before,

the accuracy is lower in the downstream part of the model. The same two piezometers as in the previous case are most dynamic and shown separately in Figure 4.15. Even though conduit C2 is only partially perforated, it has produced large conduit discharge followed by a significantly large pressure drop (especially observed by piezometer C6 in Figure 4.15). The free surface flow caused by high pipe conductivity and a larger percentage of perforated area are the main reasons for this behavior. The highest pressured drop measured by piezometers is 0.61 m, while the numerical model had a predicted value of 0.48 m. Even though discrete-continuum models are not capable of the detailed resolution of flow interface, it seems that this model quite realistically simulates the hydrodynamics of the physical model.

Figure 4.16 demonstrates the numerical solution for the hydraulic head inside the conduit where it dominates free surface flow. However, in part of the conduit, the water surface reaches the top of the conduit, and flow is slightly pressurized. The conduit solution is shown only for one time step because there are no significant changes in the hydraulic head during the solution. Finally, Figure 4.17 presents the matrix solution in $t = 75$ min.

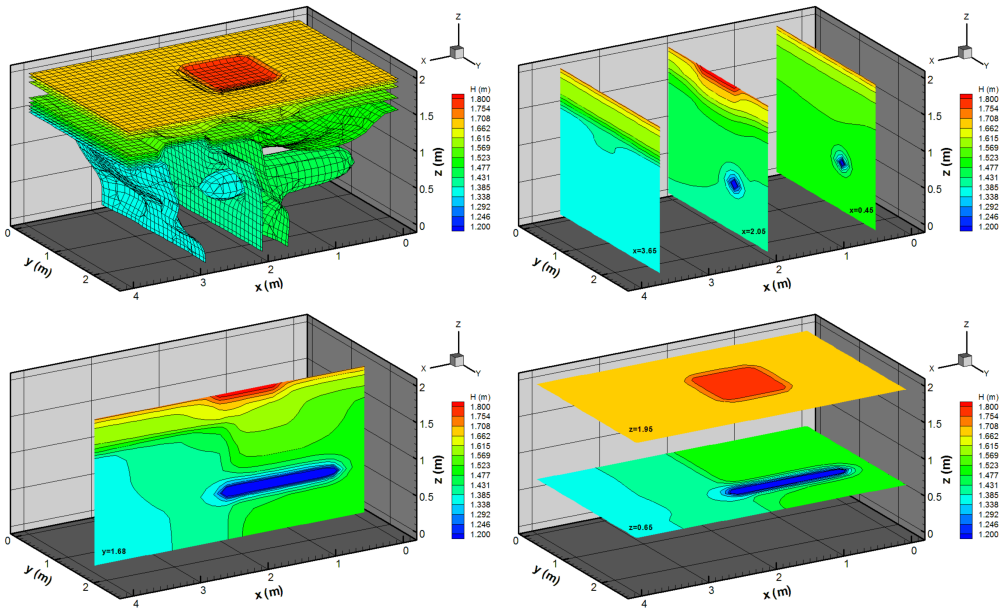


FIGURE 4.17: Contours of matrix hydraulic head H (m) in $t = 75.0$ min.

4.6 Discussion

In this chapter, the development of the hybrid (discrete-continuum) karst flow model is presented. With respect to the wide range of karst distributed

models (continuum, discrete, hybrid, etc. ...), these hybrid models have potential to describe many karst complex properties, especially karst heterogeneity including usually uncertain conduit geometry and soil properties of matrix. Moreover, by taking into account the description of different flow regimes in conduits (turbulent flow) and matrix (laminar Darcy flow) as well as their complex relationship, hybrid models temporally serve as the best candidate for realistic karst flow modeling. The two main issues that appear most important for the development of realistic hybrid karst flow models are as follows: a) its numerical complexity, which requires the solving of high nonlinearity, preserving of numerical stability and adequate description of karst multiphysics, and b) uncertain measurements and sparse collection of input data causing difficult (in many cases, impossible) verification of karst flow models under realistic catchment conditions.

Complex karst flow modeling usually requires even more than the "state-of-the-art" modern numerical techniques. This is the main reason why in this work, a new karst flow model is developed using few important ingredients: a) control volume formulation, which ensures local and global mass balance (e.g., finite element usually enables only global mass balance); b) Fup basis functions inside relatively novel isogeometric analysis – IGA (Hughes *et al.* [6] and Cottrell *et al.* [8]) enable the description of geometry, high approximation properties, continuity of velocities and fluxes, multiresolution and efficient adaptive algorithms (Gotovac *et al.* [11]); and c) handling of high nonlinearity requires presented segregated iterative algorithm and very "careful" modeling between matrix and conduit separate solvers involving underrelaxation for increased stability. Possible further improvements include the solving of the large system of nonlinear equations due to the description of the large catchment area. One possible direction is incorporating this algorithm into the open source code Petiga (Dalcin *et al.* [73]), which is based on robust parallel nonlinear library PETSc (Balay and Brown [74]) and IGA. Furthermore, one bottleneck is the high nonlinearity introduced by Richards' equation and definition and measurements of unsaturated curves. A new improved description of unsaturated zone is highly desirable. Last, but not least, the definition of exchange flux is still an open research topic. In this chapter, classical linear relationship depending on the head differences and questionable first-order exchange parameter was used. Some improvements using the Peaceman well index are given for example in Rooij *et al.* [52].

The realistic verification of the karst flow models is also open research

topic, especially for distributed models such as the presented one. Realistic verification includes knowledge of wide range of parameters and karst properties (for example, Croatian karst aquifers in Jadro and Ombla catchments; see Bonacci [75]). The most needed data are conduit geometry (diameters, positions and network structure), matrix heterogeneity (conductivity and porosity), unsaturated matrix properties in terms of van Genuchten curves and uncertainty of its parameters, boundary conditions such as matrix inland conditions and outlet conduit condition, uncertainty of rainfall distribution, etc. Due to usual lack of collected data, its great number and relatively large number of needed calibration parameters, realistic verification is definitely complex and usually an extremely difficult task. Therefore, as the first step for the verification of the presented karst flow model, in this chapter, it is performed under controlled laboratory conditions.

Therefore, a large 3D physical karst flow model is constructed (5.66 m x 2.95 m x 2.00 m). To the best of our knowledge, this is the first such model that enables many mentioned karst flow characteristics and detailed knowledge of most mentioned input data (see other more simpler laboratory models in Faulkner *et al.* [55] and Castro [56]). Despite the still presented uncertainty in soil properties, most of the input data are defined by measurements and "fine-tuning" by additional smaller flow test experiments (see Appendix AppendixPhysicalKFM). Thus, verification is performed on two different flow test cases, where the only parameters that were calibrated during the presented experiments were the matrix-conduit exchange coefficient and roughness in the perforated part of larger conduit C2. Since it is a substantial improvement in comparison with realistic flow conditions in karst, the presented verification can be regarded as the first attempt for the realistic verification of karst flow models.

The two flow test cases were presented, and numerical results were confronted with experimental data. In the first test case, the matrix flow is disturbed by opening the smaller conduit (diameter: 0.0155 m). The activation of the conduit produces a decrease in both matrix discharge and heads since the conduit represents a highly permeable zone and redirects a significant amount of flow. The total discharge (from both matrix and conduit) is significantly higher, whereas the conduit flow is pressurized due the small conduit diameter and sufficient exchange capacity. Moreover, moderate rainfall was applied to test nonlinear infiltration through unsaturated zone, while direct recharge was used as an additional test for water exchange between the matrix and conduit. While conduits drain the matrix during most of

the experiment, the numerical results show that for maximum applied direct recharge (from $t = 70 - 80$ min), there is head inversion in upstream part where the conduit recharges the matrix. The numerical results are in excellent agreement during most of the experiment and inside the measurement error. The largest deviation from measurement results is observed during direct recharge through conduit. This also can be due to measurement error; however, there is a possibility that an exchange term that uses constant exchange coefficient (i.e., linear dependency of exchange discharge with head difference) is not capable of accurately capturing this range of pressure variations. However, it can be stated that the overall dynamics of the physical model are realistically and accurately simulated by the numerical model.

In the second test case, the matrix flow interacts with the partially perforated pipe/conduit (diameter: 0.042 m). This setup was shown to be more demanding for flow modeling. The free surface conduit flow that uses demanding and unknown outlet boundary condition, longer infiltration depth, higher and concentrated rainfall intensity and larger discharge rates followed by large pressure variations in the matrix are some of the main differences that represent serious challenges for the numerical model. Again, the model captured complex physics very realistically. The largest difference occurred in the model response to applied rainfall. For this case, the water table is lower, and due to deeper infiltration depth, the influence on matrix discharge is observed later than by the experiment. The watering of dry soil represents a very serious problem and is highly influenced by (always) uncertain van Genuchten parameters and imperfection in rainfall distribution. Further, the concentrated high-intensity rainfall produces a lower matrix discharge than measured. Measurement errors are likely to be more pronounced for higher flow rates values, but the underestimation of the hydraulic head in the downstream part of the model suggest that the model response to this rainfall is something different than in the physical model. The imperfection of rainfall distribution, complex heterogeneity and complex physics of infiltration above the water table are the greatest challenges in obtaining accurate results. Finally, measurement errors for both test cases are considered to be dominated by photogrammetry errors, which are estimated at approximately 1.5 cm for head and 1.0 L/min for discharges.

The two presented test cases were significantly different and captured many complicated processes that occur in real karst aquifers. The numerical model was mostly inside the range of measurement precision, and the

observed experimental behavior is realistically simulated, which is an encouraging result and demonstrates the robustness of the developed numerical model. Finally, the first test case used approximately 60 min of CPU time, while second test case needed approximately 81 min. Both calculations were performed by serial computing on a single processor (Intel Core i7, 2.60 GHZ).

In future work, the plan is to use the presented karst flow model to investigate the sensitivity and validity of water exchange term. The exchange coefficient is the main calibration parameter that directly determines matrix-conduit exchange. The numerical model has been verified in the sense that it can produce accurate and realistic results by calibrating the exchange parameter; however, the serious sensitivity analysis of this parameter is not considered. Thus, we are planning to use different experimental setups and investigate how obtained values of exchange parameters describe different flow conditions on the scale of our physical model. This is an important topic for practical application since numerical karst flow models are needed for the prediction of different scenarios and not only for understanding and reproducing obtained measured data. Finally, the quality of the velocity results of developed karst flow model will enable more realistic analysis of tracer tests and contaminant transport.

4.7 Conclusions

In this chapter, a novel numerical model for groundwater flow modeling in karst aquifers is presented. The hybrid (discrete-continuum) approach is used as the best candidate for realistic karst flow modeling. Due to the high nonlinearity introduced by the unsaturated zone, most karst flow models neglect important hydrodynamic behavior above the water table. Thus, in this work, matrix flow is described by a more realistic variably saturated flow equation (usually called Richards' equation). Additionally, the mixed form of the equation is used to obtain mass conservative solution in both the saturated and unsaturated zones. The flow in the karst conduits is represented as one-dimensional and captured by a noninertia wave equation. This equation conveniently describes both free surface and pressurized flow conditions by simply modifying the capacity term in governing equations. Since two flow domains (i.e., matrix and conduit) are governed by different partial differential equations (as well as dimensionality), the equations need to be coupled.

The coupling is established on a classical first-order exchange term, which is proportional to the head difference on the matrix-conduit interface.

A novel numerical technique, developed in Chapter 3 and labeled control volume isogeometric analysis (CV-IGA), is used for spatial discretization. The classical backward Euler method is used for temporal discretization, while linearization is based on Picard iterative scheme. The karst flow model is based on a segregated (iteratively coupled) scheme, in which each flow domain is solved independently by using the last known solution of the other domain to calculate the exchange term.

A particular difficulty with complex 3D karst flow models is the verification and validation process due to lack of input (such as conduits geometry and parameters of aquifer) and measured (head distribution and discharges) data. Therefore, in this work, a novel 3D physical karst flow model capable to reproduce many important flow features in karst is presented. This physical model enables substantial improvement in comparison with real aquifers, especially in terms of the detailed knowledge of mentioned input and measured data. Two complex experimental setups were used and simulated by the numerical model. The numerical model was mostly inside the range of measurement precision and has been capable of realistically simulating demanding flow in the physical model.

Chapter 5

Full space-time adaptive Fup collocation method for advection-dominated problems

This chapter presents a full space-time adaptive collocation algorithm that is particularly efficient for modeling advection-dominated problems. The automatic grid adaptation is based on the previously developed Adaptive Fup Collocation Method (Gotovac *et al.* [11]), while additional computational speedup is achieved via the construction of a novel adaptive strategy in time through an efficient local time-stepping procedure. The spatial strategy presented in the following chapter is different from the approaches used previously in this work. Thus, the presented algorithm could be regarded as a hybrid approach in which the collocation formulation and Fup basis functions are used for interpolation and grid adaptation and spatial derivatives (occurring in the governing PDEs) are discretized by the construction of the finite difference or the (novel) local collocation operator.

5.1 Introduction

Many engineering problems exhibit a wide range of spatial and/or temporal scales, which are usually arbitrarily distributed in space and time. Numerical modeling of such processes typically faces many difficulties, particularly for accurate resolution of highly localized step gradients while controlling spurious numerical oscillations and excessive numerical diffusion. Adaptive numerical modeling is one efficient way to address these mentioned numerical problems.

The first attempt at the application of space adaptive numerical procedures has been focused upon using classical finite element/volume methods

[76]–[84]. Moreover, there have been many attempts to develop new adaptive procedures which, among others, are focused on using an adaptive wavelet collocation methods [85]–[89] and adaptive spline collocation methods [19], [90], [91]. Adaptive Fup Collocation Method (AFCM, [11], [24]) with application to the groundwater flow and transport problems [92], [93] was also developed earlier.

All mentioned methods use only a spatial adaptive procedure, while temporal discretization is obtained by classical time-marching procedures that utilize the same time step for all spatial locations. It is well known that, in that case, the time step is restricted by the highest spatial resolution level (CFL—Courant–Friedrichs–Lewy condition). To overcome this restriction, local time stepping was introduced by Osher and Sanders [94]. Among others, description of space adaptive methods with local time stepping can be found in [95]–[101]. These methods are based on applying different local time steps in different parts of the spatial domain, where CFL condition needs to be satisfied only locally. Müller and Stiriba [96] applied a methodology for explicit and implicit methods, while other mentioned authors mostly studied explicit Runge-Kutta methods.

In this work, a novel implicit algorithm that is adaptive in both space and time is presented. Contrary to common space-time adaptive methods, the developed method resolves all spatial and temporal scales independently of each other based on separate error criteria.

The methodology is based on second-order Fup_2 basis functions. Fup_2 basis function (used in this chapter) and its first three derivatives are shown in Figure 2.5.

In addition to the full space-time adaptive procedure, this chapter describes the construction of a novel spatial collocation operator for accurate approximation of spatial derivatives. Although Fup basis functions can be used to define a trial function space for numerical approximation as it was presented in Chapters 3—4, in this chapter, a localized properties of Fup functions are used to create a local collocation operator similar to the finite difference (FD) operator. However, besides neighboring solution values, the developed operator includes physical constraints defined by governing differential equation, which makes the operator problem dependent but also significantly improves the approximation properties compared with the common FD operator. The novel operator is currently limited to the 1D problems, and 2D example presented in this chapter uses classical FD operator. The main reasons for construction of such FD-type operator are the properties

of adaptive spatial strategy, where creation of stable multilevel trial function space for adaptive collocation approximation is not straightforward.

All concepts are described in details for 1D initial-boundary value problem (IBVP) governed by an advection-diffusion equation (ADE). However, first 2D results are included to demonstrate that proposed space-time adaptive strategy can be efficiently applied for multidimensional problems.

5.2 Methodology

This section addresses a detailed description of the **A**daptive **F**up **C**ollocation **M**ethod (AFCM), including its features in space and time. First, a discrete form of the advection-diffusion equation (ADE) is derived by a collocation approach and the construction of Fup collocation operators for approximation of spatial and temporal derivatives. Next, adaptive strategies in space and time are described. The section ends by describing some additional details and possibilities of the methodology. Before pursuing details, a short chronology overview of the space-time adaptive algorithm is given.

At the start of calculation it is assumed that the initial condition is given as an analytic function. Then, the adaptive Fup collocation transform (FCT) algorithm is used to approximate (interpolate) the known initial condition with the prescribed accuracy. The result of this transform is an adaptive non-uniform grid of collocation points where finer spatial resolutions are associated only with parts of the domain where the solution changes are demanding. This arrangement of collocation points represents the appropriate computational grid for spatial discretization of the partial differential equation (PDE) for the current global time step. Note that the term *global* time step is defined as the time interval along which the spatial adaptive grid remains unchanged. The global time step is not the computational time step for discretization of the PDE, and it will be denoted as Δt_{global} to distinguish it from the *local* (discretization) time step Δt used for time-marching. Local time step size is chosen according to global time step $\left(\Delta t = 2^{-j_{\min T}} \cdot \Delta t_{global}\right)$, where parameter $j_{\min T}$ determines number of local time steps inside single global time step.

Using the same local time step for all collocation points the discrete solution at the end of the global time step is reached. Then, calculation is repeated with a twice-smaller local time step, and differences between the two solutions are compared for each collocation point individually. For those collocation points where the absolute difference between solutions is larger than

some user-prescribed threshold, calculation is repeated using a new, again twice-smaller, local time step. Other points are excluded for further calculation. This procedure is repeated until the difference between the current and previous solution for each kept collocation point is not below the prescribed threshold.

The final solution at the end of a global time step is a set of discrete values in collocation points. Before moving on to the next global time step, it is necessary to transform the discrete solution into the continuous function. This is possible by applying FCT again, satisfying calculated solution values on the same spatial adaptive grid obtained at the beginning of the global time step. The result of this transform is a smooth Fup representation (with all smooth derivatives) of the new initial condition for the next global time step.

The procedure is then repeated for all global time steps while the adaptive strategy dynamically changes the grid in space and time and accurately follows solution changes. A flow chart of the methodology is illustrated in Figure 5.1.

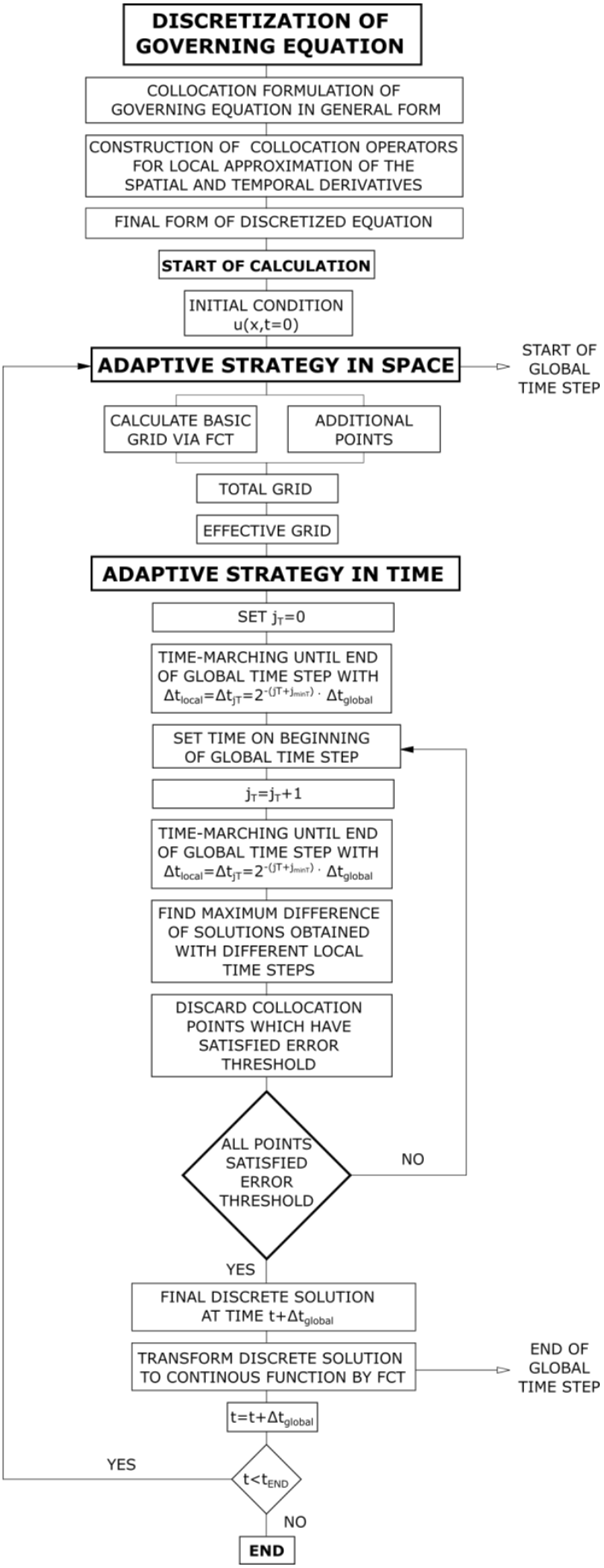


FIGURE 5.1: Flow chart for the AFCM methodology.

5.2.1 Discretization of the governing equation

Collocation formulation of governing equation

Discretization of the PDE will be presented on a 1D advection-diffusion equation (ADE) in the form:

$$\frac{\partial u(x,t)}{\partial t} = D(x,t)\frac{\partial^2 u(x,t)}{\partial x^2} - v(x,t)\frac{\partial u(x,t)}{\partial x}, \quad x \times t \in (a,b) \times (0,\infty) \tag{5.1}$$

with the appropriate initial:

$$u(x,0) = U_0(x) \tag{5.2}$$

and boundary conditions:

$$u(a,t) = G(t), \quad \frac{\partial u(b,t)}{\partial x} = H(t) \tag{5.3}$$

Here, u is solution of the initial-boundary value problem (5.1)-(5.3), and D and v are linear or nonlinear functions of diffusion and average velocity, respectively. U_0, G and H are prescribed initial and boundary functions. Following the classical collocation approach (e.g., [102]) and evaluating (5.1) at each i-th internal collocation point at end of local time step yields the implicit collocation formulation in the form:

$$\frac{\partial u_i^{t+\Delta t}}{\partial t} = D_i^{t+\Delta t}\frac{\partial^2 u_i^{t+\Delta t}}{\partial x^2} - v_i^{t+\Delta t}\frac{\partial u_i^{t+\Delta t}}{\partial x}, \quad i = 2, ..., n-1 \tag{5.4}$$

where n is the number of all collocation points (internal and boundary) while subscript and superscript are introduced to denote spatial and temporal coordinates, respectively. The same collocation procedure for boundary conditions (5.3) derives boundary equations as:

$$u_1^{t+\Delta t} = G^{t+\Delta t}, \quad \frac{\partial u_n^{t+\Delta t}}{\partial x} = H^{t+\Delta t} \tag{5.5}$$

At the end of each local time step, a differential-algebraic system of equations (5.4)-(5.5) satisfies the differential equation (5.1) and boundary conditions (5.3) in all internal and boundary collocation points, respectively. This system of equations still contains solution derivatives that can be expressed in terms of solution values in the surrounding collocation points to obtain

a fully discretized algebraic system of equations. The construction of spatial and temporal collocation operators for derivative approximations is explained in the sequel.

Approximation of spatial derivatives

The goal is now to construct a collocation operator in the general form that can be used for numerical approximation of spatial derivatives for any collocation point. One possible way is to fit a local polynomial through solution values of corresponding and few neighboring collocation points and to differentiate the resulting curve. On the uniform grid, this reduces to a classical finite difference operator, which is a common approach in many adaptive techniques [e.g., [11], [89]]. In this chapter, a novel collocation operator that uses local Fup approximation around a corresponding collocation point is constructed. The operator will be developed by assuming a uniform collocation stencil, while situations where this condition is not fulfilled will be addressed later.

For every internal collocation point, it is possible to find two neighboring points to create a local computational stencil. This stencil defines a local subdomain that consists of three collocation points and two characteristic intervals, as shown in Figure 5.2.

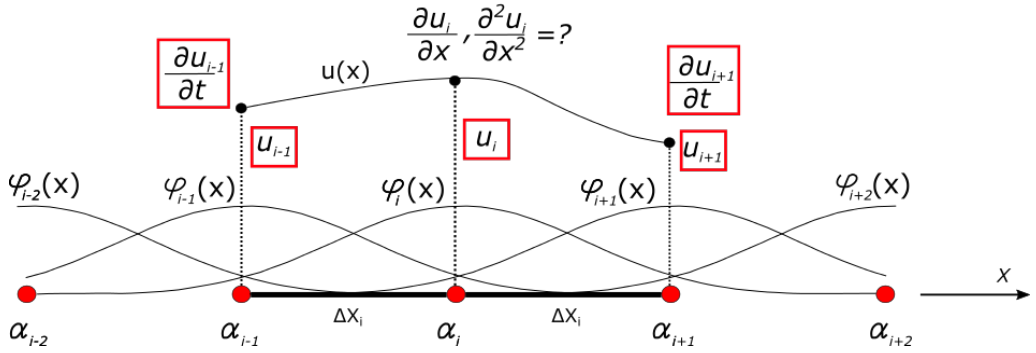


FIGURE 5.2: Local subdomain approximation for construction of a spatial collocation operator.

Local Fup approximation of an unknown solution and its m-th derivatives can be expressed as:

$$u(x) = \sum_j \alpha_j \cdot \varphi_j(x) \tag{5.6}$$

$$\frac{\partial^{(m)} u(x)}{\partial x^{(m)}} = \sum_j \alpha_j \cdot \frac{\partial^{(m)} \varphi_j(x)}{\partial x^{(m)}} \tag{5.7}$$

where α_j are local Fup coefficients and $\varphi_j = \varphi_j(x)$ are Fup basis functions. Note that the dependency of time is not included because the construction of a spatial operator is generally independent of time.

To derive a collocation operator in general form, coefficients α_j have to be defined. For consistent approximation, there are five non-zero Fup_2 basis functions on the local subdomain, as shown in Figure 5.2. Hence, to obtain a unique collocation operator, five conditions need to be enforced. An obvious way is to impose solution values in three collocation points (*i-1*, *i*, *i*+1):

$$\sum_{j=k-1}^{k+1} \alpha_j \varphi_j(x_k) = u_k, \quad k = i-1,\ i,\ i+1 \tag{5.8}$$

where only three of five basis functions are non-zero in the *k*-th collocation point. For the additional two conditions, one possibility is to set a third derivative of the solution to zero in the middle of both characteristic intervals. In this manner, the local Fup approximation is reduced to the second-order polynomial, and the collocation operator is the same as the second-order central finite difference operator.

In this work, another possible choice is used by satisfying the differential equation (5.4) in the subdomain boundary points (*i-1* and *i+1*):

$$D_k \sum_{j=k-1}^{k+1} \alpha_j \frac{\partial^2 \varphi_j(x_k)}{\partial x^2} - v_k \sum_{j=k-1}^{k+1} \alpha_j \frac{\partial \varphi_j(x_k)}{\partial x} = \frac{\partial u_k}{\partial t}, \quad k = i-1,\ i+1 \tag{5.9}$$

Note that only spatial derivatives from equation (5.4) are expressed as a linear Fup combination (5.7). Other variables, such as coefficients of the PDE and temporal derivative, are treated as known values. In essence, equation (5.9) expresses a linear combination of spatial derivatives specified by the differential equation.

The local system of (5.8)-(5.9) can be represented in matrix form:

$$\begin{bmatrix} D_{i-1}\frac{\partial^2 \varphi_{i-2}(x_{i-1})}{\partial x^2} - v_{i-1}\frac{\partial \varphi_{i-2}(x_{i-1})}{\partial x} & D_{i-1}\frac{\partial^2 \varphi_{i-1}(x_{i-1})}{\partial x^2} - v_{i-1}\frac{\partial \varphi_{i-1}(x_{i-1})}{\partial x} & D_{i-1}\frac{\partial^2 \varphi_i(x_{i-1})}{\partial x^2} - v_{i-1}\frac{\partial \varphi_i(x_{i-1})}{\partial x} & 0 & 0 \\ \varphi_{i-2}(x_{i-1}) & \varphi_{i-1}(x_{i-1}) & \varphi_i(x_{i-1}) & 0 & 0 \\ 0 & \varphi_{i-1}(x_i) & \varphi_i(x_i) & \varphi_{i+1}(x_i) & 0 \\ 0 & 0 & \varphi_i(x_{i+1}) & \varphi_{i+1}(x_{i+1}) & \varphi_{i+2}(x_{i+1}) \\ 0 & 0 & D_{i+1}\frac{\partial^2 \varphi_i(x_{i+1})}{\partial x^2} - v_{i+1}\frac{\partial \varphi_i(x_{i+1})}{\partial x} & D_{i+1}\frac{\partial^2 \varphi_{i+1}(x_{i+1})}{\partial x^2} - v_{i+1}\frac{\partial \varphi_{i+1}(x_{i+1})}{\partial x} & D_{i+1}\frac{\partial^2 \varphi_{i+2}(x_{i+1})}{\partial x^2} - v_{i+1}\frac{\partial \varphi_{i+2}(x_{i+1})}{\partial x} \end{bmatrix} \cdot \begin{bmatrix} \alpha_{i-2} \\ \alpha_{i-1} \\ \alpha_i \\ \alpha_{i+1} \\ \alpha_{i+2} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_{i-1}}{\partial t} \\ u_{i-1} \\ u_i \\ u_{i+1} \\ \frac{\partial u_{i+1}}{\partial t} \end{bmatrix} \tag{5.10}$$

where values of Fup basis functions in collocation points are known numbers, while Fup derivatives can be expressed as function of Δx_i only. The

solution of system 5.10 yields five local Fup coefficients (Appendix B.1):

$$\alpha_j = \alpha_j \left(u_{i-1}, u_i, u_{i+1}, \frac{\partial u_{i-1}}{\partial t}, \frac{\partial u_{i+1}}{\partial t}, D_{i-1}, v_{i-1}, D_{i+1}, v_{i+1}, \Delta x_i \right), \quad j = i-2, ..., i+2 \tag{5.11}$$

Finally, by substituting the calculated Fup coefficients α_j into (5.7), the collocation operator is obtained in general form and can be used for approximation of spatial derivatives that arise in (5.4). The collocation operator and semi-discrete form of equation (5.4) obtained after applying the developed operator are shown in Appendix B.1. Treatment of temporal derivatives and possible nonlinearity that can arise from implicit time discretization will be explained in the following sections.

The developed operator is problem-dependent, and different PDE's need a particular construction of a spatial operator.

A boundary point with the Neumann boundary condition requires approximation of the first spatial derivative on the domain boundary. The local subdomain for the boundary collocation point is defined with the boundary and two internal neighboring points (*n-2, n-1, n*), and the same developed local Fup approximation as for central point is still valid. The only difference is the position where the derivative on the local subdomain is calculated from (5.7).

Approximation of the temporal derivative

After application of the developed Fup spatial collocation operator to the semi-discrete equation (5.4), three temporal derivatives are obtained (one for every point from the computational stencil, as shown (B.5) in Appendix B.1). These derivatives have to be approximated to obtain the final algebraic system of equations.

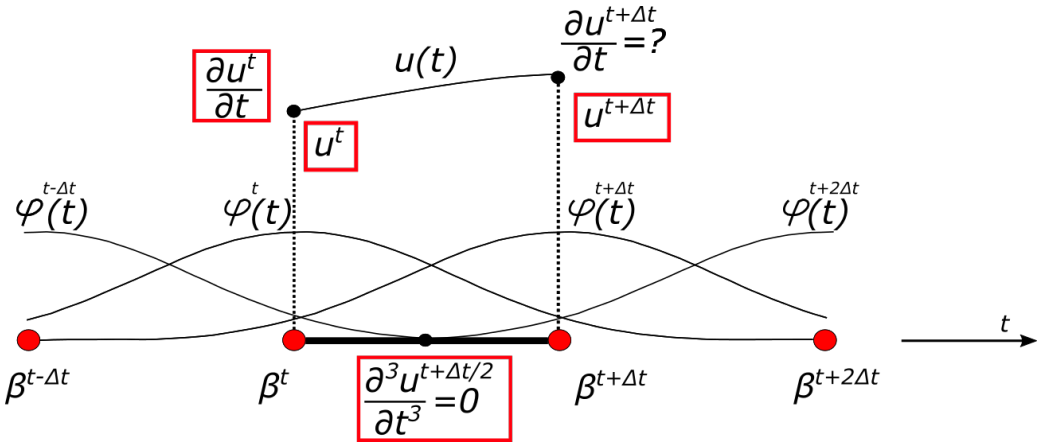


FIGURE 5.3: Local Fup approximation along the temporal line.

For each collocation point in space, there is one temporal line along which the solution changes in time direction. Local Fup approximation (during one local time step) on the temporal line is generally independent of the space location and can be derived by assuming function u as a function of time only:

$$u(t) = \sum_j \beta^j \cdot \varphi^j(t) \tag{5.12}$$

where β^j are temporal local Fup coefficients and $\varphi^j = \varphi^j(t)$ are Fup basis functions defined on the temporal line. Approximation of the temporal derivative can be obtained by differentiating (5.12):

$$\frac{\partial u(t)}{\partial t} = \sum_j \beta^j \cdot \frac{\partial \varphi^j(t)}{\partial t} \tag{5.13}$$

During a local time step Δt, four Fup_2 basis function are needed for full local approximation, as illustrated in Figure 5.4. Two conditions (zero and first time derivatives) describe the initial conditions, while a third condition satisfies the unknown solution value at the end of the local time step. Due to the absence of a meaningful fourth condition, the last equation says that the third derivative is zero in the middle of the local time step, forcing the local Fup approximation to be the same as the second-order polynomial approximation. The local system for temporal Fup coefficients has the form:

$$\begin{bmatrix} \varphi^{t-\Delta t}(t) & \varphi^{t}(t) & \varphi^{t+\Delta t}(t) & 0 \\ \frac{\partial \varphi^{t-\Delta t}(t)}{\partial t} & \frac{\partial \varphi^{t}(t)}{\partial t} & \frac{\partial \varphi^{t+\Delta t}(t)}{\partial t} & 0 \\ 0 & \varphi^{t}(t+\Delta t) & \varphi^{t+\Delta t}(t+\Delta t) & \varphi^{t+2\cdot\Delta t}(t+\Delta t) \\ \frac{\partial^3 \varphi^{t-\Delta t}(t+\Delta t/2)}{\partial t^3} & \frac{\partial^3 \varphi^{t}(t+\Delta t/2)}{\partial t^3} & \frac{\partial^3 \varphi^{t+\Delta t}(t+\Delta t/2)}{\partial t^3} & \frac{\partial^3 \varphi^{t+2\cdot\Delta t}(t+\Delta t/2)}{\partial t^3} \end{bmatrix} \cdot \begin{bmatrix} \beta^{t-\Delta t} \\ \beta^{t} \\ \beta^{t+\Delta t} \\ \beta^{t+2\cdot\Delta t} \end{bmatrix} = \begin{bmatrix} u^t \\ \frac{\partial u^t}{\partial t} \\ u^{t+\Delta t} \\ 0 \end{bmatrix} \tag{5.14}$$

As in the case of spatial approximation, Fup coefficients are a solution of the above system (5.14) and can be described as a function of initial conditions and unknown solution values at $t + \Delta t$ (Appendix B.2). Substituting the calculated β^j into (5.13), the temporal collocation operator for the time derivative at the end of the local time step is obtained:

$$\frac{\partial u^{t+\Delta t}}{\partial t} = 2 \cdot \frac{u^{t+\Delta t} - u^t}{\Delta t} - \frac{\partial u^t}{\partial t} \tag{5.15}$$

By simple rearrangement, equation (5.15) can be written in the following form:

$$\frac{u^{t+\Delta t} - u^t}{\Delta t} = \frac{1}{2} \cdot \left(\frac{\partial u^{t+\Delta t}}{\partial t} + \frac{\partial u^t}{\partial t} \right) \tag{5.16}$$

This form clearly illustrates that implicit Fup approximation in time is identical to the Crank-Nicolson scheme, which is a consequence of the fourth condition in (5.14) that reduces Fup approximation to the second-order polynomial approximation.

In extremely nonlinear and stiff problems, it is well known that the Crank-Nicolson scheme can have convergence problems and cause non-physical results due to the lack of boundedness property (e.g., [103], [104]). In those cases, it is rational to use a more stable lower-order method, such as backward Euler. Due to the universal property of Fup vector space, any polynomial of the same or lower order than that of the used Fup basis function can be exactly represented. This means that backward Euler can also be derived with Fup_2 basis functions. One simple way to derive backward Euler method is to replace the second equation from (5.14) with equation that sets second temporal derivative to zero in the middle of the time step. In that case the Fup approximation becomes exactly a first-order polynomial. Finally, approximation (5.15) for all three temporal derivatives that occur in a semi-discrete equation for collocation point i can be expressed as:

$$\frac{\partial u_k^{t+\Delta t}}{\partial t} = 2 \cdot \frac{u_k^{t+\Delta t} - u_k^t}{\Delta t} - \frac{\partial u_k^t}{\partial t}; \qquad k = i-1,\, i,\, i+1 \tag{5.17}$$

In the following, term (5.17) is used for approximation of temporal derivatives unless stated otherwise.

Final form of discretized equations

Substitution of Fup temporal derivatives (5.17) into the semi-discretized equation (B.5) and writing equations for all internal collocation points together with boundary equations results in a final set of discretized algebraic equations, written in matrix form as:

$$\mathbf{A}\mathbf{u} = \mathbf{b} \tag{5.18}$$

where $\mathbf{A} = \{a_{ij}\}$ is the system matrix, $\mathbf{u} = \left\{u_i^{t+\Delta t}\right\}$ is the vector of unknowns, and $\mathbf{b} = \{b_i\}$ is the right-hand side (RHS) vector. Full expressions for discrete counterparts of the differential equation (5.4) and boundary conditions (5.5) are given in Appendix B.3.

Figure 5.4 shows all variables included in the discretized equation for internal collocation point *i*. In addition to solution values and spatial and temporal discretization steps, additional information is provided through temporal derivatives in time t, and velocity and diffusion coefficients in all three points of the computational stencil. Because only solution values in time $t+\Delta t$ are sought, all other variables need to be known. Solution values in time t are given as the initial condition or solution from the previous time step. Temporal derivatives in time t also represent initial conditions for the current time step. Because they are not explicitly given, they have to be calculated. Two different cases appear. For the first local time step (i.e., beginning of every global time step), temporal derivatives in time t are obtained from a known solution (initial condition) by calculating the right-hand side of equation (5.4) in time t, i.e., by satisfying the differential equation for every node i at the beginning of the global time step. Spatial derivatives on the RHS are easily calculated using the FCT solution (described in the next section) of the initial condition. For every other local time step, temporal derivatives in time $t+\Delta t$ are obtained from expression (5.17) using the calculated solution and now serve as initial conditions for the next local time step. Temporal derivatives on boundary points needed by the spatial operator for the first and last internal points (i.e., points *2* and *n-1*) are treated in the same manner as internal points.

In the case when the coefficients v and D depend on the solution u, the discretized system of equation (5.18) is nonlinear. This is true for any implicit discretization method so the fact that the developed spatial operator becomes nonlinear itself does not present any particular difficulties. Standard techniques (e.g., the Picard or Newton-Raphson method) can be used

for the linearization of (5.18).

Finally, comparison with finite difference (FD) spatial discretization is discussed. In addition to that it is a commonly used method, FD is chosen because, as stated earlier, identical discretization can be obtained using Fup functions. Differences between required data for the i-th discretized equation are marked in Figure 5.4. It is clear that the developed Fup collocation operator requires more information, but it also enables improved accuracy, as will be presented in the numerical examples. While the expressions for the calculation of matrix coefficients and RHS vector values are not simple as in the FD case (see Appendix B.3), it is important to note that an identical non-zero matrix pattern is obtained, so the memory requirements and cost for solution of the linear system are identical when a direct solver is used.

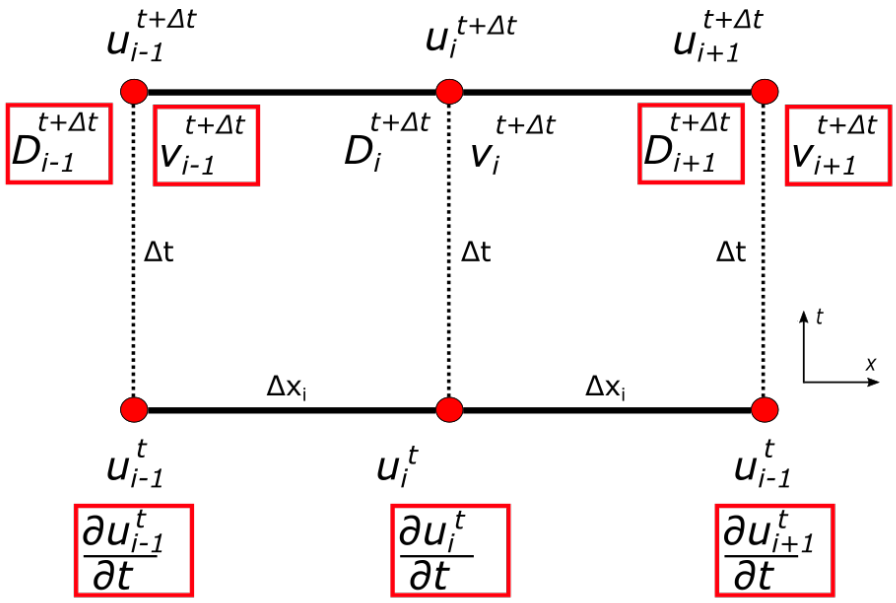


FIGURE 5.4: Space-time computational stencil for collocation point i during one local time step: all values that are included in the discretized equation. Values in frames are not included when the common FD spatial operator is used.

5.2.2 Adaptive strategy in space

The adaptive spatial strategy used in this work is based on multilevel wavelet collocation approach suggested by Vasilyev et al. [88], [89], [105]. This strategy was already applied to the previous versions of AFCM ([11], [24]) that could be regarded as adaptive in the space only. In the sequel, an overview of the spatial adaption strategy is given for completeness of work.

Adaptive Fup collocation transform

This subsection describes the adaptive Fup collocation transform (FCT) that is the main part of the spatial adaptive strategy. FCT is an efficient multiresolution analysis tool for describing various types of functions and signals using the collocation procedure and a linear combination of Fup basis functions. The main idea is to represent (interpolate) the known function in an adaptive multiresolution manner so that coarse grids are used in regions where the solution is smooth, and fine grids are only used in those parts where the solution varies strongly (abruptly).

First, we define the hierarchy of regular dyadic grids corresponding to different resolution levels:

$$G^j = \left\{ x_k^j \in \mathrm{R} : x_k^j = 2^{-j}k, \;\; k \in \mathrm{Z} \right\}, \quad j \in \mathrm{Z} \tag{5.19}$$

where x_k^j is the k-th (location index) collocation point on the j-th resolution level as shown in Figure 5.5. Note that even-numbered collocation point of G^{j+1} already exists on G^j $\left(x_{2k}^{j+1} = x_k^j \right)$, which implies the relation $G^j \subset G^{j+1}$. If we consider the spatial domain $\Omega = [a, b]$, then the characteristic interval at each level is defined as the distance between two adjacent points:

$$\Delta x^j = \frac{b-a}{2^{j_{\min X}+j}} \tag{5.20}$$

where $j_{\min X}$ is the user-defined parameter that defines the resolution at the zero level.

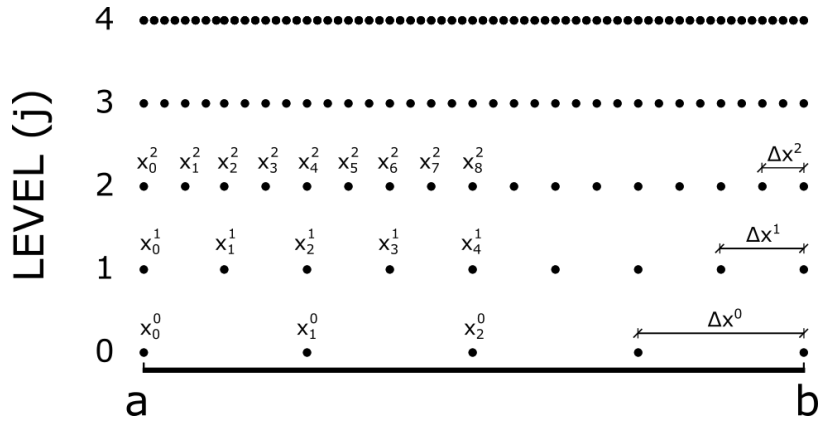


FIGURE 5.5: Example of the hierarchy of regular dyadic grids for j_{minX}=2.

For each collocation point of the dyadic grid (on any resolution level), we can associate the corresponding Fup basis function with its vertex (peak)

located in a particular collocation point and with characteristic interval Δx^j. This means that the trial function space for numerical approximation is obtained by shifting the reference Fup function for $\Delta x^j$and rescaling it to fit the spatial discretization on different resolution levels. Because Fup basis functions are defined with respect to a basic characteristic interval $\Delta x = 2^{-n}$ (see section 2.2), where n is order of Fup_n function, the calculation of Fup basis function values and their derivatives for general characteristic interval is given by:

$$\varphi_k^{j(m)}(x) = \frac{1}{\left(2^n \Delta x^j\right)^{(m)}} Fup_n^{(m)} \left(\frac{x - b_k^j}{2^n \Delta x^j} \right) \tag{5.21}$$

where m is order of the derivative, Fup_n denotes Fup basis function defined with a basic characteristic interval and $b_k^j = a + k\Delta x^j$ is vertex of particular Fup basis function. As described in section 2.2, for each collocation point, there are additional $n/2$ non-zero Fup basis functions on each side. Thus, to have consistent approximation on the boundary, an additional basis functions with vertices outside of the domain are needed. Those functions are regarded as external basis functions, and only their influence inside the domain is considered. This approach with external basis functions is inherited from previous versions of AFCM, and is different from approach used in previous chapters. Thus, on any level, index k is defined within the interval $k \in \left[-n/2, \quad 2^{j_{\min X}+j} + n/2\right]$. Now, the adaptive multi-resolution approximation $u^J(x)$ of function $u(x)$ can be expressed as:

$$u^J(x) = \sum_{j=0}^{J} \sum_{k \in Z^j} d_k^j \varphi_k^j(x) \tag{5.22}$$

where J determines the maximum used resolution level and Z^j is an irregular (adaptive) grid containing only significant collocation points to obtain solution $u^J(x)$ within prescribed error threshold. Unknown Fup coefficients d_k^j and Fup basis functions φ_k^j are associated with the specific resolution level j and location k in the spatial domain. The Fup coefficients are calculated (independently for each resolution level) from the system of equations so that the Fup approximation satisfies the function values at collocation points:

$$\sum_{k \in Z^j} d_k^j \varphi_k^j(x_p^j) = \Delta(x_p^j) \ , \ p \in Z^j : 0 \leq p \leq 2^{j_{\min}+j} \tag{5.23}$$

and boundary derivatives (related to external degrees of freedom – basis functions) at a and b points:

$$\sum_{k \in Z^j} d_k^j \varphi_k^{j(i)}(x_b) = \Delta^{(i)}(x_b) \quad , \quad i = 1, ..., n/2 \quad ; \quad x_b = a, b \tag{5.24}$$

where the residual vector has the following form:

$$\begin{array}{lll} \Delta(x_p^j) & = u(x_p^j) & , \quad p \in Z^j : 0 \leq p \leq 2^{j_{\min}+j} \quad ; \quad j = 0 \\ & u(x_p^j) - u^{j-1}(x_p^j) & , \quad p \in Z^j : 0 \leq p \leq 2^{j_{\min}+j} \quad ; \quad j = 1, ..., J \\ \Delta^{(i)}(x_b) & = u^{(i)}(x_b) & , \quad i = 1, ..., n/2 \; ; \quad x_b = a, b \quad ; \quad j = 0 \\ & 0 & , \quad j = 1, ..., J \end{array} \tag{5.25}$$

Calculation of coefficients and construction of the adaptive irregular grid will be explained in more detail considering the following function:

$$u(x) = -\tanh\left(\frac{x - 2/3}{0.008}\right) \tag{5.26}$$

with $j_{\min X} = 2$ and an error threshold of $\varepsilon = 0.1$, which implies that the residual between the Fup approximation and function (5.26) must be less than this prescribed threshold. Adaptive multiresolution FCT for the chosen function is presented in Figure 5.6.

First, Fup coefficients on the zero level are calculated by satisfying the function values in all $\left(2^{j_{\min X}} + 1\right)$ collocation points of the zero level and the first *n/2* function derivatives in both boundary points by solving the system (5.23)-(5.25) for *j=0*. Then, the absolute difference between the zero level approximation $u^0(x)$ obtained by (5.22) and the true function $u(x)$ is check in all points of the first resolution level. All points on the first level with an absolute difference above the prescribed threshold ε are kept for further calculation. Other points are dropped from the first level. For each retained collocation point, besides the basis function with a vertex in that particular point, there should be an additional *n+2* surrounding basis functions on the first level to have a consistent approximation on both characteristic intervals around the kept point. Those are basically all first-level basis functions that have a non-zero value in the area where the accuracy requirements are not satisfied, and they have to be added to those already kept points. For the first and every next level, the collocation algorithm should only satisfy the *residual* between the true function and the previous level approximation for all kept points on the current resolution level, as indicated by (5.25). This

means that coefficients on previous levels remain unchanged and that only the contribution of higher levels to the final approximate solution is calculated. Note that residuals are always zero for even collocation points because they are already satisfied by the approximation on previous levels. The procedure is repeated until all residuals are less than the prescribed threshold; as shown in Figure 5.6, fine grids with higher resolution levels are obtained only around the front, while in other regions, the adaptive grid uses only lower levels and thus significantly reduces the number of collocation points and computational cost. Note in Figure 5.6 that the FCT algorithm needed up to 6 resolution levels (i.e. *J=6*) to represent given function within prescribed accuracy $\varepsilon = 0.1$. Moreover, due to the C^{∞} property of the Fup basis function, the FCT approximation (as well as all of its derivatives) is expressed as a smooth function in the form of the linear Fup combination (5.22). In this manner, FCT can also be used for efficient calculation of function derivatives just by multiplying calculated coefficients with derivatives of the Fup basis functions (5.21).

Additional points, total and effective grid

The main idea of the adaptive spatial strategy is to employ FCT in the AFCM algorithm to solve initial-boundary value problems (IBVP). At the beginning of every *global* time step, the FCT procedure is used to approximate the known solution (initial condition or solution from the previous time step) with the prescribed accuracy. The result of the FCT approximation is an adaptive grid of collocation points, which is referred to as a basic grid. Apart from basic points (which form the appropriate grid at the beginning of the global time step), additional points are needed for the description of possible solution changes during the global time step. The basic hypothesis behind the algorithm is that the solution does not "move" outside the border of the adaptive non-uniform grid.

Usually, this can be fulfilled by incorporating some type of prediction procedure that increases the overall cost of the algorithm [106]. In this work, we do not focus on a sophisticated procedure to control grid reliability; rather, a heuristic approach similar to Harten's original strategy [84] is used.

First, to take into account solution movement (e.g., advection of the solution) during the global time step, additional neighboring points at the same level for each existing basic point are added. Here, parameters N_L and N_R are defined as the number of additional points to the left and to the right, respectively. Note that these points already exist everywhere except on level

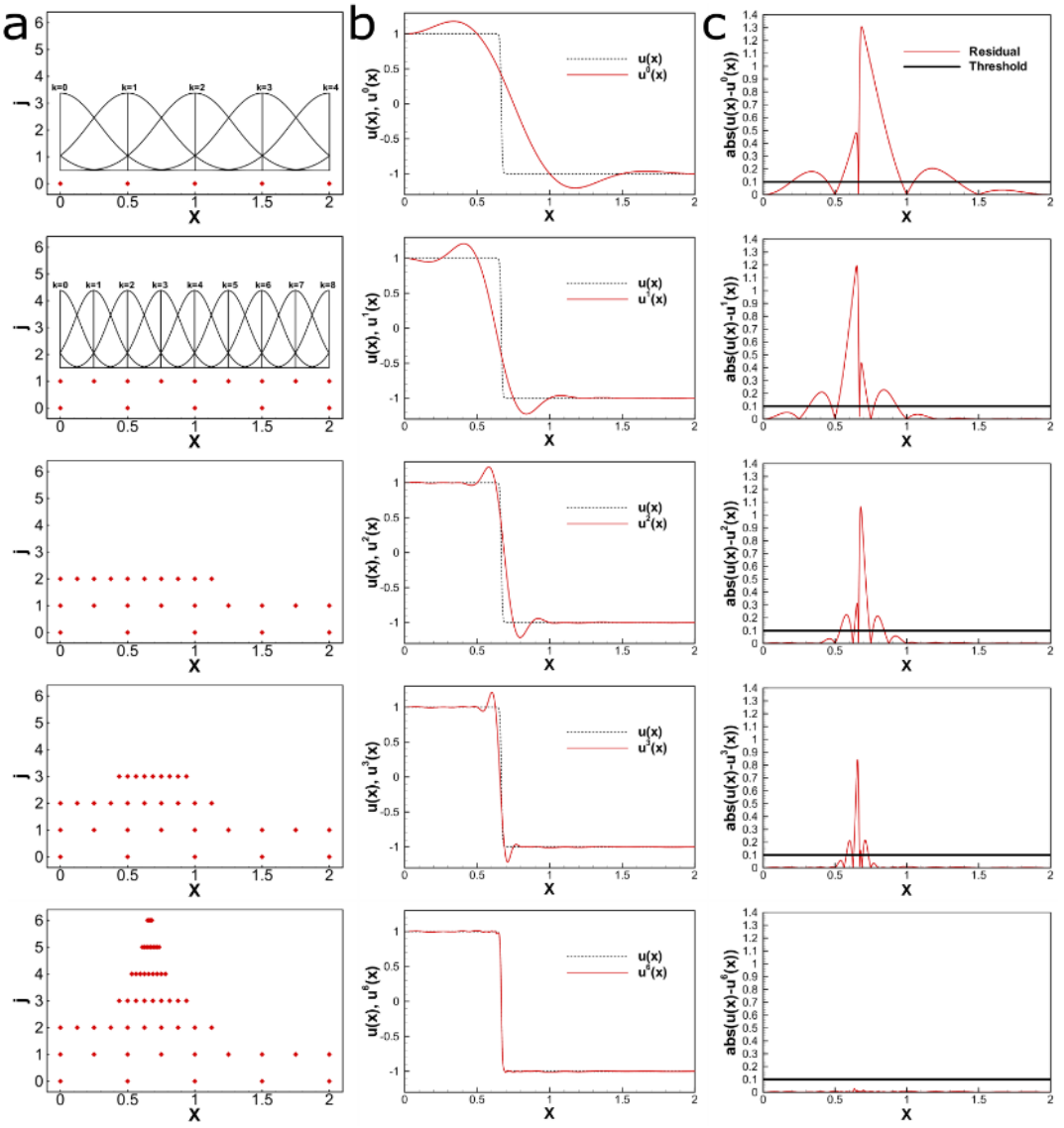


FIGURE 5.6: FCT approximation of the function (5.26): a) adaptive grid with the corresponding Fup basis function on the zero and first levels, b) given function and its FCT approximation and c) absolute difference and error threshold.

transitions. Second, to include the possibility that the solution can become more demanding (e.g., the formation of sharp gradients), additional points for a specific number (M) of higher resolution levels are defined. Basically, for every basic point on the highest level, the same collocation point on M higher levels is created with all of their corresponding neighbors to have a consistent approximation.

Basic and additional points create the total grid, which is the appropriate grid for solution description during the entire global time step. In practice,

inclusion of these additional points is incorporated during the FCT procedure (in a level-by-level manner), which means that additional points, together with basic points, create a set of basis functions for which Fup coefficients have to be calculated on every resolution level. To avoid any confusion, we here emphasize that, during the FCT procedure, our unknowns are Fup coefficients. As presented in the previous section, while solving IBVP, unknowns are solution values in collocation points rather than Fup coefficients. In that case, the total grid is not the appropriate grid for spatial discretization of the PDE due to the repetition of some collocation points over different levels, as it would produce linearly dependent equations (basically, multiple identical equations for the same collocation point on different levels). For that reason, the effective grid is created in the following manner. At the highest resolution level, all points belong to the effective grid. At any lower level, only points that do not exist at higher levels are kept. In that manner, the effective grid has a unique collocation point for a specific location and can be used for spatial discretization of the PDE. The effective grid corresponding to the total grid of FCT example (Figure 5.6) is shown in Figure 5.7.

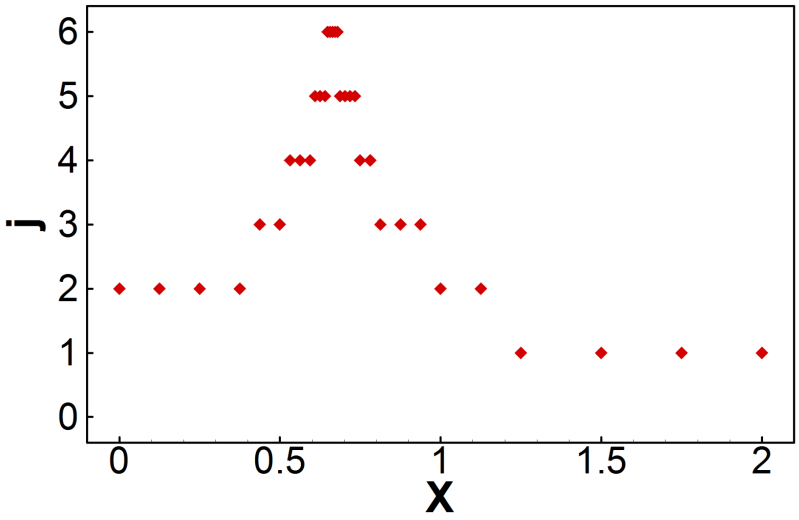


FIGURE 5.7: Example of the effective adaptive grid.

After the solution of IBVP at the end of the global time step is obtained, the FCT algorithm is used once again, this time to transform known solution values (discrete values in collocation points of the effective grid) into the corresponding Fup representation by calculating new Fup coefficients (in $t + \Delta t_{global}$) by applying (5.23)-(5.25) on the *total* grid defined at the beginning

of the global time step. Furthermore, the solution is expressed as a continuous and smooth function, which now represents the new initial condition for next global time step. Applying the FCT procedure again on this new initial condition for the next global time step produces the adaptive grid, which, if the solution has changed, differs from the previous one. In this manner, the adaptive spatial strategy dynamically changes the spatial grid over time and significantly reduces the overall computational cost.

5.2.3 Adaptive strategy in time

In the literature, there are two approaches that can be regarded as adaptive in time. The first one (e.g. [107]–[109]), called adaptive time stepping (or adaptive step size), defines the time step as a function of time, i.e., $\Delta t = \Delta t(t)$. This means that the time step can be changed from step to step, but it is the same for all points in the domain. This approach can be easily incorporated in a developed methodology for adaptive control of the *global* time step.

In the second approach, which is more related to temporal adaptive strategy developed in this work, the time step is a function of space: $\Delta t = \Delta t(x)$, i.e., different parts of the spatial domain have different time steps. This type of adaptation is usually defined in the literature as local time stepping or multi-rate time stepping (e.g., [94], [97]). In this context, parts of the domain, as well as corresponding time steps, have to be determined in advance. For the adaptive mesh refinement (AMR) techniques, this is usually done by relating spatial and temporal discretization steps. This means that each spatial resolution level has corresponding local time step. Therefore, time steps are based on a CFL condition that needs to be satisfied only locally for every spatial resolution level. Most works apply the methodology for an explicit method (e.g. [97]–[101]), but there are also works based on implicit discretization [96]. The described approach requires proper treatment of the interface between parts of the domain with different temporal discretizations.

In this work, emphasis is placed on development of a temporal adaptive strategy that is independent of the spatial resolution level. The idea is that the algorithm uses smaller local time steps only in those parts of the domain where accuracy requirements are not fulfilled. To achieve this goal, the following steps are proposed:

1. For all points of the effective spatial grid, time-marching is performed from $t + \Delta t_{global}$ with the chosen number of $2^{j_{\min T}}$ local time steps,

where j_{minT} is the user defined number that determines the local time steps size ($\Delta t_0 = 2^{-j_{\min T}} \cdot \Delta t_{global}$) on the zero time level.

2. The procedure is repeated from $t + \Delta t_{global}$ using the double number of local time steps at the first time level. The time step for any level is determined by $\Delta t_{j_T} = 2^{-(j_{\min T} + j_T)} \cdot \Delta t_{global}$, where j_T denotes the current time level.

3. For every temporal line (one temporal line is assigned to each spatial collocation point, as already stated), two solutions (the zero and first time levels) are compared in all temporal points (discrete time moments) of the lower time level. The absolute difference in all temporal points on the corresponding line is compared with the prescribed threshold for the temporal error.

4. For temporal lines where the maximum computed error is greater than the threshold, time-marching should be repeated on the next time level and a two-times-smaller local time step. Other lines where accuracy is achieved are removed from further calculation, and they are referred to as inactive lines. On the next time level, some temporal points on inactive lines will be missing for the calculation of spatial derivative on active lines. These values have to be interpolated from the previous time level solution. This is done by performing FCT on these inactive lines by satisfying the known solution and boundary derivatives (as described in section 5.2.2). Interpolated values are then incorporated in the system of equations for active lines as Dirichlet boundary conditions.

5. The procedure is repeated until the prescribed accuracy is not achieved at all temporal lines or the maximum user-defined time level $J_{\max T}$ is not reached.

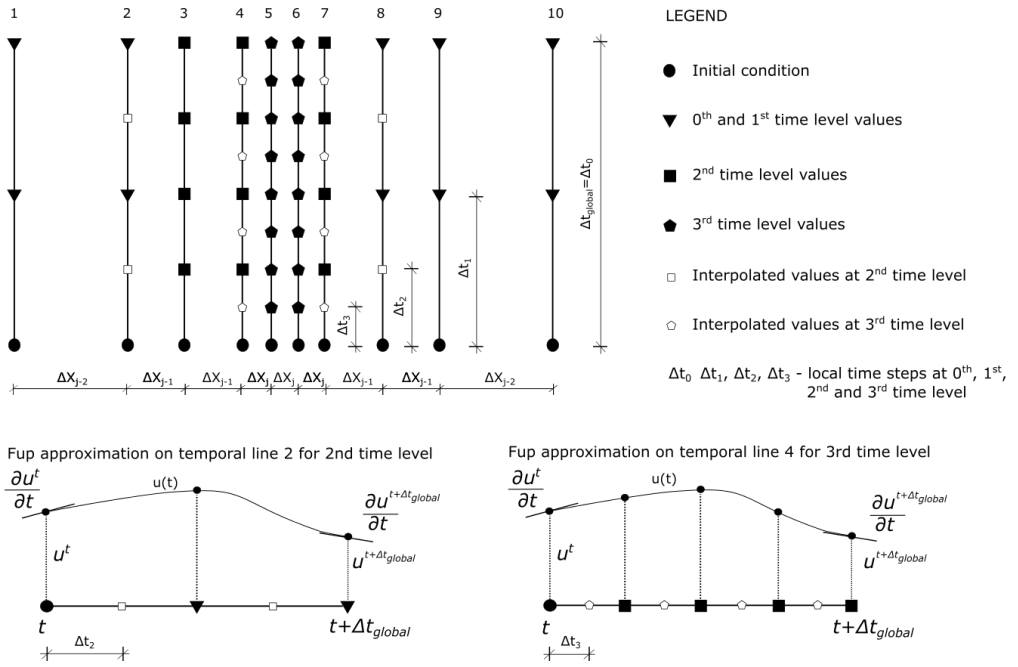


FIGURE 5.8: Temporal adaptive strategy.

This procedure is adaptive in time because different collocation points use different local time steps according to the accuracy criteria. Moreover, Figure 5.8 presents the described adaptive temporal procedure. For instance, the zero level includes one-time step for all points, i.e., $j_{\min T} = 0$. Next, the solution is obtained using two local time steps (black triangle points). Accuracy is not obtained at lines 3-7, which use four local time steps in the next second time level. Lines 3 and 7 need Fup interpolation in lines 2 and 8 (white squares) from the previous first time level. Interpolated values are calculated by FCT on the corresponding temporal line and represent Dirichlet boundary conditions for the second time level. Only lines 5 and 6 need to be resolved in the third time level with eight local time steps. Interpolated values are calculated in lines 4 and 7 (white pentagons). The procedure can be further repeated until accuracy is not achieved in all temporal lines and all temporal collocation points on them.

In this manner, the adaptive procedure resolves different time scales on different temporal lines independently of the spatial resolution. The consequence of this independence is apparent if one considers a problem with slow temporal changes (or even reached a steady-state solution). Still, solution in the space domain can be very demanding, resulting in a highly adaptive spatial grid. Contrary to common local time stepping methods, which would use highly different time steps based on the adaptive spatial grid, the developed strategy would resolve all temporal scales within the same time step for all

spatial locations.

Moreover, while common local time stepping procedures are mostly based on an explicit method because of the additional complexity of implicit methods, the developed strategy is the same for explicit and implicit methods. The procedure is not restricted to a single temporal discretization, so it can be used for other methods, such as Runge-Kutta or Adams–Moulton.

Further, accurate calculation of interpolated values on inactive lines is crucial for accurate simulation. FCT enables continuous (and smooth) interpolation along every temporal line, and it is an important tool for the accuracy and robustness of the temporal adaptive strategy. The resolution level on corresponding temporal line is determined by the time level of the adaptive strategy, which directly determines the accuracy of interpolation. If one, for example, tries to replace the FCT algorithm and use some *local* polynomial approximation, such as the one derived during the construction of the temporal collocation operator (see Figure 5.4 and expressions (5.12) and (B.6)), the results of interpolation are usually poor and suffer from oscillatory behavior.

In addition to the FCT algorithm, the parameter j_{minT} has an important role. Low values of this parameter can lead to reduced accuracy when compared to the reference non-adaptive solution. This is mostly because of loose adaptive criteria and reduced interpolation accuracy on the first time levels. Every higher level possibility of this type of error is smaller, but error created on lower levels can contaminate the overall solution. Numerical experiments show that in most cases, for $j_{minT} = 2$, the solution remains within the same accuracy as the reference non-adaptive solution while significantly reducing the overall computational cost.

The described procedure represents an efficient way to solve problems with highly different solution time scales, as will be demonstrated via numerical examples.

5.2.4 Implementation details and additional possibilities

This section discusses some implementation details and additional possibilities of the described methodology that were previously omitted for clarity of the presentation.

First, some details regarding the construction of the computational stencil for the *i*-th point of the effective grid are discussed. For most points, the situation is clear because two neighboring points (*i-1* and *i+1*) can be found

at the same resolution level. For collocation points that are located in the transition zones between different resolution levels, some points can be missing from the collocation stencil at the corresponding level. There are a few possible solutions (Figure 5.9): a) a missing point can be interpolated from existing points, b) three uniformly spaced points on different levels can be selected for the construction of a non-compact stencil, c) in 1D, it is possible to a construct collocation operator on non-uniform grid of three closest neighboring points. In this work, all three options were tested. Differences in accuracy were negligible; however, different procedures lead to different non-zero matrix patterns near level transitions. Only the last option guarantees a tridiagonal matrix. Accordingly, the collocation operator has to be modified; for the first two options developed operator from section 5.2.1 is still valid, but option: a) needs the interpolation of all values for a missing point in the collocation operator, b) needs the modification of indexes *i-1* and *i+1* to denote the correct global indexes of collocation points from the computational stencil, and option c) needs a particular construction of the irregular collocation operator. The irregular operator converges toward the previously developed operator as the spacing converges toward a uniform grid. Finally, option b) is used for all results presented in this work.

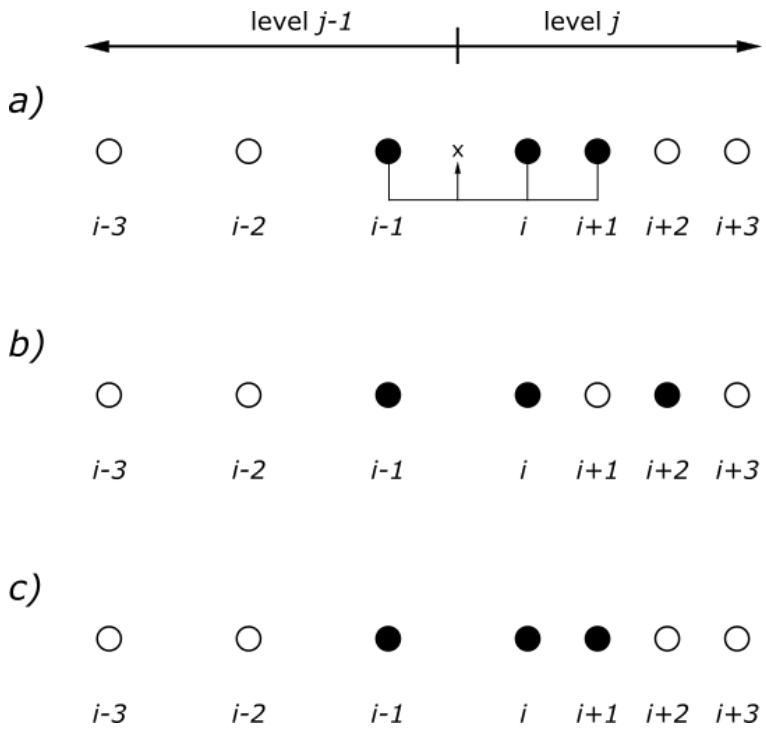


FIGURE 5.9: Possible solutions for the construction of the spatial computational stencil for node "i" near a level transition.

Next, a few comments about the developed spatial collocation operator

are given. Generally, the spatial operator is independent of the used time discretization. This means that the same developed spatial operator can be used with different time discretization methods (explicit or implicit, single-step or multi-step methods, etc.). Moreover, construction of this type of operator is not limited to the time-dependent problem and can be constructed for stationary problems. For a stationary advection-diffusion problem, the collocation operator and final discretized equation are obtained by simply setting the temporal derivatives to zero in (B.3)-(B.4) and (B.5), respectively. Additionally, the constructed operator can be used for a pure diffusion (or advection) equation by setting the coefficient v (or D) to zero.

Development of this type of operator is not limited to Fup_2 functions. The same five conditions for development of the spatial collocation operator could be fulfilled with five cubic B-spline functions or a single algebraic polynomial of the 4^{th} degree. Of course, different functions should produce different operators, so it would be interesting to compare them. Additionally, higher-order functions could be employed, but the outcome would be a larger computational stencil and increased overall cost.

A temporal adaptive strategy is developed by using only calculated solution values and interpolating data along (one-dimensional) temporal lines from previous time levels. It is independent of spatial and temporal discretization. This potentially means that it could be easily incorporated within different numerical methods.

The first 2D example of space-time adaptive algorithm is presented in this work (section 5.3.4). Since each collocation point is related with single temporal line (which is always one-dimensional in time direction) the transition of adaptive temporal strategy to higher dimensions is straightforward. Regarding to adaptive strategy in space, generally the same structure as in 1D is followed. However, some parts require more detailed implementation for higher dimensions. For this we refer to previous work [11]. However, the construction of Fup collocation operator for higher dimensions is not straightforward. Following the 1D work, for a full Fup_2 multidimensional local approximation on a 2D subdomain (defined by 9 collocation points), there are 25 non-zero basis functions. This increases the complexity for the development of the collocation operator but opens an opportunity to test different options and possibly provide an even higher gain than in 1D case. Since the main contribution in this chapter is full space-time adaptive algorithm, the construction of multidimensional collocation operator is left for further investigation. Thus, for 2D example in this chapter, the spatial derivatives are

approximated with finite difference operator.

Certainly, all these additionally possibilities should be implemented and tested before claiming that it will work. They are only mentioned here based on the authors' impressions.

5.3 Numerical examples

In this section, numerical examples to illustrate the efficiency and robustness of the AFCM (Adaptive Fup Collocation Method) for modeling advection-dominated problems are presented. The first example is the moving steep front arising from the linear advection-diffusion equation that has the exact solution obtained by Ogata and Banks [110]. This example shows the ability of AFCM to follow the movement of the sharp front inside the domain. The second example describes the solution of Burgers equation, which also has an exact solution. This example shows the ability of AFCM to capture the evolution of the sharp front. The third example presents the moving front arising from the unsaturated groundwater flow described by the Richards equation. This example confirms the ability of AFCM to solve strongly nonlinear problems while keeping all of the desired properties. The last example is linear 2D example that demonstrates validity and efficiency of AFCM for multidimensional problems. All examples present the ability of AFCM to efficiently and accurately describe all spatial and temporal scales. The following is a description of the control AFCM parameters:

$j_{\min X}$ − minimum spatial resolution at zero level for FCT in x − direction
$j_{\min Y}$ − minimum spatial resolution at zero level for FCT in y − direction
$J_{\max S}$ − maximum spatial resolution level for FCT
N_L − number of additional points to left for FCT
N_R − number of additional points to right for FCT
N_B − number of additional points to bottom for FCT
N_T − number of additional points to top for FCT
M − number of additional levels for FCT
ε_S − spatial error threshold
$j_{\min T}$ − minimum temporal resolution at zero level for adaptive time procedure
$J_{\max T}$ − maximum temporal resolution level for adaptive time procedure
ε_T − temporal error threshold
Δt_{global} − global time step

5.3.1 1D linear advection-diffusion equation

The advection-diffusion equation describes transport problems driven by advection due to velocity and diffusion (or dispersion) due to spreading physical properties of the injected plume in the related medium:

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} - v\frac{\partial u}{\partial x} \tag{5.27}$$

where D and v are constant with initial and boundary conditions:

$$u(x,0) = 0 \tag{5.28}$$

$$u(0,t) = 1 \tag{5.29}$$

$$\frac{\partial u(L,t)}{\partial x} = 0 \tag{5.30}$$

The physical interpretation of the main variable in this example is concentration, while domain, velocity and diffusion are given in the dimensionless form:

$$L = 1.2, \quad ; \quad D = 5 \cdot 10^{-7} \quad ; \quad v = 10^{-3} \tag{5.31}$$

AFCM parameters are the following:

$$\begin{array}{ll} j_{\min X} = 2, & J_{\max S} = 12 \\ \varepsilon_S = 10^{-5}, & M = 2 \\ N_L = 1, & N_R = 20 \\ j_{\min T} = 1, & J_{\max T} = 10 \\ \varepsilon_T = 10^{-6}, & \Delta t_{global} = 10.0 \end{array} \tag{5.32}$$

As we consider a 1D advection-diffusion problem with constant velocity and diffusion, the solution is given by an analytic expression [41]:

$$u\,(x,t) = \frac{u_0}{2}\left[erfc\ (v) + \exp\left(\frac{v\,x}{D}\right)\ erfc\left(\frac{x + v\,t}{2\sqrt{D\,t}}\right)\right] \tag{5.33}$$

Figure 5.10 presents the evolution of the solution and corresponding adaptive spatial grid at four different time moments. The results show that, in the case of advection-dominated transport problem, the adaptive grid accurately follows the moving front and controls the numerical oscillations. At the same time, the temporal adaptive strategy resolves all temporal scales (local time steps) during each global time step. Figure 5.11 shows the number of collocation points as well as the minimum local time step during calculation. The

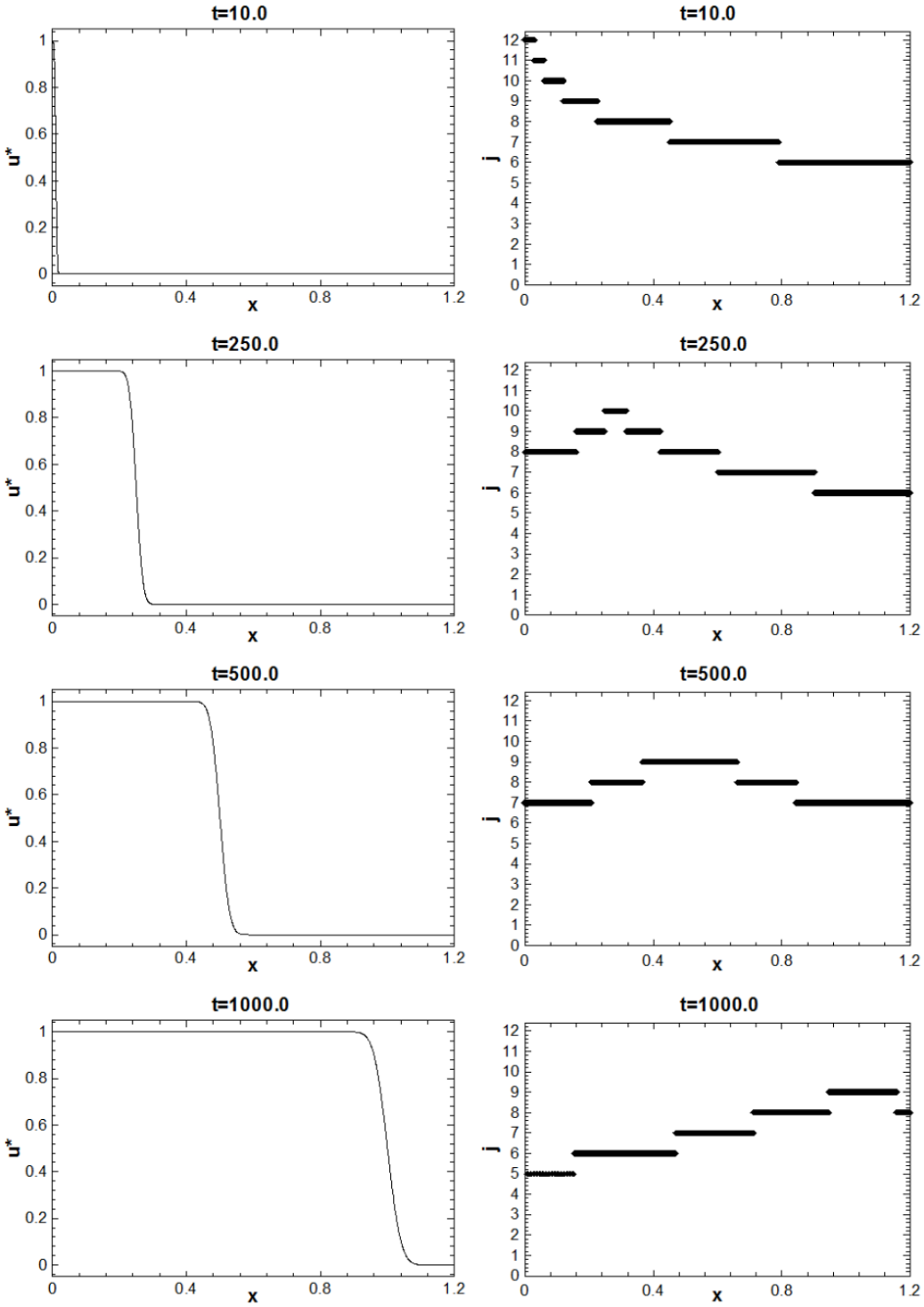


FIGURE 5.10: Numerical solution (left) and effective spatial grid (right) at different time steps.

number of collocation points is fixed inside a single global time step, while local time steps vary from point to point. Here is shown the smallest local time step used inside each global time step. At the beginning of calculation,

the solution is most demanding so AFCM uses the largest number of collocation points and smallest local time step. Over time, the diffusion acts on the solution and decreases its sharpness such that the number of collocations point decreases, while larger local time steps are able to satisfy same accuracy requirements (the temporal error threshold).

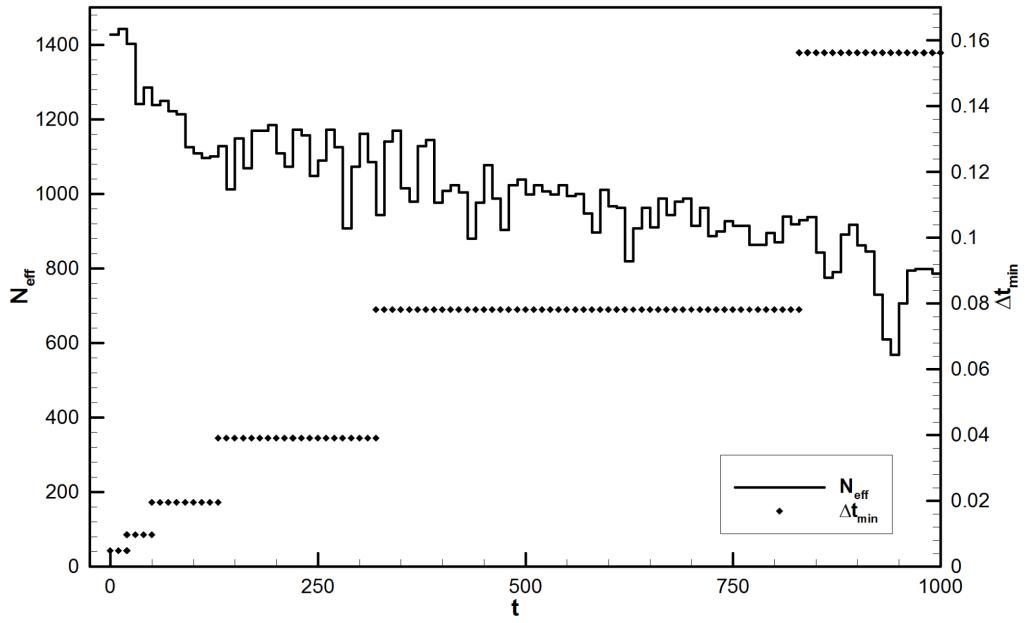


FIGURE 5.11: Number of collocation points and minimum local time step (each value corresponds to a single global time step).

Figure 5.12 presents an adaptive space-time collocation grid where each symbol marks the location and time where a discrete solution is calculated. Here, is presented only portion of the total space-time grid because the high accuracy defined by the error thresholds results in a very fine space-time grid. There are three global time steps shown in the figure and only part of the spatial domain where the solution is most demanding. After each global time step $\Delta t_{global} = 10.0$, the FCT algorithm finds a new adaptive spatial collocation grid. Then, the adaptive temporal strategy uses different local time steps for different collocation points based on temporal error estimation, as explained in section 5.2.3. Figure 5.12 also shows that only collocation points around the moving front require smaller local time steps. In this example, the finer spatial resolutions roughly follow the finer time resolutions and vice versa, but the present form of the AFCM finds all temporal scales independently of spatial resolution levels.

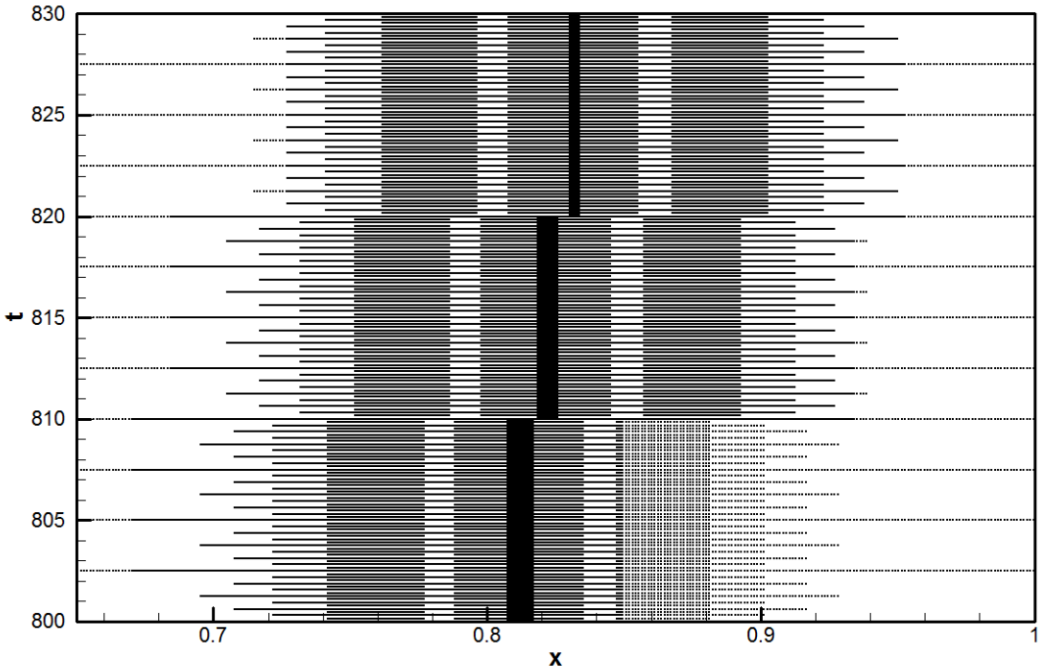


FIGURE 5.12: Portion of the space-time grid used by AFCM. Each symbol corresponds to the point where a discrete solution is calculated.

Next, Figure 5.13 presents the accuracy of the AFCM in terms of the L_∞ norm (absolute difference between exact and numerical solutions) over time. The first time step introduces an initial error greater than the threshold due to the discontinuous solution at t=0. Then, the error decreases, but after a few time steps, the error continuously increases due to the systematic error accumulation over time. The accuracy differences between the common finite difference operator (second-order CDS) and the new applied collocation operator is illustrated. The collocation operator usually gives more accurate solutions (by an order of magnitude) than the classical finite difference operator. Although the Fup basis functions belong to the polynomial vector spaces, the constructed Fup collocation operator involves two additional pieces of information (in the form of the differential equation at boundary stencil points) that significantly improve the solution, adding physical constraints in the operator and decreasing the accumulation of time integration errors. The results confirm that the actual numerical error is one order higher than the threshold due to the discontinuous initial conditions and systematic accumulation of errors over time.

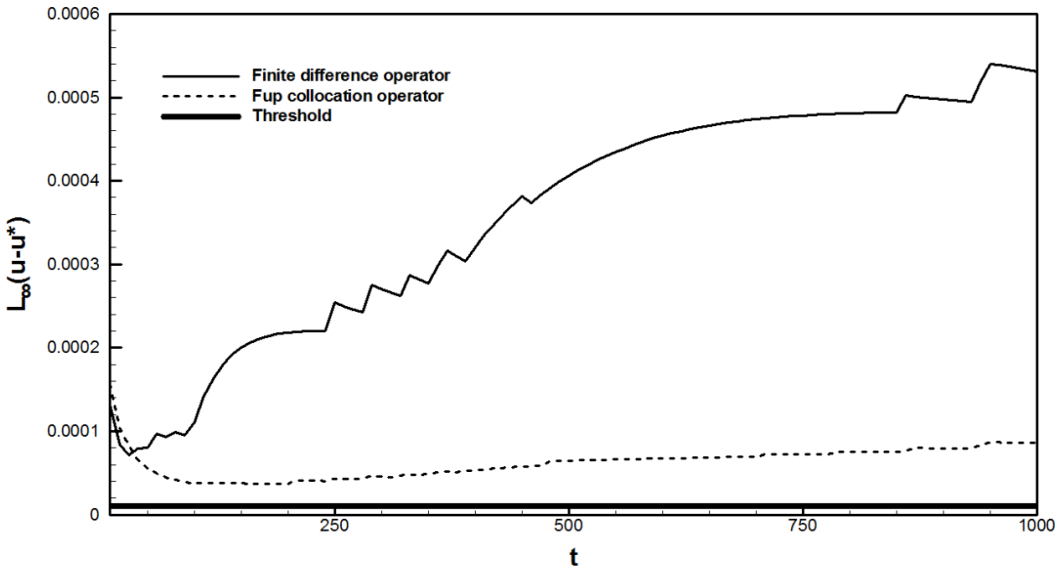


FIGURE 5.13: Comparison of the maximum absolute error between exact and numerical solution for two different spatial operators.

Moreover, to demonstrate the Fup adaptive strategy in time, a particular calculation is performed (Figure 5.14). The calculation is performed from the exact initial condition of t=400 to t=500 using only one global time step. The zero time level uses all temporal lines and only one local time step defined for all spatial collocation points (i.e., jminT=0). The solution contains oscillations around the front. The first level is also full, but all other time levels require fewer temporal lines. Every level has two-times-smaller local time steps than the previous one. Each temporal line (spatial collocation point) is removed from the calculation procedure if the solution satisfies the adaptive criterion that the difference between two levels is less than the prescribed threshold. Higher time levels require only temporal lines around the front using the smaller local time steps and reducing numerical oscillations. Note that even used spatial operator is a symmetric (central) operator in space, which is not typical for the approximation of the advection term, stability is still maintained by the developed space-time adaptive mechanism. Finally, Figure 5.15 presents the total space-time adaptive grid for the solution in Figure 5.14.

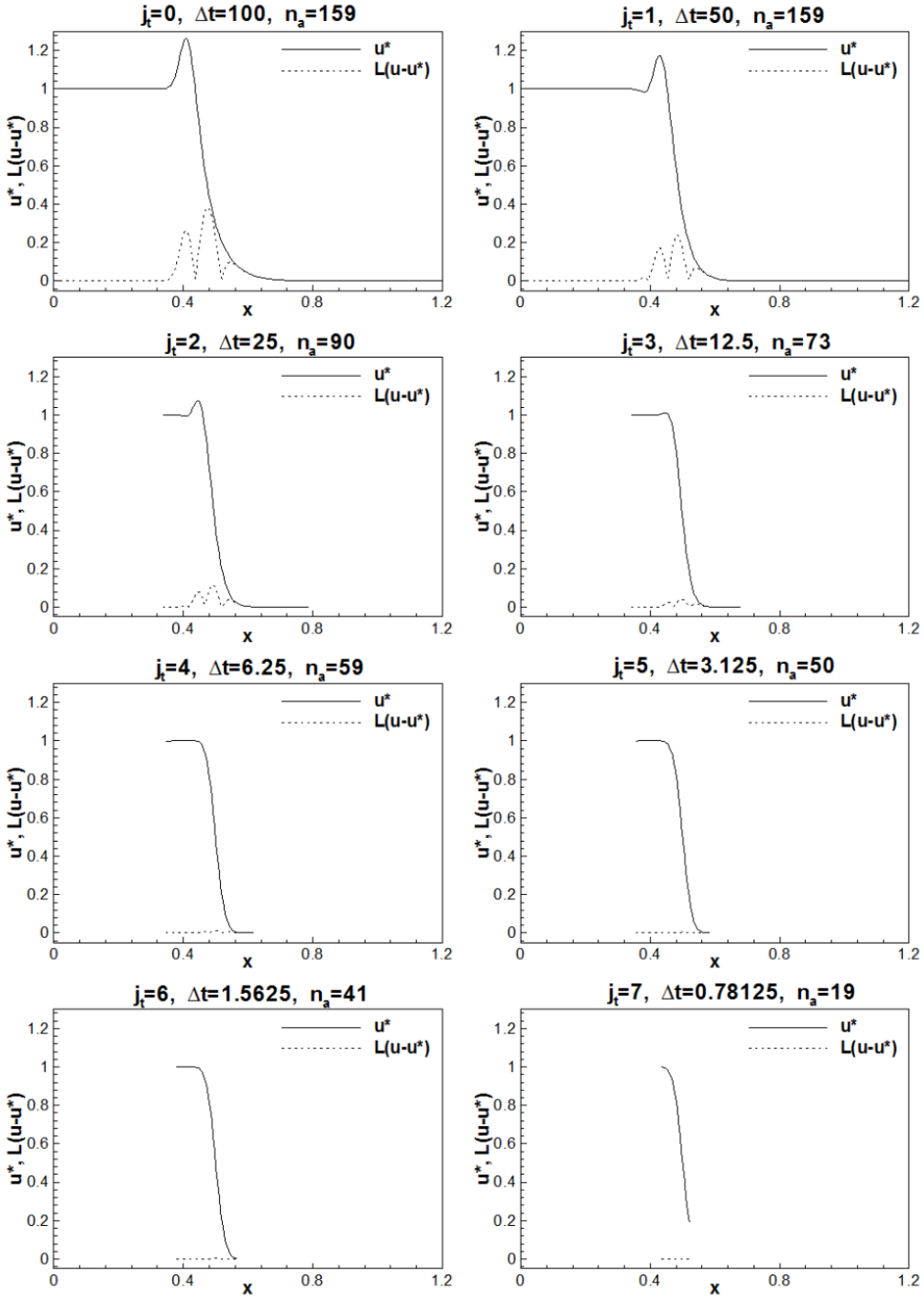


FIGURE 5.14: AFCM solution at $t = 500$ for a single $\Delta t_{global=100}$ and for all time levels; n_a here represents the number of active temporal lines (collocation points); the solution is shown in the space domain.

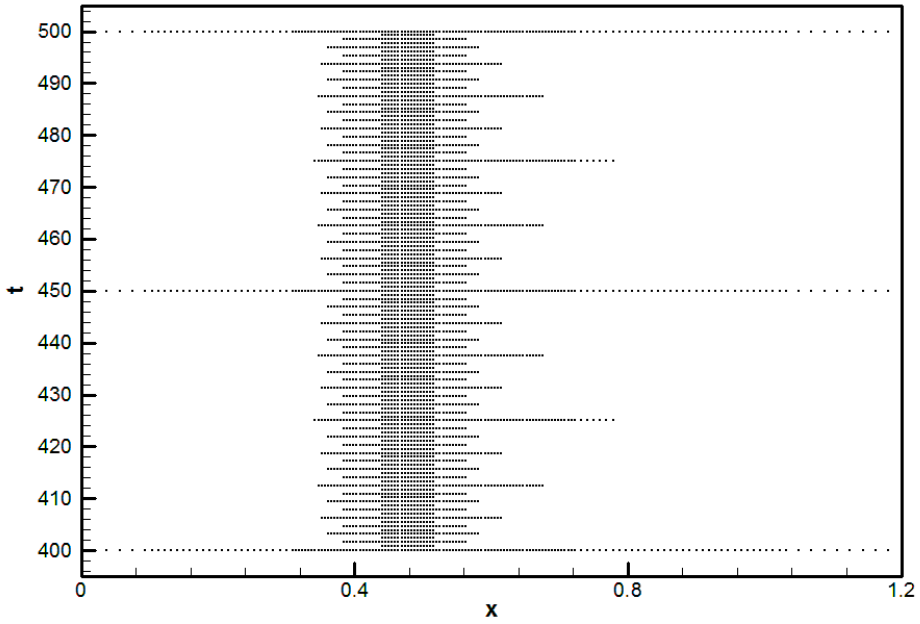


FIGURE 5.15: Space-time adaptive collocation grid for the solution in Figure 5.14

5.3.2 1D Burgers equation

The Burgers equation results from the application of the Navier–Stokes equations to unidirectional flow without a pressure gradient. It is a classical nonlinear benchmark example due to the existence of the analytic solution included to show more general properties of AFCM. The Burgers equation is given by:

$$\frac{\partial u}{\partial t} = v\frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial x} \tag{5.34}$$

with the initial and boundary conditions:

$$u(x,0) = -\sin(\pi x) \tag{5.35}$$

$$u(\pm 1, t) = 0 \tag{5.36}$$

where the velocity is dimensionless and the viscosity is given by:

$$x \in [-1,\, 1]\,,\;\; t \in [0,\, 1.5/\pi]\;;\quad v = 10^{-2}\Big/\pi \tag{5.37}$$

The AFCM parameters are the following:

$$\begin{array}{ll} j_{\min X} = 1, & J_{\max S} = 14 \\ \varepsilon_S = 10^{-2}, & M = 2 \\ N_L = 1, & N_R = 1 \\ j_{\min T} = 0, & J_{\max T} = 12 \\ \varepsilon_T = 10^{-4}, & \Delta t_{global} = 1.5/(6\pi) \end{array} \tag{5.38}$$

The analytic solution is given by the following expression (see Vasilyev and Bowman [89]):

$$u(x,t) = -\frac{\int_{-\infty}^{\infty} \sin(\pi(x-\eta)) \exp\left(-\frac{\cos(\pi(x-\eta))}{2\pi v}\right) \exp\left(-\frac{\eta^2}{4vt}\right) d\eta}{\int_{-\infty}^{\infty} \exp\left(-\frac{\cos(\pi(x-\eta))}{2\pi v}\right) \exp\left(-\frac{\eta^2}{4vt}\right) d\eta} \tag{5.39}$$

Gotovac *et al.* [11] presented the solution of this benchmark example using a spatial adaptation procedure, but with classical non-adaptive time-marching schemes and a finite difference operator. In this work, the solution with a space- and time-adaptive procedure and novel collocation operator for the approximation of spatial derivatives is presented. Figure 5.16 shows the evolution of the solution and corresponding adaptive spatial grid, while Figures 5.17 and 5.18 present the space adaptive grid (previous version of AFCM) and full space-time adaptive grid (presented version with novel temporal strategy). This example shows the ability of AFCM to address the developing front inside the domain with a clear independent resolution of all spatial and temporal scales.

Similar results were presented by simultaneous space-time adaptive collocation algorithms [111], [112]. However, that approach regards the time direction as additional dimensionality, and the algorithm becomes very expensive, especially for 3D problems. The presented approach separates space and time approximation and more cheaply resolves all spatial and temporal scales.

In addition, error analysis with comparison between the developed Fup collocation operator and the FD operator is demonstrated. Independent analysis can be carried out to show that both spatial and temporal discretizations used in this work are second-order accurate. Contrary to the Galerkin approach where one could expect third-order accuracy of Fup_2 functions, the collocation approach usually has reduced convergence order (see Figure 3.7 in section 3.4). This is also in accordance with finite difference method, where

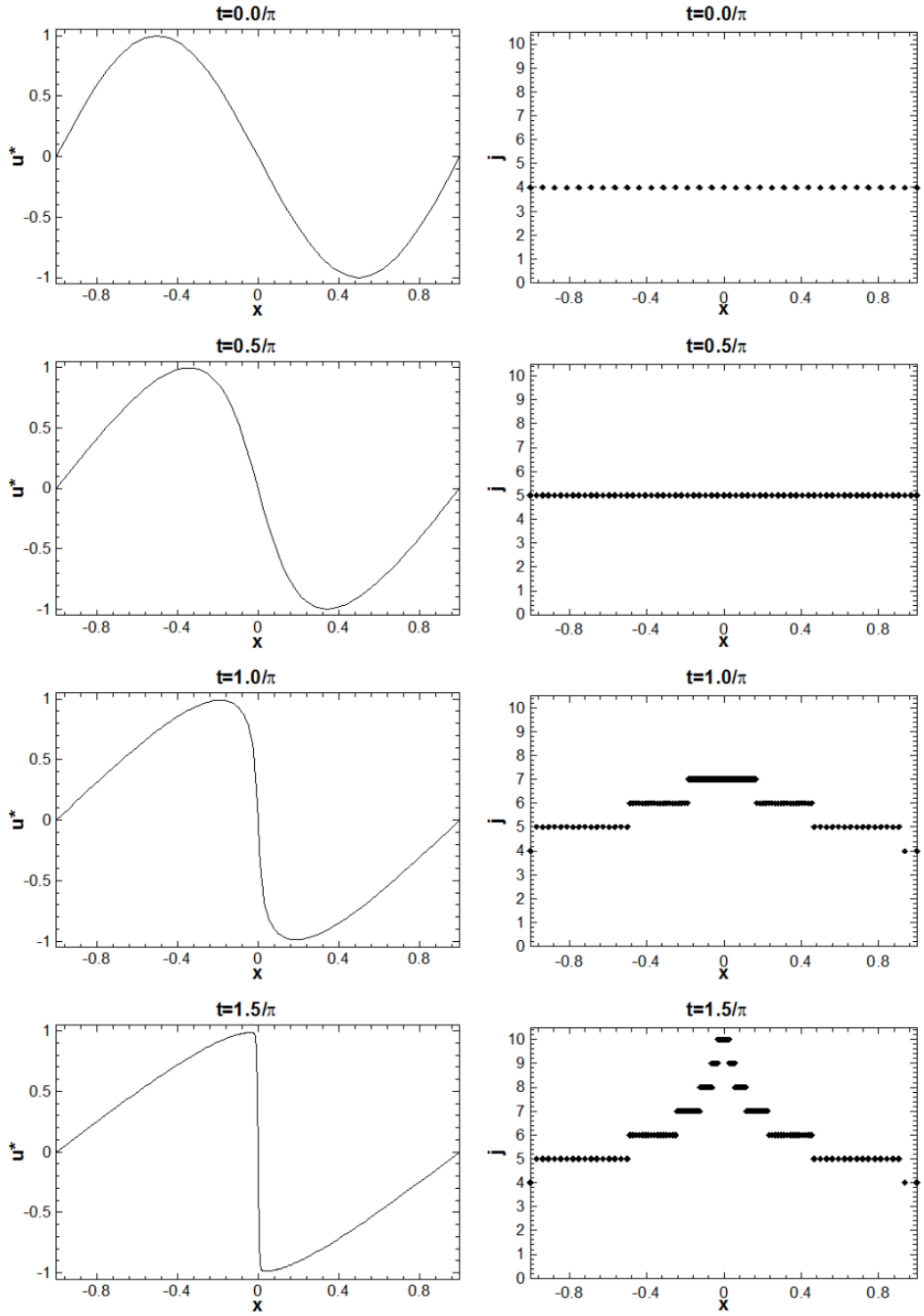


FIGURE 5.16: Numerical solution (left) and effective spatial grid (right) at different time steps.

fitting parabola to the three data points produces second-order central difference operator. Moreover, Fup_2 temporal discretization is identical to the Crank-Nicolson scheme, which is well known to be globally second-order accurate. To expect the full convergence rate of the method, a proper relation

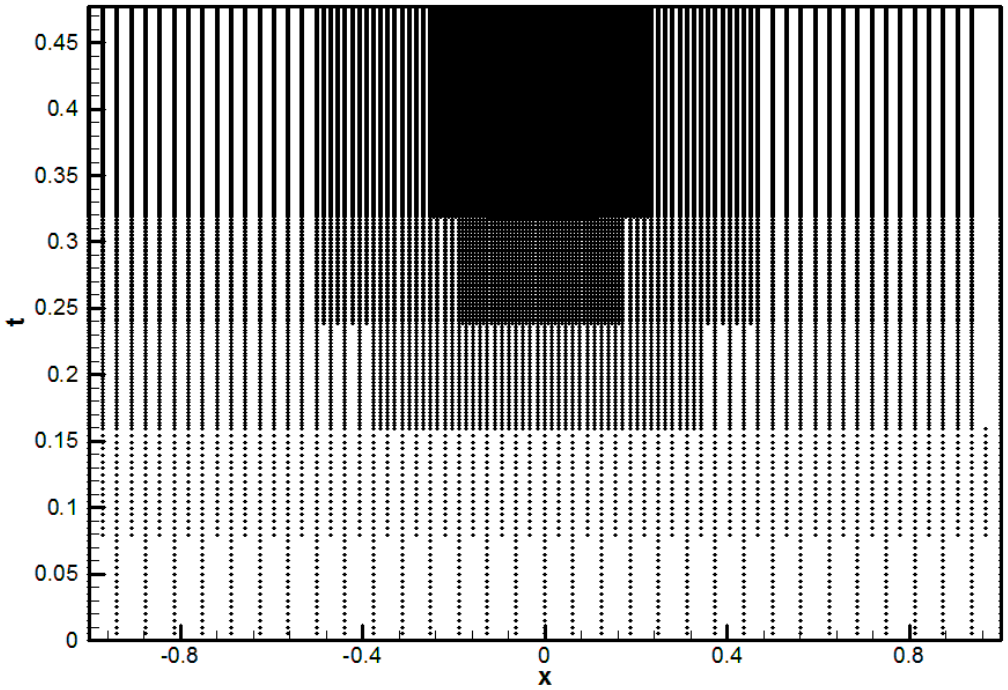


FIGURE 5.17: Space – time grid for space adaptation.

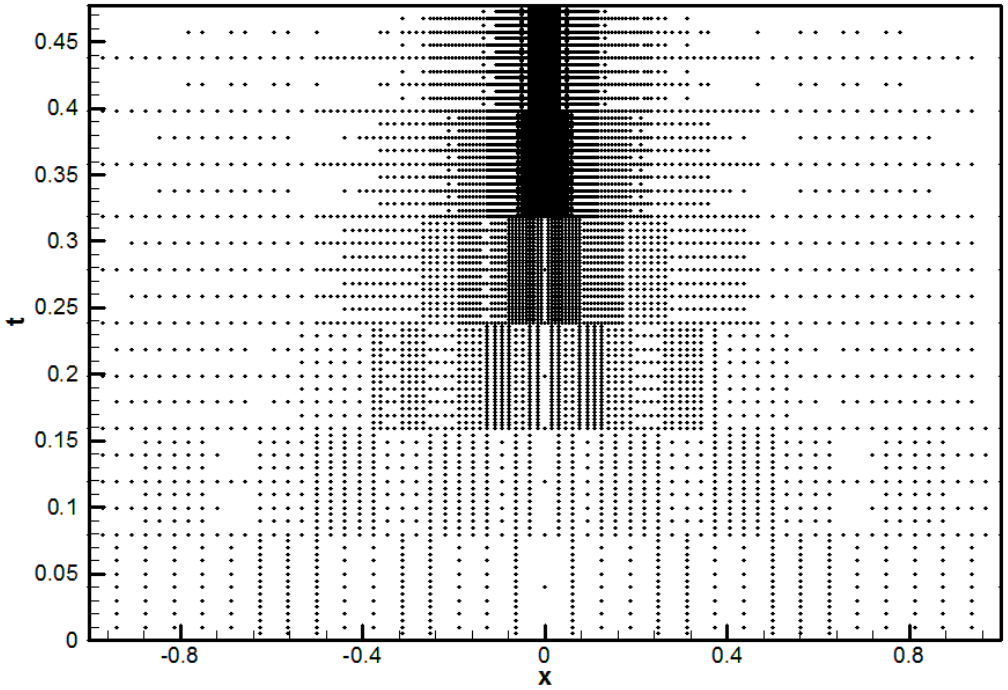


FIGURE 5.18: Space – time grid for space and time adaptation.

between the time step and grid spacing should be established. In presented methodology, this relation is the outcome of the space-time adaptive procedure and generally varies in both the space and time directions. Therefore,

classical convergence analysis is not straightforward in the case of developed space-time adaptive procedure.

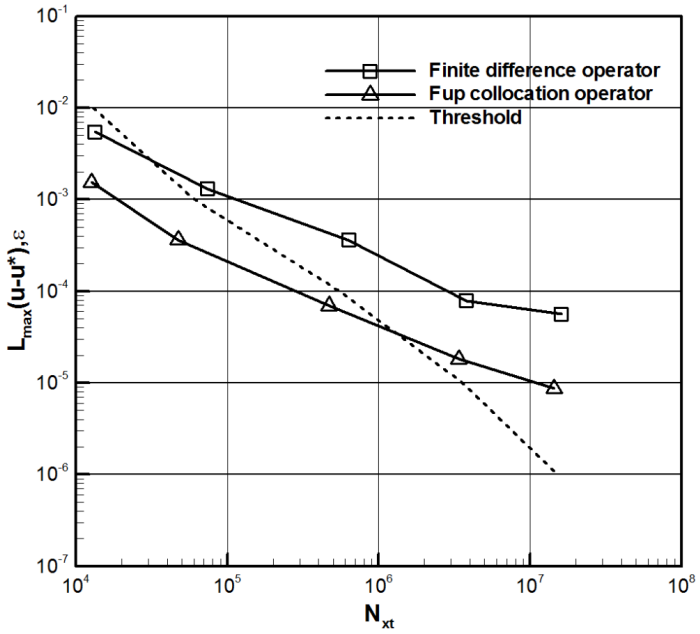


FIGURE 5.19: AFCM accuracy analysis: absolute error (solid lines) for two types of spatial operators and error threshold (dashed line) as a function of the total number of collocation points in the space-time domain.

The main idea of the algorithm is to control numerical error through spatial grid and (local) time step adaptation by employing error thresholds. Hence, we are interested in investigating how the solution accuracy behaves when changing the thresholds. An example of the dependence between the absolute error (for two different spatial operators) and the threshold as a function of the total degrees of freedom is demonstrated in Figure 5.19. The actual threshold is the larger one, i.e., the spatial threshold in this particular case, because the relation $\varepsilon_t = 0.1 \cdot \varepsilon_s$ is used. Other used parameters are $j_{\min X} = j_{\min T} = 2$, $J_{\min S} = J_{\min T} = 12$, $N_L = N_R = 1$, $M = 2$. Once again, the Fup collocation operator is one order more accurate than the solution that uses the common finite difference operator. The required accuracy is achieved only for higher thresholds. There are a few main reasons why the actual numerical error becomes larger than the prescribed threshold: a) discontinuous initial conditions (as in the first example), b) the accuracy of the description of spatial derivatives (the spatial threshold is applied on solution values, while the PDE includes its derivatives), c) systematic error accumulation in time due to the finite size of the time step and grid spacing.

Off course, to obtain full efficiency, space and time thresholds should be related so that spatial and temporal discretization errors are of the same magnitude. This is, however, a difficult task since one does not know the solution's spatial and temporal scales in advance. Estimation of the true discretization error and optimal adaptive strategy is an important unresolved topic and definitely requires further investigation.

5.3.3 1D Richards equation

The Richards equation describes the unsaturated flow in porous media that is usually faced with strong nonlinearities. Physically, strong nonlinearity arises due to the complex relationship between pressure, saturation and hydraulic conductivity. Therefore, numerical modeling of such problems is very demanding and usually requires very fine grids, special formulations, implicit conservative methods, different stabilization techniques and robust nonlinear solvers.

A mixed formulation of the Richards equation is used:

$$\frac{\partial \theta(h)}{\partial t} = \frac{\partial}{\partial z}(K(h)\nabla h) + \frac{\partial K(h)}{\partial z} \tag{5.40}$$

where h (L) is the pressure head, θ (L^3/L^3) is the moisture content, K (L/T) is the unsaturated hydraulic conductivity, and z (L) denotes the vertical dimension. The first term on RHS can be differentiated and split into advection and diffusion terms, while the existence of the second term necessitates the construction of a new spatial colocation operator. In this example, Picard linearization is applied, and the moisture content θ is linearized by expansion in a truncated Taylor series with respect to h so that the only unknown in the discretized equation is the pressure head (see also section 4.3.1). The more information regarding the Richards equation and used example can be found in Celia *et al.* [50].

Soil properties are described by the classical Van Genuchten-Mualem model [67]:

$$\theta(h) = \frac{\theta_s - \theta_r}{\left[1 + (\alpha |h|)^n\right]^m} + \theta_r \tag{5.41}$$

$$K(h) = K_s \frac{\left\{1 - (\alpha |h|)^{n-1} \left[(\alpha |h|)^n\right]^{-m}\right\}^2}{\left[1 + (\alpha |h|)^n\right]^{m/2}} \tag{5.42}$$

where $\alpha = 0.0335$, $\theta_s = 0.368$, $\theta_r = 0.102$, $n = 2$, $m = 0.5$, $K_s = 0.00922\,cm/s$. The initial condition is $h(z,0) = -1000\,cm$, and the boundary conditions are $h(0,t) = -1000\,cm$ and $h(100,t) = -75\,cm$. The AFCM parameters are:

$$\begin{array}{ll} j_{\min X} = 4, & J_{\max S} = 10 \\ \varepsilon_S = 10^{-2}, & M = 1 \\ N_L = 1, & N_R = 1 \\ j_{\min T} = 1, & J_{\max T} = 4 \\ \varepsilon_T = 10^{-3}, & \Delta t_{global} = 360.0 \end{array} \tag{5.43}$$

Kirkland *et al.* [113] found that the use of the Crank-Nicolson scheme on the closely related mixed form of the Richards equation fails to reduce the truncation error and is, of course, subject to potential instabilities [58]. Numerical experiments show that AFCM also requires very fine temporal discretization if expression (5.15) is used. Therefore, backward Euler temporal discretization is used.

Figure 5.20 and Figure 5.21 present the AFCM solution and space-time grid, respectively. The Richards equation requires special care for temporal discretization. AFCM uses a global time step Δt_{global}=*10 s* only for the first time step, and after that, Δt_{global} =*360 s* is used until the end of the simulation at *t=72000 s*. Actually, AFCM adaptively finds all small spatial and temporal scales around the front to efficiently obtain the solution during the global time step. The accuracy of the AFCM solution is proved by comparison with a very fine mass conservative FD solution (Figure 5.22) suggested by [50]. Note that Celia *et al.* [50] stated that the solver for the Richards equation should be mass conservative. AFCM does not satisfy the mass conservation property due to the collocation nature of the algorithm. Despite that, an accurate solution, confirming the robustness of AFCM, is obtained.

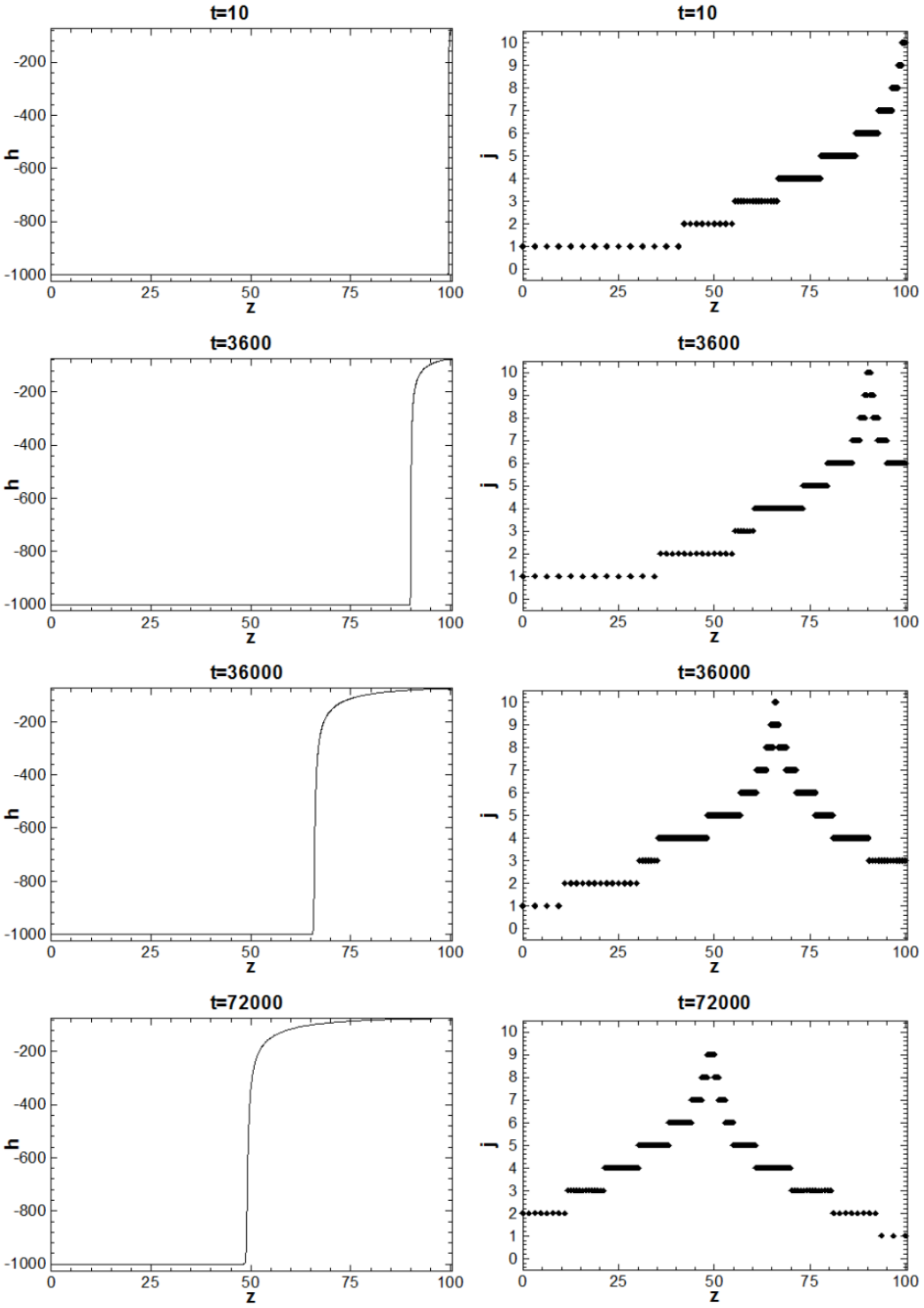


FIGURE 5.20: Numerical solution (left) and effective spatial grid (right) at different time steps.

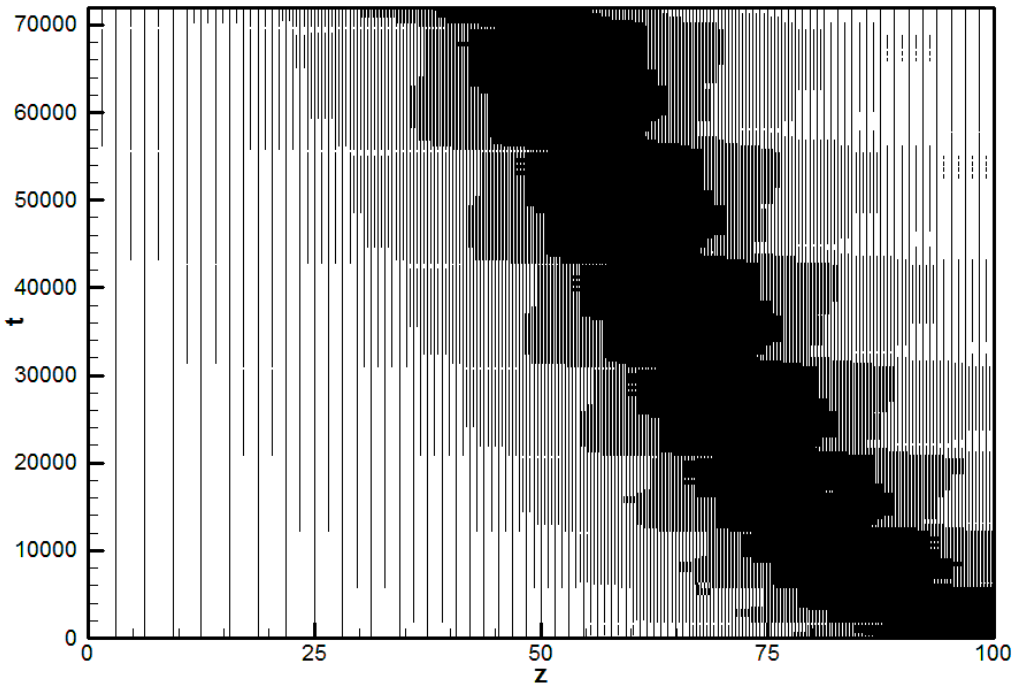


FIGURE 5.21: Space – time grid for space and time adaptation.

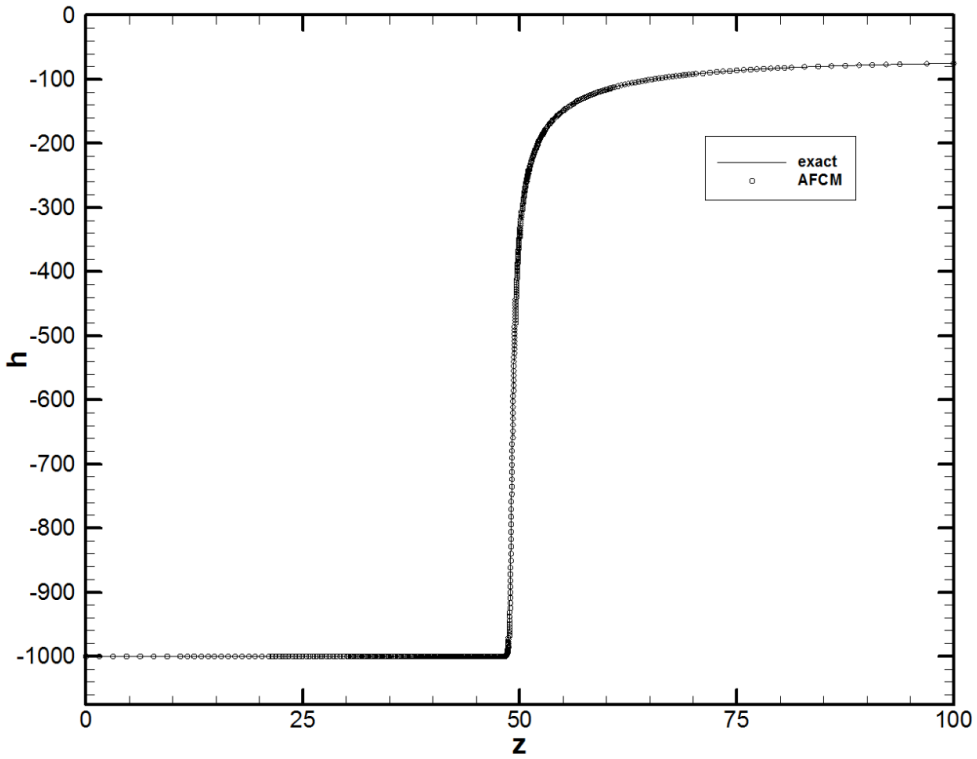


FIGURE 5.22: Comparison between AFCM and the "exact" (mass conservative FD reference solution with fine space-time discretization) solution at t=72000 s.

5.3.4 2D linear advection-diffusion equation

The last example is included to illustrate the generalization and efficiency of AFCM for multidimensional problems. Consider the advection-diffusion equation in the form:

$$\frac{\partial u}{\partial t} = D_x \frac{\partial^2 u}{\partial x^2} + D_y \frac{\partial^2 u}{\partial y^2} - v_x \frac{\partial u}{\partial x} - v_y \frac{\partial u}{\partial y} \tag{5.44}$$

For the problem in an infinite domain with constant coefficients and initial conditions given by:

$$u(x, y, 0) = M_0 \delta(x_0, y_0) \tag{5.45}$$

the analytical solution is given ([114], [115]) as:

$$u(x, y, t) = \frac{M_0}{4\pi t \sqrt{D_x D_y}} \exp\left[-\frac{(x - v_x t)^2}{4 D_x t} - \frac{\left(y - v_y t\right)^2}{4 D_y t} \right] \tag{5.46}$$

where M_0 is a mass initially introduced at the origin of a coordinate system. Here, a unit square domain $(x, y) \in [0, 1] \times [0, 1]$ with Dirichlet boundary conditions set as $u(x, y, t) = 0$ on all boundaries is considered. To avoid the influence of the boundary conditions on the solution, the simulation starts by specifying the initial conditions as the exact solution at $t = 250$, and the calculation is performed until $t = 750$. The initial mass, diffusion and velocity coefficients are given by:

$$\begin{aligned} M_0 &= 1 \cdot 10^{-3} \\ D_x &= D_y = 5 \cdot 10^{-7} \\ v_x &= v_y = 1 \cdot 10^{-3} \end{aligned} \tag{5.47}$$

while the AFCM numerical parameters are the following:

$$\begin{aligned} & j_{\min X} = j_{\min Y} = 2 \\ & J_{\max S} = 12 \\ & \varepsilon_S = 10^{-3}, \quad M = 2 \\ & N_L = 1, \quad N_R = 2 \\ & N_B = 1, \quad N_T = 2 \\ & j_{\min T} = 1, \quad J_{\max T} = 9 \\ & \varepsilon_T = 10^{-4}, \quad \Delta t_{global} = 10.0 \end{aligned} \tag{5.48}$$

Figure 5.23 presents the evolution of the solution and the corresponding computational grid at three different time moments. As explained in section 5.2.3,

a full adaptive spatial grid (Figure 5.23b) is used to calculate the solutions on both the 0^{th} and 1^{st} time levels. The temporal adaptive strategy then estimates the discretization error for each collocation point as the difference between two solutions calculated with different local time steps. Figure 5.23c shows all points that are needed for the solution of the 2^{nd} time level. Since the accuracy requirements are not satisfied only around the moving front, all points far from the front are excluded from further calculation. This procedure is repeated on the next time levels until the local time steps are not sufficiently fine to satisfy the prescribed accuracy for all remaining collocation points.

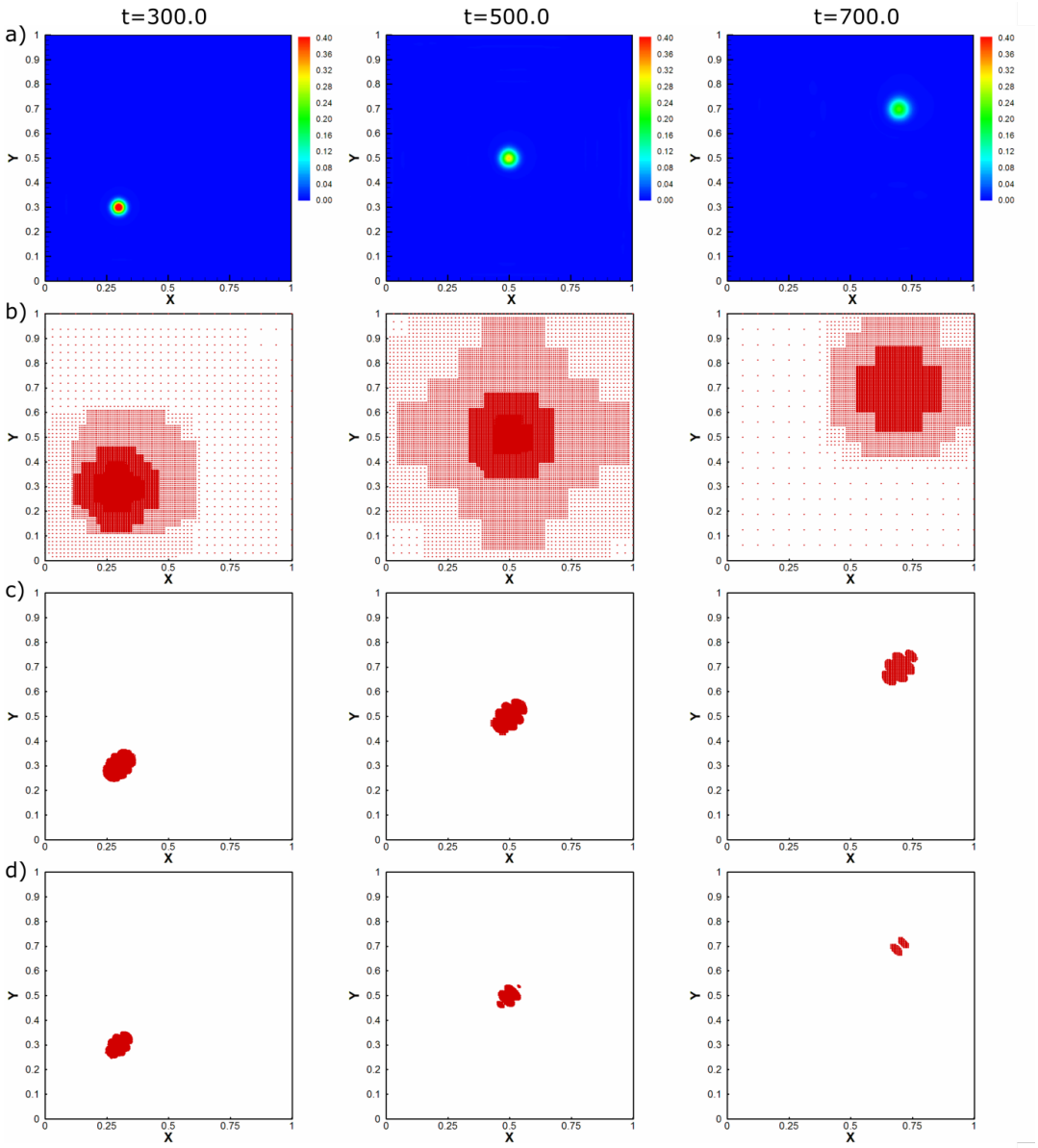


FIGURE 5.23: AFCM numerical solution (a) and active collocation points on: 0^{th} and 1^{st} time levels (b), 2^{nd} time level (c) and 3^{rd} time level (d).

To demonstrate the efficiency of the method, the full space-time AFCM (denoted here as S-T-AFCM), space adaptive version without temporal adaptivity (S-AFCM) and referent non-adaptive finite difference solution (FD) are compared. The solutions of S-T-AFCM are obtained by changing the spatial thresholds between $10^{-1} \leq \varepsilon_S \leq 10^{-5}$ while holding the relation $\varepsilon_T = 0.1 \cdot \varepsilon_S$ constant. Then, for each given threshold, the solution of S-AFCM is calculated by using the same spatial threshold and the finest local time step, while the FD solution (except the finest, due to the high computational cost) is obtained by using both the finest grid spacing and the finest local time step used by the corresponding S-T-AFCM. Here, the efficiency in terms of the maximum absolute error as function of CPU time (Figure 5.24) and the total degrees of freedom (Figure 5.25) is demonstrated. The total number of degrees of freedom (DOF) represents all collocation points in the space-time domain needed to obtain the solution.

First, the performance of the adaptive temporal strategy by comparing S-T-AFCM and S-AFCM is investigated. While the S-AFCM solution is always more precise, the difference in accuracy of the two schemes is practically negligible (for the current example, the average difference is less than 3%). The effectiveness of temporal adaptivity increases with increasing accuracy, and for the most accurate solution, the CPU time and DOF are reduced by approximately a factor of 5. The non-efficiency of the temporal strategy for low-accuracy solutions is due mostly to the repetition of time integration for all collocation points on the 0^{th} and 1^{st} time levels (while needing only few, if any, additional time levels).

Next, S-T-AFCM and FD are compared. The referent non-adaptive FD solution is usually an order of magnitude more accurate than the full adaptive S-T-AFCM solution, mostly because the smallest grid spacing and time step obtained by S-T-AFCM at the beginning of the solution (where the solution is the most demanding) are used during the whole FD simulation, which clearly reduces the accumulation of time integration error. However, the substantial benefit of using S-T-AFCM becomes evident by comparing the computational burden for reaching the same accuracy. The presented results show that the same accuracy can be reached up to 37 times faster and by up to 46 times fewer DOF. Additionally, since these are the first results of the presented algorithm in 2D, the additional programing effort could produce an even more efficient algorithm.

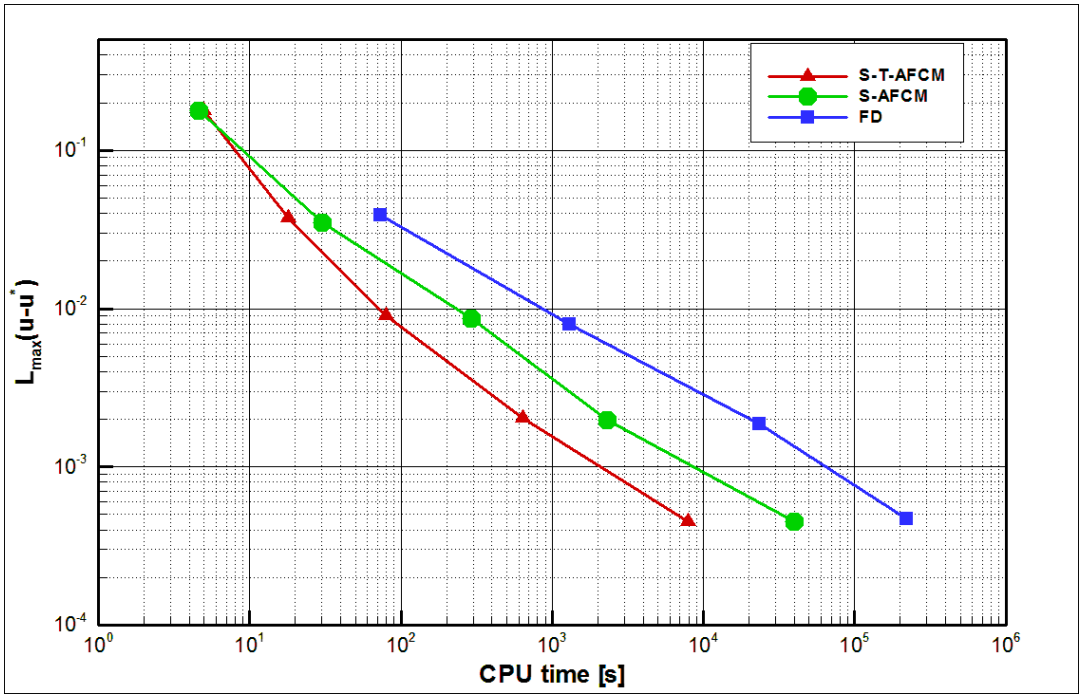


FIGURE 5.24: Efficiency of AFCM: CPU time vs. L_∞ error.

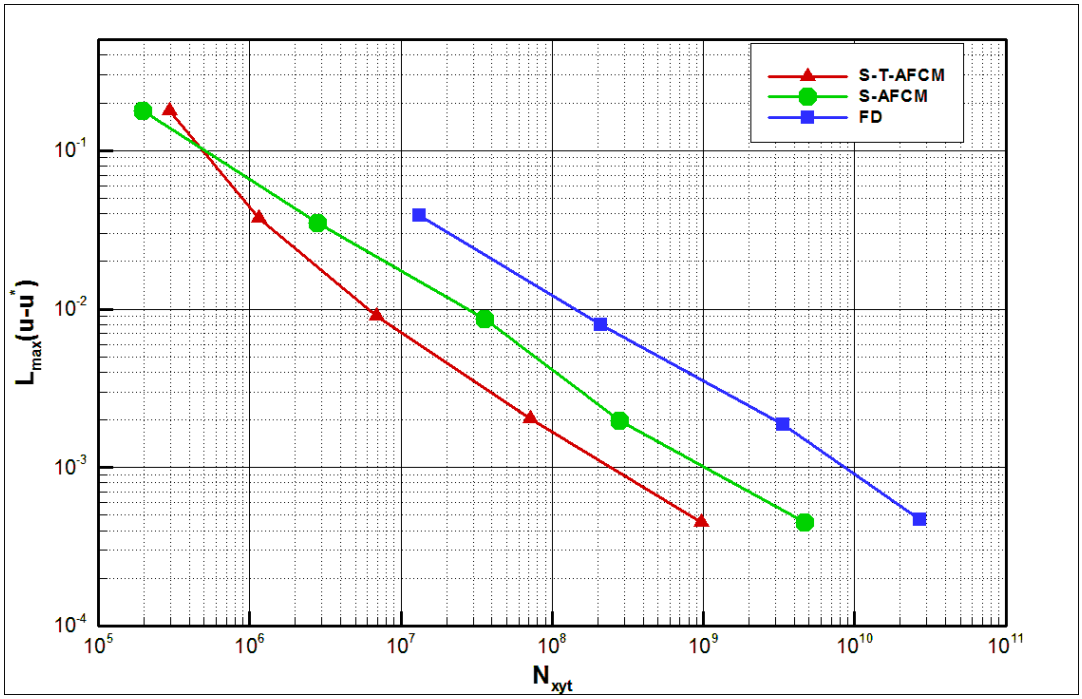


FIGURE 5.25: Efficiency of AFCM: degrees of freedom vs. L_∞ error.

5.4 Conclusions

In this chapter, an implicit full space-time Adaptive Fup Collocation Method (AFCM) with particular application to advection-dominated problems was presented.

The governing equation is discretized by using a collocation approach that separates the spatial and temporal approximations. A new type of 1D spatial discretization is obtained by construction of a novel Fup collocation operator for the approximation of spatial derivatives. In addition to the neighboring solution values at nodes *i-1*, *i*, and *i+1* (as a classical second-order FD operator), the novel collocations operator satisfies the governing differential equation at nodes *i-1* and *i+1*. In comparison with the common FD operator, the developed operator reduces the discretization error (usually by one order of magnitude) while preserving the same non-zero matrix pattern.

Temporal discretization of the governing equation is obtained by construction of temporal Fup collocation operator. However, due to lack of the relevant additional condition needed for local Fup approximation, temporal discretization is identical as Crank-Nicolson.

A particular contribution of this work is the creation of an adaptive procedure not only in space, but also in time. For each global time step, an adaptive spatial grid is obtained by adaptive FCT approximation of the initial condition or previous time step solution. An adaptive strategy in time is developed to use different local time steps for different collocation points during each global time step. This is achieved in the following way: First, solution at the end of the global time step is obtained by using the same local time step for all collocation points. Then, calculation is repeated with a two-times-smaller local time step. The discretization error for each collocation point is estimated as the difference between solutions with different local time steps. For those collocation points where the difference is larger than some user-prescribed temporal threshold, calculation is repeated, again, with two-times-smaller local time step. Other points are excluded for further calculation, while FCT algorithm is used for accurate and stable calculation of the missing values for the next time level. This procedure is repeated until the difference between the current and previous solution for each kept collocation point is not below the prescribed threshold. In this manner, the final solution at the end of each global time step is obtained with different local time steps, and smaller local time steps are used only for collocation points

where temporal solution changes are intensive.

The developed procedure is opposite of the common space-time adaptive schemes where each spatial resolution level has particular local time step. Therefore, the novel adaptive strategy in time resolves all temporal scales independently of the spatial resolution. Moreover, the presented strategy can be applied for both implicit and explicit discretization, which is not a case with most of the existing strategies.

The methodology is applied to the advection-dominated problems in 1D and verifies the efficiency and robustness of proposed concept. First 2D results use finite difference operator and demonstrate that space-time adaptive algorithm can be efficiently applied for multidimensional problems. The extension of collocation operator to higher dimensions is not straightforward due to increased number of additional conditions. This opens different possibilities for its construction and will require further investigations.

Chapter 6

Conclusions

6.1 Summary

In this thesis, spline basis functions are used to develop numerical models with particular application to problems in the field of fluid mechanics. The main motivation for spline basis functions is their powerful numerical approximation properties, which can be summarized as follows:

- The usage of splines as basis functions for numerical analysis provides the opportunity for exact CAD geometry description
- The approach is fundamentally higher order and enables higher continuity throughout the whole domain
- Increased accuracy and robustness per-degree-of-freedom when compared with the same order FEM
- Spline functions of all orders have strictly positive values, which increases the numerical stability
- Demanding mesh generation is generally avoided (spline functions operate on a uniform grid)
- A multiscale description of the geometry, heterogeneity and solution using different numerical resolutions
- Refinement can be performed without affecting the computational geometry or material properties
- Hierarchical spline functions enable efficient adaptive refinement

In the first part of the thesis, the isogeometric analysis (IGA) approach is used as a unified framework for representation of the geometry, material

heterogeneity and solution (flow variables) in a multiscale sense (each system can be represented by its particular numerical resolution). Moreover, all fields are described as continuous and smooth functions by using a linear combination of smooth spline basis functions. Since classical IGA uses the Galerkin and collocation approach, in this work, a third concept, in the form of control volume isogeometric analysis (CV-IGA), is developed and compared with the other two mentioned options. The control volume formulation possesses a few interesting properties, such as the following: 1) local and global conservation property; 2) direct physical meaning of the discretized equations; 3) accuracy very close to that of the Galerkin solution for substantially less computational cost; 4) enhanced accuracy and stability compared to the collocation solution; and 5) optimal groundwater velocity convergence rate, which is essential for tracer or contaminant transport analysis.

These advantages are the main reasons for adopting the control volume formulation as the foundation for the development of a complex 3D karst flow model. The hybrid (discrete-continuum) approach is selected as the best candidate for realistic karst flow modeling. Due to the high nonlinearity introduced by the unsaturated zone, most existing karst flow models neglect important hydrodynamic behavior above the water table. Thus, in this work, matrix flow is described by a more realistic three-dimensional variably saturated flow equation (usually called Richards' equation). The mixed form of the variably saturated equation is used, which, in conjunction with the modified Picard linearization and control volume formulation, enables a mass conservative solution in both the saturated and unsaturated zones. The flow in the karst conduits is represented as one-dimensional and captured by a noninertia wave equation. This equation conveniently describes both the free surface and pressurized flow conditions by simply modifying the capacity term in the governing equations. Since the two flow domains (i.e., matrix and conduit) are governed by different partial differential equations (as well as dimensionality), the equations must be coupled. The coupling is established using a classical first-order exchange term, which is proportional to the head difference on the matrix-conduit interface. The numerical model is based on a segregated (iteratively coupled) scheme in which each flow domain is solved independently by means of the last known solution (of the other domain) to calculate the exchange term.

A particular difficulty with complex 3D karst flow models is the verification and validation process due to the lack of input (such as conduit geometry

and the parameters of the aquifer) and measured data (head distribution and discharge). Therefore, in this work, a novel 3D physical karst flow model that can reproduce many important characteristics of karst is constructed and presented. The most important flow features, such as laminar porous medium flow (through both unsaturated and saturated zones), turbulent conduit flow (both free surface and pressurized conditions), and groundwater exchange between two systems, can be captured. Different boundary conditions and recharge characteristics, in addition to many other possibilities, enable testing of different flow conditions. The pressure distribution in a karst matrix and discharge from both systems are continuously measured during the experiments.

The physical model enables substantial improvement in comparison with real aquifers, especially in terms of the detailed knowledge of the mentioned input and measured data. Two complex experimental setups were used and simulated by the numerical model. The numerical model describes the experimental results mostly inside the range of measurement precision and is capable of realistically simulating demanding flow in the physical model.

Since many problems in fluid dynamics can be regarded as advection-dominated, the second part of this thesis addresses the development of a spline-based collocation algorithm that is particularly applicable for these types of problems. In many cases, advection-dominated problems feature large gradients or sharp interfaces, which can be strongly localized in both spatial and/or temporal directions. For such problems, fine spatial and/or temporal resolutions are needed only in specific locations to resolve the demanding small-scale solution features. For the remaining part of the domain, coarse resolution is usually sufficient. For such cases, adaptive numerical modeling represents one of the most efficient ways to address these types of problems.

Most of the existing adaptive numerical methods are based on adaptive mesh refinement, a procedure that efficiently resolves demanding spatial scales. However, temporal discretization is obtained mostly by classical time-marching procedures, which utilize the same time step for all spatial locations. In that case, the time step is restricted by the finest spatial resolution (CFL condition). To overcome this restriction, scale-dependent local time stepping, where each different spatial resolution level uses a different local time step to satisfy the CFL condition only locally, was introduced.

In this work, two drawbacks of the existing adaptive strategies are recognized. The first is the limitation of space-time adaptive strategies for explicit

temporal discretization. The developed procedures cannot be implemented within implicit methods in a straightforward manner. The second concern is the scale-dependent approach, which uses the smallest time step in regions where the finest spatial resolution is needed. However, the spatial and temporal scales are not necessarily related in such a manner.

Thus, in this work, the previously developed adaptive Fup collocation method (AFCM) is extended with a novel adaptive temporal strategy. The novel strategy addresses both drawbacks of typical local time stepping procedures. Thus, a novel full space-time adaptive algorithm is implemented within implicit temporal discretization and constructed in such a way that all spatial and temporal scales are independently resolved. This independence, as well as the efficiency of the proposed method, is clearly demonstrated via numerical examples.

6.2 Scientific contributions

The main scientific contributions can be summarized as follows:

- Development of control volume isogeometric analysis (CV-IGA)
- Multiscale description of material heterogeneity and flow variables
- Development of a coupled numerical model for the simulation of groundwater flow in karst aquifers
- Construction of a complex and unique 3D karst flow physical model
- Verification and validation of a 3D numerical karst flow model with experimental results
- Construction of a (problem-dependent) collocation operator for accurate discretization of spatial derivatives (only for 1D problems)
- Development of a novel implicit temporal adaptive strategy with independent resolution of all spatial and temporal scales that is especially well suited for advection-dominated problems

6.3 Future perspectives

Finally, some possibilities and directions for future progress can be summarized as follows:

- In this work, the exchange flux validity for a karst flow model is discussed. Further work could investigate different coupling strategies and their validation with additional experimental results. Moreover, extension of the developed model with surface flow and solute transport modeling capabilities are possible directions. Finally, parallelization is needed to enable large-scale simulations for realistic catchment modeling.

- The space-time adaptive algorithm presented in Chapter 5 is applied to scalar advection-diffusion problems. An extension to coupled vector equations (such as Navier-Stokes equations) is a natural step. Some drawbacks of the previously developed adaptive spatial strategy could be improved by using recent developments, such as truncated hierarchical splines (e.g., Wei *et al.* [116]). In this way, Fup basis functions could be used to define the multilevel trial function space capable of direct discretization of PDEs without the construction of a finite difference or collocation operator. Actually, this topic is an active field, and current investigations of our group and initial results show that truncated hierarchical Fup basis functions are quite suitable and efficient for the *hp*-adaptive strategy (Kamber *et al.* [117]). Finally, an ambitious attempt to extend the proposed algorithm to the resolution of multiple-variable solution space-time scales arising in complex multiphysics problems would be extremely interesting.

Appendix A

3D physical karst flow model

A.1 Introduction

The karst aquifers are characterized by the existence of the highly permeable conduit network embedded in the less permeable karst (rock) matrix. This duality of permeability results in different flow conditions and makes karst distinct from other aquifers and more difficult to model. For the purposes of the project: "Groundwater flow modeling in karst aquifers", a 3D karst flow model (Figure A.1) was built in front of the Hydrotechnical Laboratory in Žrnovnica near Split.

FIGURE A.1: Photography of the physical model.

A.2 Technical description

A.2.1 Karst conduits

There are three different pipes installed and labeled C1, C2 and C3 (Figures A.2 and A.3). Pipes C1 and C3 have diameter 0.0155 m, whereas pipe C2's diameter is 0.042 m. The dashed lines in Figure A.2 represent the perforated parts of the pipes. For C1 and C3, the perforations are handmade by drilling holes of 3 mm diameter. The C1 has 80 perforation uniformly distributed along perforated length, while C3 has 4 x 13 perforations on 4 branches as shown in Figure A.2. The C2 pipe is made of two parts: the first is unperforated PVC pipe, and the second is manufactured drainage pipe. The specified values of Manning coefficents are $n_M = 0.009\ \mathrm{m}^{1/3}$ s for C1 and C3 pipes, and $n_M = 0.015\ \mathrm{m}^{1/3}$ s for unperforated part of C2. The Manning coefficient for perforated part of C2 pipe was not specified; thus it needed numerical calibration. The perforated parts were wrapped into geotextile and installed inside fine quartz sand (see section A.2.2) to prevent clogging of perforations. Table A.1 reports the coordinates of points defining pipe geometries (according to Figures A.2).

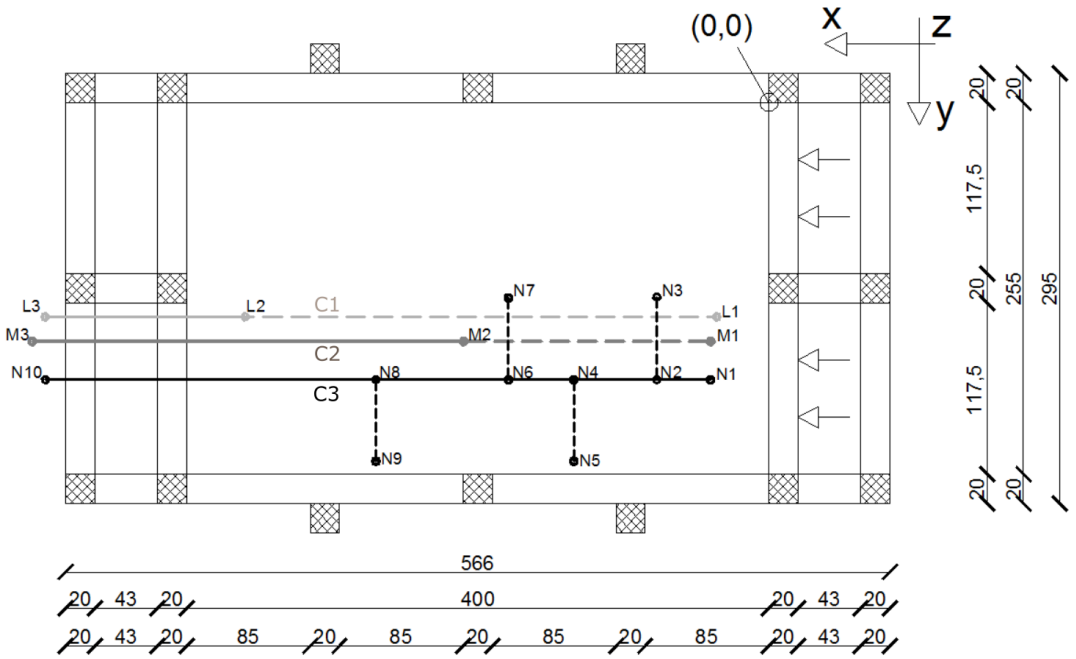


FIGURE A.2: Ground plan of physical model with positions of pipes (karst conduits).

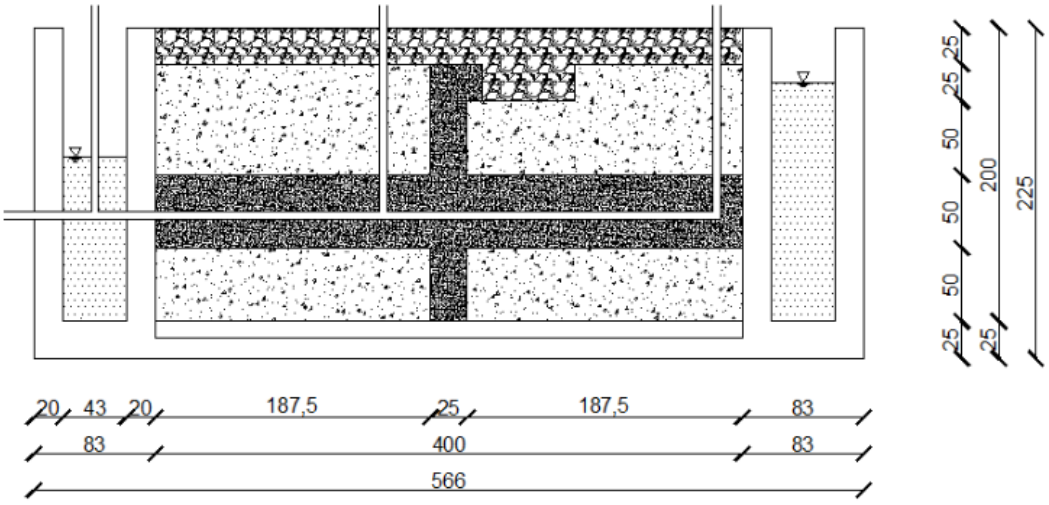


FIGURE A.3: Cross section through conduit C2.

FIGURE A.4: Photography of three conduit outlets.

TABLE A.1: Coordinates of the conduit geometries.

COORDINATES	**x [cm]**	**y [cm]**	**z [cm]**
L1	36	147	75
L2	360	147	71.5
L3	497	147	70
M1	40	164	75
M2	210	164	73
M3	506	164	70
N1	40	190	75
N2	77	190	74.6
N3	77	134	74.6
N4	134	190	74
N5	134	246	74
N6	179	190	73.5
N7	179	134	73.5
N8	270	190	72.5
N9	270	246	72.5
N10	497	190	70

A.2.2 Distribution of materials

Three different types of the material are used: coarse quartz sand (CQS), fine quartz sand (FQS) and gravel (G). The material is filled in the eight layers of 25 cm with a different distribution of each material. In this manner, a certain level of heterogeneity in the matrix was achieved. Gravel is filled in the highest layer (eighth), and it represents a surface layer of epikarst. In the center part of the matrix, there is a fine quartz sand throughout the total cross-section as a weak permeable core. The same material is used around the conduits. as mentioned in the previous section. Tables A.2-A.3 and Figures A.6-A.13 present the details about the distribution of material.

FIGURE A.5: Filling the material.

TABLE A.2: Used materials.

MATERIAL	LABEL	SIZE RANGE [mm]	PERCENTAGE OF TOTAL VOLUME [%]
COARSE QUARTZ SAND	CQS	0-4	67.34
FINE QUARTZ SAND	FQS	0.1-0.6	13.98
GRAVEL	G	8-16	18.68

TABLE A.3: Allocation of material.

	LAYER HEIGHT [cm]	**LAYER THICKNESS [cm]**	**MATERIAL TYPE AND DISTRIBUTION**		
			CQS [%]	FQS [%]	G [%]
Layer 1	0-25	25	11.69	0.81	-
Layer 2	25-50	25	10.76	0.81	0.93
Layer 3	50-75	25	7.54	4.34	0.62
Layer 4	75-100	25	6.89	4.34	1.27
Layer 5	100-125	25	9.48	0.92	2.10
Layer 6	125-150	25	10.89	0.81	0.80
Layer 7	150-175	25	10.10	1.94	0.46
Layer 8	175-200	25	-	-	12.5

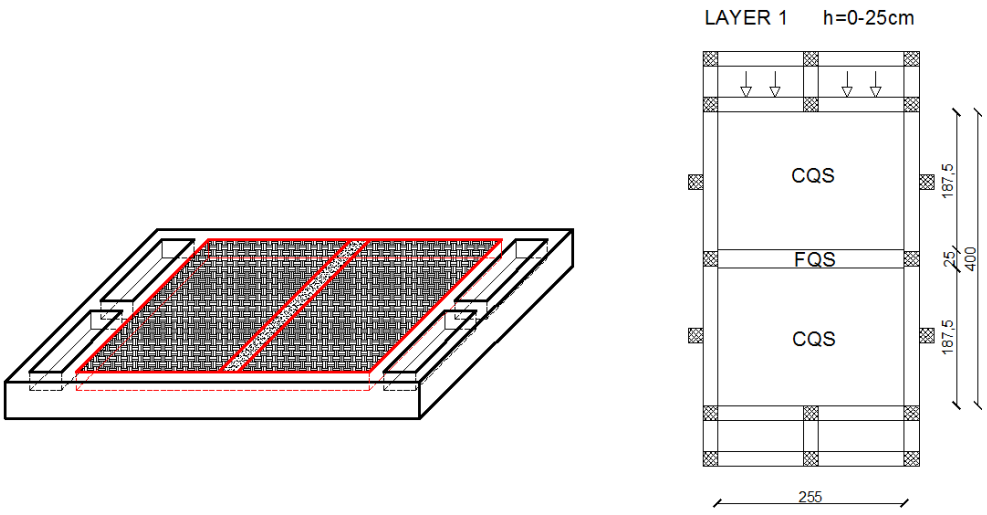


FIGURE A.6: Schematic representation of layer 1.

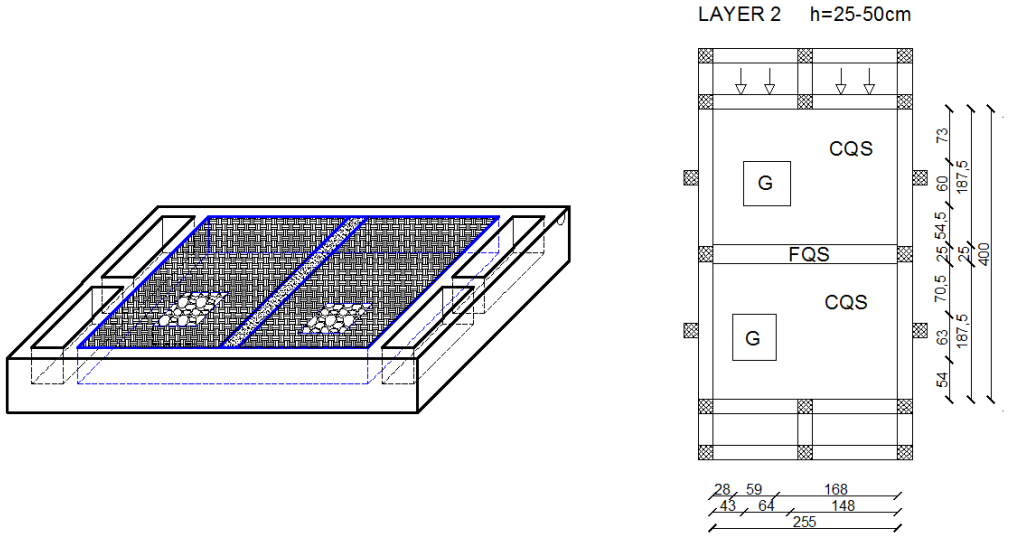


FIGURE A.7: Schematic representation of layer 2.

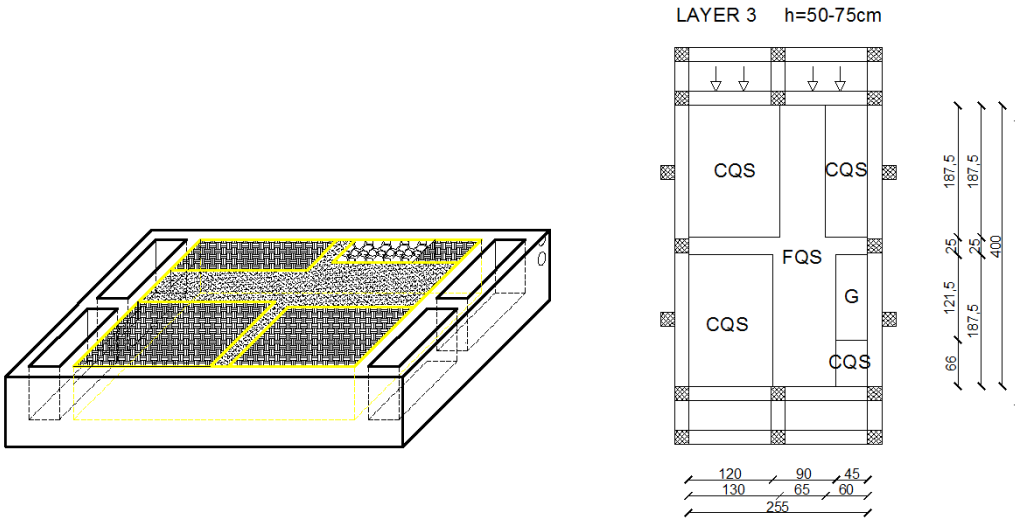


FIGURE A.8: Schematic representation of layer 3.

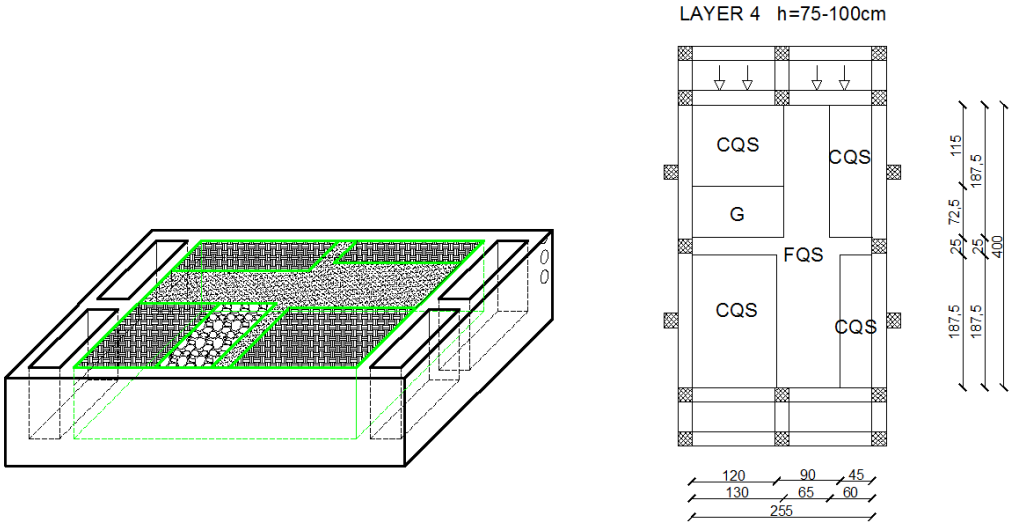


FIGURE A.9: Schematic representation of layer 4.

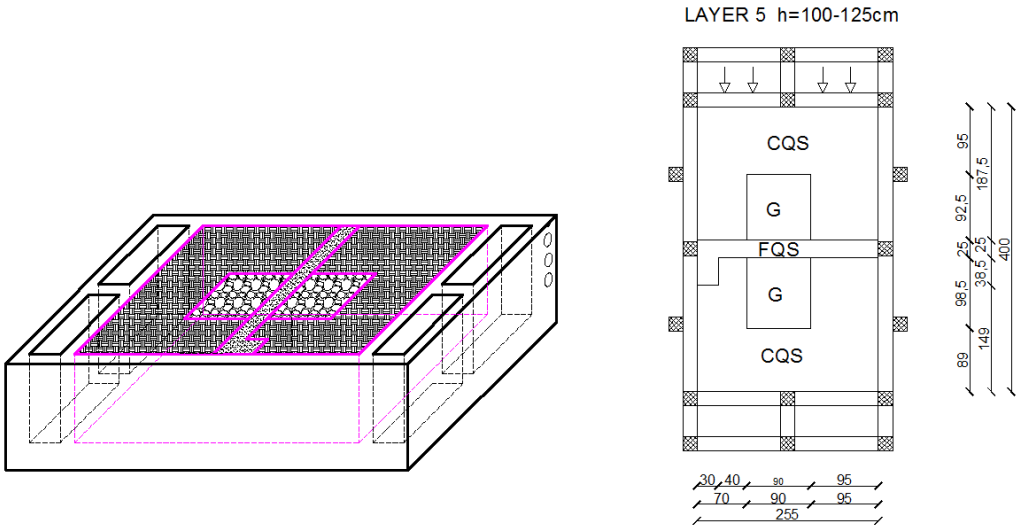


FIGURE A.10: Schematic representation of layer 5.

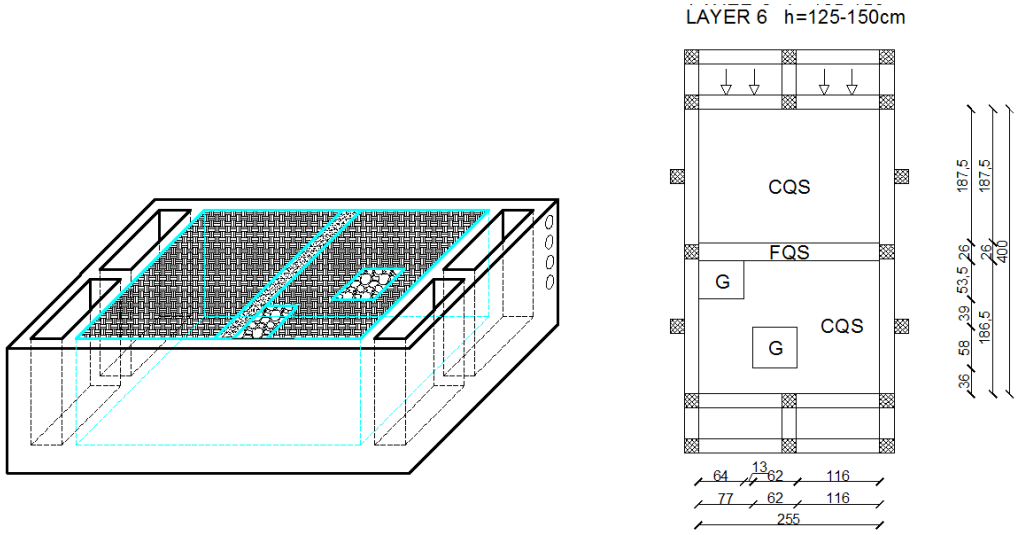


FIGURE A.11: Schematic representation of layer 6.

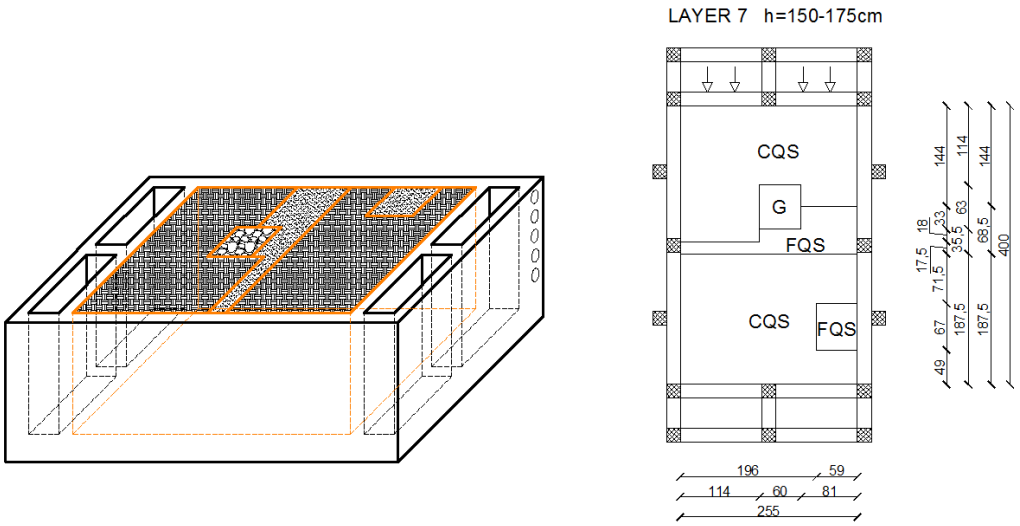


FIGURE A.12: Schematic representation of layer 7.

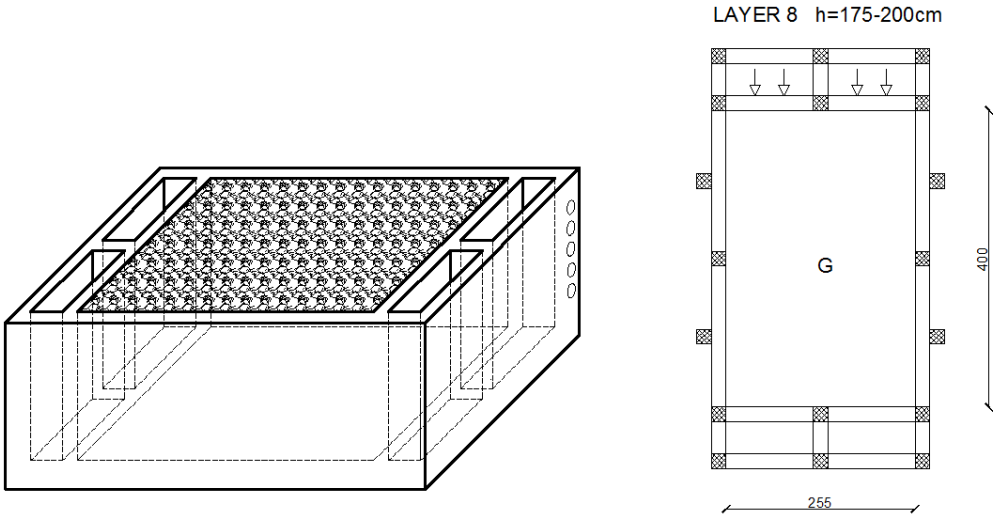


FIGURE A.13: Schematic representation of layer 8.

A.2.3 Soil properties

The soil samples were sent to a specialized laboratory for the measurements of saturated and unsaturated soil parameters. However, because of extremely high conductivity, the results for gravel (G) were not obtained. Therefore, in addition to the obtained values in the laboratory (for CQS and FQS), the Darcys experiments were performed for all three materials. Then numerical calibration was performed on additional experiments (presented in section A.3) to obtain final values. During numerical calibrations, there was a need to account for soil anisotropy to obtain reasonable results. Table A.4 presents the obtained ranges and final values for saturated hydraulic conductivities. The specific storage coefficient (S_s) values were taken from the literature.

TABLE A.4: Hydraulic conductivities: measured range and final values after calibration.

	Measurement range		**Final value**	
Material	K_{min} [m/s]	K_{max} [m/s]	K_H [m/s]	K_V [m/s]
CQS	$4.14 \cdot 10^{-4}$	$8.00 \cdot 10^{-3}$	$3.40 \cdot 10^{-3}$	$1.26 \cdot 10^{-3}$
FQS	$7.12 \cdot 10^{-5}$	$3.47 \cdot 10^{-4}$	$2.00 \cdot 10^{-4}$	$8.00 \cdot 10^{-5}$
G	$2.80 \cdot 10^{-2}$	$7.10 \cdot 10^{-2}$	$6.00 \cdot 10^{-2}$	$6.00 \cdot 10^{-2}$

Unsaturated soil properties are described by water retention curves. These curves present the relationship between soil suction and volumetric water content and are shown for three used materials in Figure A.14. The

measured values (for CQS and FQS) are fitted by the bimodal van Genuchten model, while the classical (unimodal) curve for G is obtained by the literature guidelines and numerical calibration. The values of the van Genuchten parameters are given in Table A.5.

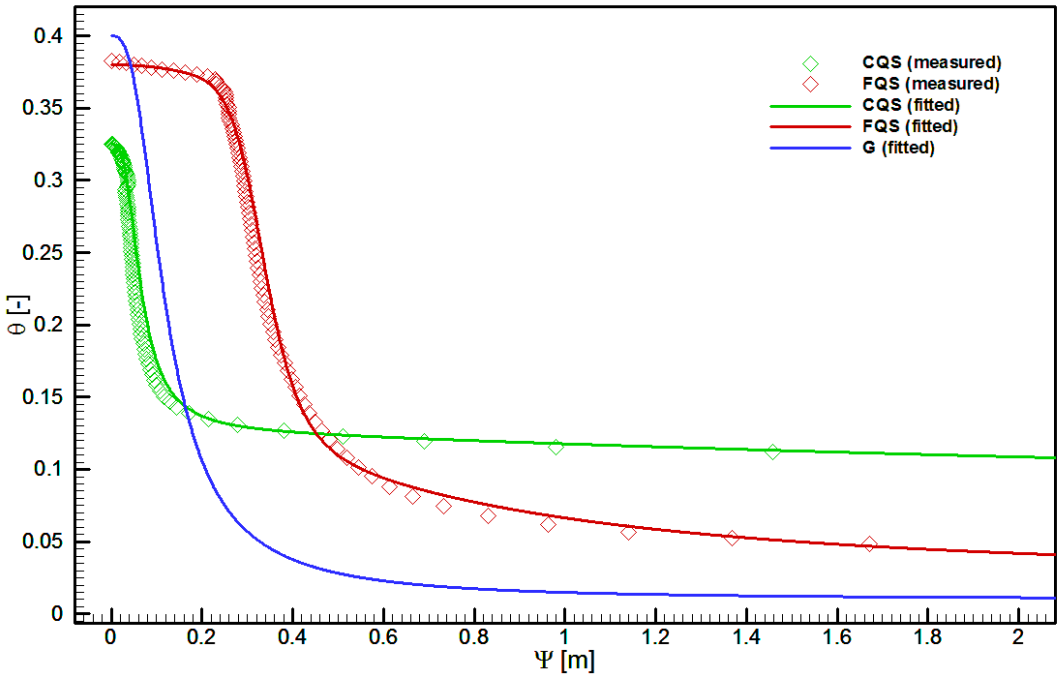


FIGURE A.14: Water retention curves for used materials.

TABLE A.5: Unsaturated soil parameters for three different materials: coarse quartz sand (CQS), fine quartz sand (FQS) and gravel (G).

Parameter	CQS	FQS	G
α_1 [m^{-1}]	18.00	3.00	10.00
α_2 [m^{-1}]	0.50	2.00	-
n_1 [-]	3.2	9.2	2.9
n_2 [-]	1.3	2.2	-
w_1 [-]	0.62	0.67	1.0
w_2 [-]	0.38	0.33	0.0
τ [-]	2.96	2.295	0.5
η [-]	0.325	0.38	0.40
θ_r [-]	0.005	0.02	0.01

A.2.4 Rainfall simulation

On the top of the model, there are sprinklers and shower heads for rainfall simulation. The sprinklers are used for rainfall simulation over total

"epikarst" area, while shower heads produce high-intensity concentrated rainfall. There are two digital flowmeters for the measurement of total discharge from both rainfall components. These discharges are not perfectly uniform distributed over the considered area, but for the simplification of the numerical modeling their distribution is considered uniform. Figure A.15 shows a picture of the rain simulation.

FIGURE A.15: Photography of rain simulation by two shower heads.

A.2.5 Water regulation in reservoirs

The water levels in the reservoirs are regulated by height-adjustable overflows (Figures A.16 and A.17). The large perimeters were used to prevent water elevations during rainfalls and high pumping discharge rates (needed for circulation of water).

FIGURE A.16: Overflow in upstream reservoir.

FIGURE A.17: Overflow in downstream reservoir.

A.2.6 Discharge measurements

In order to make sense of the experiments, it is necessary to have accurate measurements of the flow from matrix and the conduits. The matrix discharge is regarded as total water that flows out from soil inside downstream reservoir. Since the reservoir level is fixed by high capacity overflow, the matrix discharge is equal to overflow discharge. The reservoir overflow water (matrix discharge) and conduit outflow are directed toward two V-Notch weirs that are used as measurement devices (Figure A.18). Each of them has a previously calibrated discharge scale (as function of water level inside weir box), and a camera is used to record the flow rates at given time intervals (see section A.2.9). For large and fast changes in discharge (such as sudden opening of conduit), there is smoothing (time lag) in the measurements. Since different discharges are realized for different water levels, the filling/emptying of this volume of water is the main reason for this time lag. This time lag is estimated less than 1 min for maximum discharge variations used in the experiments.

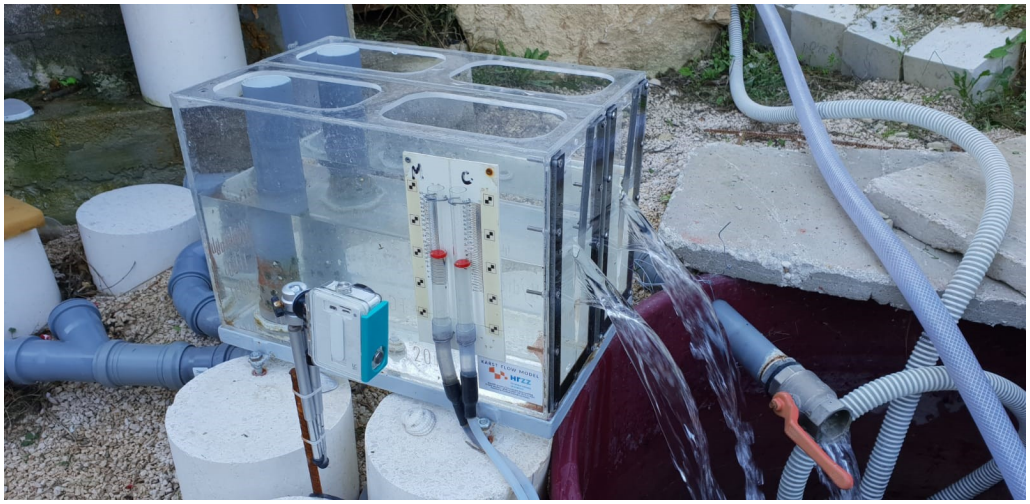


FIGURE A.18: V-Notch weirs for discharge measurment.

A.2.7 Piezometers

Pressure distribution is measured in 44 fixed points by piezometers that are installed through the foundation slab and rise vertically (Figure A.19). These pipes are perforated only on their ends (inside the porous medium) and connected with transparent tubings (level gauges) on front side of model (Figure A.1). Water levels in this tubes represent head values in particular points inside the domain, and a camera is used to record the head variations at given time intervals (see section A.2.9). Figure A.20 shows the position of all installed piezometers, while their coordinates are given in Table A.6. Some piezometers are shown to be malfunctioning and excluded from measurements, probably because of entrapped air or perforations clogging. The following is a list of piezometers that were not used: A11, B8, C9, C10, D2, D9 and D10.

FIGURE A.19: Installation of piezometers.

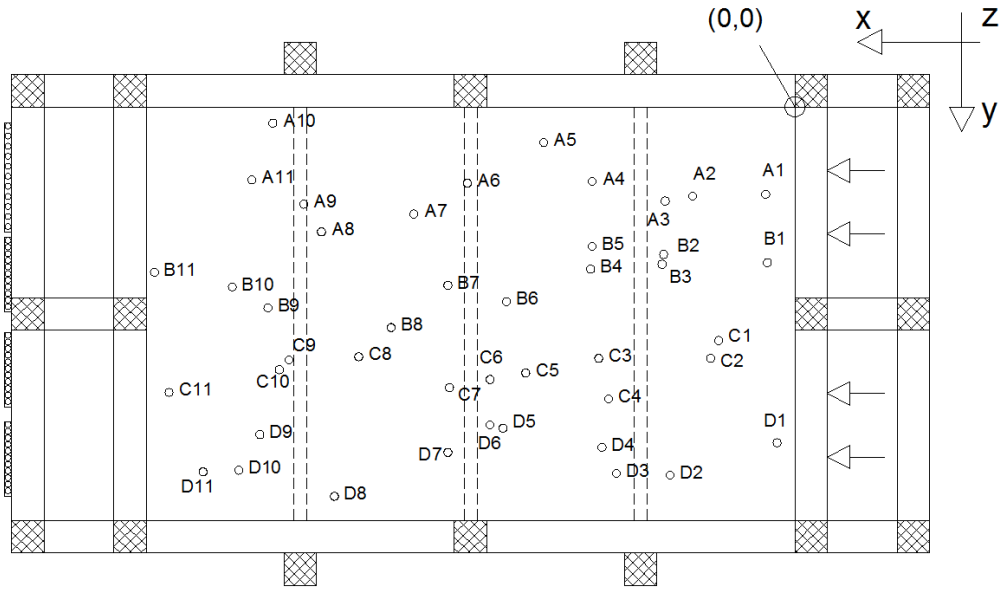


FIGURE A.20: Positions of piezometers.

TABLE A.6: Coordinates of the piezometers.

PIEZOMETER	x [cm]	y [cm]	z [cm]	PIEZOMETER	x [cm]	y [cm]	z [cm]
A1	17	54	77	C1	47	144	105
A2	63	54	100	C2	52	155	33
A3	80	58	32	C3	121	155	65
A4	125	46	75	C4	115	180	95
A5	155	22	100	C5	166	164	39
A6	202	47	42	C6	188	168	65
A7	235	66	65	C7	203	173	102
A8	292	77	106	C8	269	154	31
A9	303	60	33	C9	318	162	60
A10	322	10	100	C11	386	176	33
A11	335	45	70	C10	312	156	100
B1	17	96	33	D1	11	207	33
B2	81	91	65	D2	77	227	70
B3	82	97	100	D3	110	226	100
B4	126	100	33	D4	117	210	34
B5	125	86	65	D5	180	198	69
B6	178	120	100	D6	188	196	120
B7	216	110	43	D7	214	213	36
B8	249	136	60	D8	284	240	60
B9	325	124	100	D9	330	202	100
B10	347	111	33	D10	343	224	34
B11	395	102	60	D11	365	225	85

A.2.8 Boreholes

The 25 fully perforated vertical pipes are installed from soil surface, and they are denoted as boreholes. Since these pipes are fully perforated, borehole packers (Figure A.21) are used to isolate part of pipe (approximately 10 cm) on the specific depth. The pressure sensor or water hose can be installed at any depth of the borehole and used for pressure measurements or performing partially penetrating pumping tests, respectively. Figure A.22 shows the positions of all installed boreholes, while their coordinates are given in Table A.7.

FIGURE A.21: Borehole pakers.

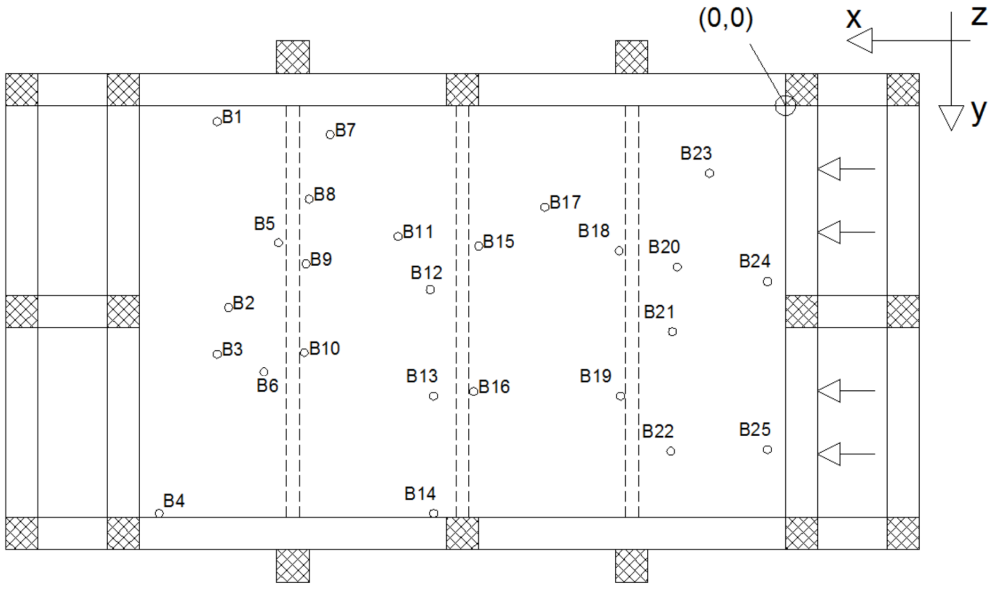


FIGURE A.22: Borehole positions.

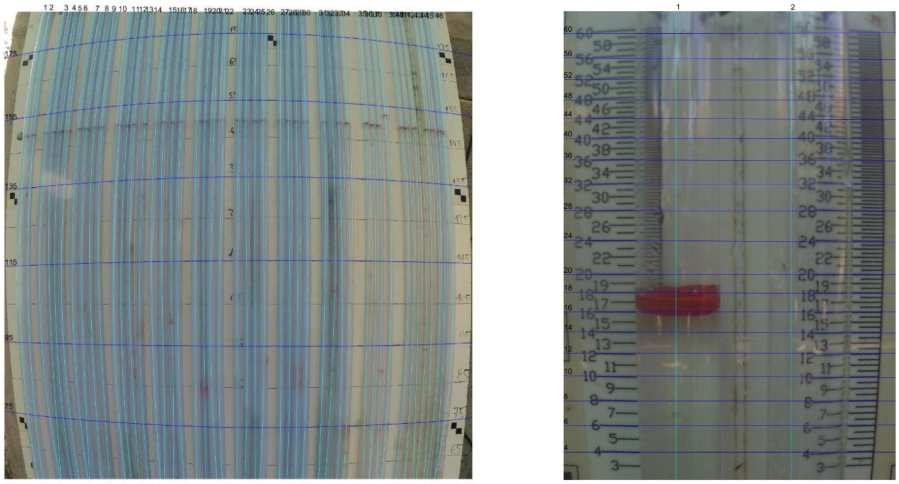
TABLE A.7: Coordinates of the boreholes.

BOREHOLE	x [cm]	y [cm]	BOREHOLE	x [cm]	y [cm]
B1	353	10	B14	219	254
B2	345	124	B15	196	87
B3	352	154	B16	204	172
B4	388	255	B17	150	58
B5	314	85	B18	105	90
B6	318	166	B19	103	177
B7	282	17	B20	75	105
B8	296	58	B21	76	184
B9	298	97	B22	84	209
B10	299	153	B23	53	42
B11	241	76	B24	9	109
B12	236	110	B25	8	213
B13	230	174	-	-	-

A.2.9 Photogrammetric tracking

Considering the large numbers of level gauges employed during the experiment (Figure A.23), experiment length, and low temporal constant of the measured system, photogrammetry was used as an inexpensive and robust measurement method. For image acquisition, an inexpensive wide-angle sports camera was used. The fixed aperture of the camera was f/2.8 with a focal length of 3 mm. The camera was mounted 1 m from level gauge panel, with optical axis horizontal and perpendicular to measurement plate. An additional camera, with the same parameters, was used to monitor flow rate. Image acquisition was automatic, relying on an integrated time-lapse mode. The interval of time-lapse was 5 s.

The data acquisition process from images was semiautomatic. Initially, the operator was defining curvilinear tracks that were aligned with the optical path of level gauges and curvilinear levels that were defined according to height marks plotted onto the measurement plane. These two sets of curves defined the prescribed paths of level gauge knobs and actual, physical world water levels. There was no need for the rectification of photographs since the described method introduced corrections via curvilinear coordinates. The



(A) Piezometric head scale (B) Discharge scale

FIGURE A.23: Curves defining paths along water height level (cyan) and curves defining levels (blue). Buoy for water level detection are red blobs.

curves used to describe paths and levels were polynomial expressions (independent variable x (horizontal) for level curves and independent variable y (vertical) for path curves). The detection of a buoy inside was based on the difference in color regarding the level gauge background. Along each track, a 5 pixel strip on both sides of the track was taken into account. The collected pixel strip was squarely differentiated with buoy color vector and averaged both over 11-pixel span and three components of the color resulting 1D plot of the most likely position of buoy along the track. In order to minimize color variations between frames due to reflections and changes in natural lighting, histogram matching was performed before the definition of the buoy position. The detected position was the mean of the 1D plot. After detection, the pixel position of the buoy was translated to the water level using the interpolation of pixel positions of the intersection of specified track and level lines and water column height (Figure A.24).

After initiation, the procedure is automated for each frame. The primary sources of error were due to rotation of buoys and image space nonlinearities. The rough mistakes were corrected by operators after completed data acquisition, and systemic mistakes were taken into account as measurement uncertainties. The number of rejected measurements were dependent on the track position and optical opaqueness of the water-level column tube, but usually less than 5% of all measurements. Typical error of measurement was 1.5 cm for the water-level column and 1 l/s for flow rate measurement. The

measurement errors were proven by using additional concurrent measurements of the stated properties as a benchmark.

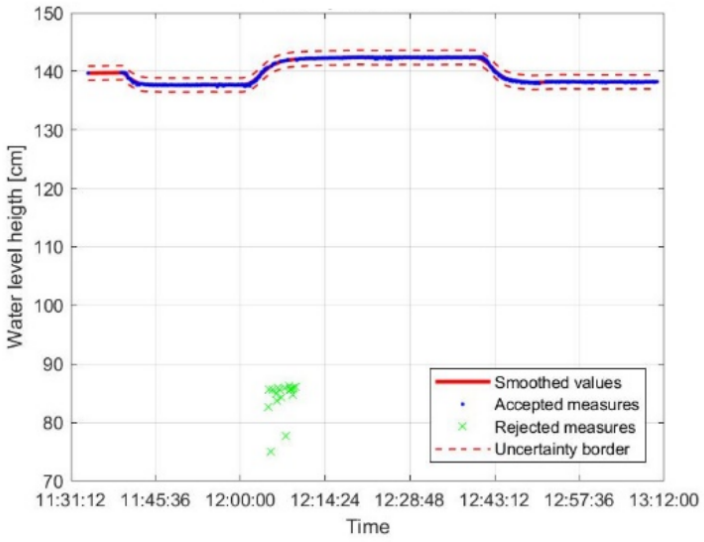


FIGURE A.24: Typical measurements output, depicting accepted values, rejected values, smoothed output and error boundary.

A.3 Additional experiments

In the following, additional experiments used for the calibration of hydraulic conductivities are briefly described. The full experimental data for the additional experiments, together with two presented test cases, are available online (see [26].

Seven (E01-E07) steady-state flow experiments were performed (Table A.8). The first four experiments (E01-E04) consider only matrix flow since all three conduit systems were clogged. The borehole packers were used to perform partially penetrating pumping tests, and the difference in water levels in the two reservoirs was practically negligible (0.003 m) to ensure similar hydraulic gradients in both the horizontal and vertical directions. Boreholes B11, B12, B13 and B17 (see Figure A.22) were used, and similar pumping discharge was applied at different depths.

The remaining three experiments (E05-E07) were used to test the influence of the conduit on the matrix steady-state flow. The head difference in the two reservoirs was unchanged and set to $\Delta H = 0.165$ m. Experiment E05 considers only matrix flow, whereas E06 and E07 consider matrix interaction with conduits C1 and C2, respectively. The discharges and hydraulic head

values in the piezometers were measured after the conduit-matrix steady-state was achieved.

TABLE A.8: Description of additional experiments.

Experiment ID	H_u [m]	H_d [m]	Q_r [l/min]	Borehole	Packer depth [m]	Active conduit
E01	1.455	1.452	32.30	12	0.74	-
E02	1.455	1.452	32.40	13	0.87	-
E03	1.455	1.452	33.30	17	0.87	-
E04	1.455	1.452	32.20	11	0.62	-
E05	1.455	1.290	0.00	-	-	-
E06	1.455	1.290	0.00	-	-	C1
E07	1.455	1.290	0.00	-	-	C2

Appendix B

Derivation of Fup collocation operator

B.1 Spatial Fup collocation operator

The solution of system (5.10) yields Fup coefficients as a general function of three solution values, two temporal derivatives and related physical and numerical parameters:

$$
\begin{aligned}
\alpha_{i\text{-}2} = & \frac{\Delta x_i^2}{144 \cdot S_i} \cdot \left[\begin{array}{l} u_{i\text{-}1} \cdot [36 \cdot D_{i\text{-}1} \cdot (2052 \cdot D_{i+1} \text{-} 691 \cdot \Delta x_i \cdot v_{i+1}) \text{-} 90 \cdot \Delta x_i \cdot v_{i\text{-}1} \cdot (36 \cdot D_{i+1} \text{-} 13 \cdot \Delta x_i \cdot v_{i+1})] + \\ + u_i \cdot [36 \cdot (13 \cdot \Delta x_i \cdot v_{i+1} \text{-} 36 \cdot D_{i+1}) \cdot (36 \cdot D_{i\text{-}1} \text{-} 13 \cdot \Delta x_i \cdot v_{i\text{-}1})] + \\ + u_{i+1} \cdot [90 \cdot (2 \cdot D_{i+1} \text{-} \Delta x_i \cdot v_{i+1}) \cdot (36 \cdot D_{i\text{-}1} \text{-} 13 \cdot \Delta x_i \cdot v_{i\text{-}1})] + \\ + \frac{\partial u_{i\text{-}1}}{\partial t} \cdot [\Delta x_i^2 \cdot (23436 \cdot D_{i+1} \text{-} 7813 \cdot \Delta x_i \cdot v_{i+1})] + \\ + \frac{\partial u_{i+1}}{\partial t} \cdot [25 \cdot \Delta x_i^2 \cdot (\Delta x_i \cdot v_{i\text{-}1}) \cdot (13 \cdot \Delta x_i \cdot v_{i+1} \text{-} 36 \cdot D_{i+1})] \end{array} \right] \\
\alpha_{i\text{-}1} = & \frac{\Delta x_i^2}{144 \cdot S_i} \cdot \left[\begin{array}{l} u_{i\text{-}1} \cdot [18 \cdot (936 \cdot D_{i+1} \text{-} 313 \cdot \Delta x_i \cdot v_{i+1}) \cdot (2 \cdot D_{i\text{-}1} + \Delta x_i \cdot v_{i\text{-}1})] \text{-} \\ \text{-} u_i \cdot [180 \cdot \Delta x_i \cdot v_{i\text{-}1} \cdot (36 \cdot D_{i+1} \text{-} 13 \cdot \Delta x_i \cdot v_{i+1})] \text{-} \\ \text{-} u_{i+1} \cdot [400 \cdot \Delta x_i \cdot v_{i\text{-}1} \cdot (\Delta x_i \cdot v_{i+1} \text{-} 2 \cdot D_{i+1})] \text{-} \\ \text{-} \frac{\partial u_{i\text{-}1}}{\partial t} \cdot [5 \cdot \Delta x_i^2 \cdot (936 \cdot D_{i+1} \text{-} 313 \cdot \Delta x_i \cdot v_{i+1})] \text{-} \\ \text{-} \frac{\partial u_{i+1}}{\partial t} \cdot [125 \cdot \Delta x_i^2 \cdot (\Delta x_i \cdot v_{i\text{-}1})] \end{array} \right] \\
\alpha_i = & \text{-} \frac{\Delta x_i^2}{144 \cdot S_i} \cdot \left[\begin{array}{l} u_{i\text{-}1} \cdot [90 \cdot (36 \cdot D_{i+1} \text{-} 13 \cdot \Delta x_i \cdot v_{i+1}) \cdot (2 \cdot D_{i\text{-}1} + \Delta x_i \cdot v_{i\text{-}1})] + \\ + u_i \cdot [36 \cdot \Delta x_i \cdot v_{i\text{-}1} \cdot (13 \cdot \Delta x_i \cdot v_{i+1} \text{-} 36 \cdot D_{i+1}) \cdot (36 \cdot D_{i\text{-}1} + 13 \cdot \Delta x_i \cdot v_{i\text{-}1})] + \\ + u_{i+1} \cdot [90 \cdot (2 \cdot D_{i+1} \text{-} \Delta x_i \cdot v_{i+1}) \cdot (36 \cdot D_{i\text{-}1} + 13 \cdot \Delta x_i \cdot v_{i\text{-}1})] \text{-} \\ \text{-} \frac{\partial u_{i\text{-}1}}{\partial t} \cdot [25 \cdot \Delta x_i^2 \cdot (36 \cdot D_{i+1} \text{-} 13 \cdot \Delta x_i \cdot v_{i+1})] \text{-} \\ \text{-} \frac{\partial u_{i+1}}{\partial t} \cdot [25 \cdot \Delta x_i^2 \cdot (36 \cdot D_{i\text{-}1} + 13 \cdot \Delta x_i \cdot v_{i\text{-}1})] \end{array} \right] \\
\alpha_{i+1} = & \text{-} \frac{\Delta x_i^2}{144 \cdot S_i} \cdot \left[\begin{array}{l} u_{i\text{-}1} \cdot [450 \cdot \Delta x_i \cdot v_{i+1} \cdot (2 \cdot D_{i\text{-}1} + \Delta x_i \cdot v_{i\text{-}1})] \text{-} \\ \text{-} u_i \cdot [180 \cdot \Delta x_i \cdot v_{i+1} \cdot (36 \cdot D_{i\text{-}1} + 13 \cdot \Delta x_i \cdot v_{i\text{-}1})] + \\ + u_{i+1} \cdot [18 \cdot (936 \cdot D_{i\text{-}1} \text{-} 313 \cdot \Delta x_i \cdot v_{i\text{-}1}) \cdot (\Delta x_i \cdot v_{i+1} \text{-} 2 \cdot D_{i+1})] \text{-} \\ \text{-} \frac{\partial u_{i\text{-}1}}{\partial t} \cdot [125 \cdot \Delta x_i^3 \cdot v_{i+1} \cdot] + \\ + \frac{\partial u_{i+1}}{\partial t} \cdot [5 \cdot \Delta x_i^2 \cdot (936 \cdot D_{i\text{-}1} \text{-} 313 \cdot \Delta x_i \cdot v_{i\text{-}1})] \end{array} \right] \\
\alpha_{i+2} = & \frac{\Delta x_i^2}{144 \cdot S_i} \cdot \left[\begin{array}{l} u_{i\text{-}1} \cdot [90 \cdot \Delta x_i \cdot v_{i\text{-}1} \cdot (36 \cdot D_{i+1} + 13 \cdot \Delta x_i \cdot v_{i+1}) \cdot (2 \cdot D_{i\text{-}1} + \Delta x_i \cdot v_{i\text{-}1})] \text{-} \\ \text{-} u_i \cdot [36 \cdot (36 \cdot D_{i+1} + 13 \cdot \Delta x_i \cdot v_{i+1}) \cdot (36 \cdot D_{i\text{-}1} + 13 \cdot \Delta x_i \cdot v_{i\text{-}1})] + \\ + u_{i+1} \cdot [648 \cdot D_{i\text{-}1} \cdot (114 \cdot D_{i+1} + 5 \cdot \Delta x_i \cdot v_{i+1}) + 18 \cdot \Delta x_i \cdot v_{i\text{-}1} \cdot (1382 \cdot \cdot D_{i+1} + 65 \cdot \Delta x_i \cdot v_{i+1})] \text{-} \\ \text{-} \frac{\partial u_{i\text{-}1}}{\partial t} \cdot [25 \cdot \Delta x_i^2 \cdot (36 \cdot D_{i+1} + 13 \cdot \Delta x_i \cdot v_{i+1})] + \\ + \frac{\partial u_{i+1}}{\partial t} \cdot [\Delta x_i^2 \cdot (23436 \cdot D_{i\text{-}1} + 7813 \cdot \Delta x_i \cdot v_{i\text{-}1})] \end{array} \right]
\end{aligned}
\tag{B.1}
$$

where:

$$S_i = \Delta x_i^2 \cdot (D_{i\text{-}1} \cdot (936 \cdot D_{i+1} \text{-} 313 \cdot \Delta x_i \cdot v_{i+1}) + \Delta x_i \cdot v_{i\text{-}1} \cdot (313 \cdot D_{i+1} \text{-} 104 \cdot \Delta x_i \cdot v_{i+1})) \tag{B.2}$$

Substituting the calculated coefficients (B.1) into (5.7), the Fup collocation operator for the first and second spatial derivative at the i-th collocation point has the form:

$$\frac{\partial u_i}{\partial x} = \frac{1}{S_i} \cdot \begin{bmatrix} u_{i\text{-}1} \cdot [\Delta x_i \cdot 18 \cdot (2 \cdot D_{i\text{-}1} + \Delta x_i \cdot v_{i\text{-}1}) \cdot (4 \cdot \Delta x_i \cdot v_{i+1} \text{-} 13 \cdot D_{i+1})] + \\ + u_i \cdot \left[\Delta x_i^2 \cdot 90 \cdot (D_{i\text{-}1} \cdot v_{i+1} + D_{i+1} \cdot v_{i\text{-}1})\right] + \\ + u_{i+1} \cdot [\Delta x_i \cdot 18 \cdot (2 \cdot D_{i+1} \text{-} \Delta x_i \cdot v_{i+1}) \cdot (13 \cdot D_{i\text{-}1} + 4 \cdot \Delta x_i \cdot v_{i\text{-}1})] + \\ + \frac{\partial u_{i\text{-}1}}{\partial t} \cdot \left[\Delta x_i^3 \cdot 5 \cdot (13 \cdot D_{i+1} \text{-} 4 \cdot \Delta x_i \cdot v_{i+1})\right] \text{-} \\ \text{-} \frac{\partial u_{i+1}}{\partial t} \cdot \left[\Delta x_i^3 \cdot 5 \cdot (13 \cdot D_{i\text{-}1} + 4 \cdot \Delta x_i \cdot v_{i\text{-}1})\right] \end{bmatrix} \tag{B.3}$$

$$\frac{\partial^2 u_i}{\partial x^2} = \frac{1}{S_i} \cdot \begin{bmatrix} u_{i\text{-}1} \cdot [18 \cdot (2 \cdot D_{i\text{-}1} + \Delta x_i \cdot v_{i\text{-}1}) \cdot (36 \cdot D_{i+1} \text{-} 13 \cdot \Delta x_i \cdot v_{i+1})] \text{-} \\ \text{-} u_i \cdot [36 \cdot (D_{i\text{-}1} \cdot (72 \cdot D_{i+1} \text{-} 31 \cdot \Delta x_i \cdot v_{i+1}) + \Delta x_i \cdot v_{i\text{-}1} \cdot (31 \cdot D_{i+1} \text{-} 13 \cdot \Delta x_i \cdot v_{i+1}))] + \\ + u_{i+1} \cdot [18 \cdot (2 \cdot D_{i+1} \text{-} \Delta x_i \cdot v_{i+1}) \cdot (36 \cdot D_{i\text{-}1} + 13 \cdot \Delta x_i \cdot v_{i\text{-}1})] + \\ + \frac{\partial u_{i\text{-}1}}{\partial t} \cdot \left[\Delta x_i^2 \cdot 5 \cdot (13 \cdot \Delta x_i \cdot v_{i+1} \text{-} 36 \cdot D_{i+1})\right] \text{-} \\ \text{-} \frac{\partial u_{i+1}}{\partial t} \cdot \left[\Delta x_i^2 \cdot 5 \cdot (36 \cdot D_{i\text{-}1} + 13 \cdot \Delta x_i \cdot v_{i\text{-}1})\right] \end{bmatrix} \tag{B.4}$$

Finally, by substituting expressions (B.3) and (B.4) into (5.4), the semi-discrete ordinary differential equation for the i-th collocation point (at the end of the local time step) reads:

$$\frac{\partial u_i^{t+\Delta t}}{\partial t} = \frac{1}{S_i} \cdot$$

$$\begin{bmatrix} u_{i\text{-}1}^{t+\Delta t} \cdot \left[18 \cdot \left(2 \cdot D_{i\text{-}1}^{t+\Delta t} + \Delta x_i \cdot v_{i\text{-}1}^{t+\Delta t}\right) \cdot \left(D_i^{t+\Delta t} \cdot \left(36 \cdot D_{i+1}^{t+\Delta t} \text{-} 13 \cdot \Delta x_i \cdot v_{i+1}^{t+\Delta t}\right) + \Delta x_i \cdot v_i^{t+\Delta t} \cdot \left(13 \cdot D_{i+1}^{t+\Delta t} \text{-} 4 \cdot \Delta x_i \cdot v_{i+1}^{t+\Delta t}\right)\right)\right] + \\ + u_i^{t+\Delta t} \cdot \left[18 \cdot \left(\begin{array}{l} D_{i\text{-}1}^{t+\Delta t} \cdot \left(2 \cdot D_i^{t+\Delta t} \cdot \left(72 \cdot D_{i+1}^{t+\Delta t} \text{-} 31 \cdot \Delta x_i \cdot v_{i+1}^{t+\Delta t}\right) + 5 \cdot \Delta x_i^2 \cdot v_i^{t+\Delta t} \cdot v_{i+1}^{t+\Delta t}\right) + \\ + \Delta x_i \cdot v_{i\text{-}1}^{t+\Delta t} \cdot \left(2 \cdot D_i^{t+\Delta t} \cdot \left(31 \cdot D_{i+1}^{t+\Delta t} \text{-} 13 \cdot \Delta x_i \cdot v_{i+1}^{t+\Delta t}\right) + 5 \cdot D_{i+1}^{t+\Delta t} \cdot \Delta x_i \cdot v_i^{t+\Delta t}\right) \end{array} \right)\right] + \\ + u_{i+1}^{t+\Delta t} \cdot \left[18 \cdot \left(2 \cdot D_{i+1}^{t+\Delta t} + \Delta x_i \cdot v_{i+1}^{t+\Delta t}\right) \cdot \left(D_{i\text{-}1}^{t+\Delta t} \cdot \left(36 \cdot D_i^{t+\Delta t} \text{-} 13 \cdot \Delta x_i \cdot v_i^{t+\Delta t}\right) + \Delta x_i \cdot v_{i\text{-}1}^{t+\Delta t} \cdot \left(13 \cdot D_i^{t+\Delta t} \text{-} 4 \cdot \Delta x_i \cdot v_i^{t+\Delta t}\right)\right)\right] \text{-} \\ \text{-} \frac{\partial u_{i\text{-}1}^{t+\Delta t}}{\partial t} \cdot \left[5 \cdot \left(D_i^{t+\Delta t} \cdot \left(36 \cdot D_{i+1}^{t+\Delta t} \text{-} 13 \cdot \Delta x_i \cdot v_{i+1}^{t+\Delta t}\right) + \Delta x_i \cdot v_i^{t+\Delta t} \cdot \left(13 \cdot D_{i+1}^{t+\Delta t} \text{-} 4 \cdot \Delta x_i \cdot v_{i+1}^{t+\Delta t}\right)\right)\right] \text{-} \\ \text{-} \frac{\partial u_{i+1}^{t+\Delta t}}{\partial t} \cdot \left[5 \cdot \left(D_{i\text{-}1}^{t+\Delta t} \cdot \left(36 \cdot D_i^{t+\Delta t} \text{-} 13 \cdot \Delta x_i \cdot v_i^{t+\Delta t}\right) + \Delta x_i \cdot v_{i\text{-}1}^{t+\Delta t} \cdot \left(13 \cdot D_i^{t+\Delta t} \text{-} 4 \cdot \Delta x_i \cdot v_i^{t+\Delta t}\right)\right)\right] \end{bmatrix} \tag{B.5}$$

B.2 Temporal Fup collocation operator

The solution of system (5.14) yields the Fup coefficients needed for the construction of the temporal operator along a temporal line as a general function of two initial conditions (the zero and first derivative at the beginning of the local time step) and the unknown solution value at the end of the local time step:

$$\begin{aligned} \beta^{t\text{-}\Delta t} &= \frac{5 \cdot u^t \text{-} 31 \cdot \frac{\partial u^t}{\partial t} \cdot \Delta t + 13 \cdot u^{t+\Delta t}}{72} \\ \beta^{t} &= \frac{23 \cdot u^t + 5 \cdot \frac{\partial u^t}{\partial t} \cdot \Delta t \text{-} 5 \cdot u^{t+\Delta t}}{72} \\ \beta^{t+\Delta t} &= \frac{5 \cdot u^t + 5 \cdot \frac{\partial u^t}{\partial t} \cdot \Delta t + 13 \cdot u^{t+\Delta t}}{72} \\ \beta^{t+2\cdot\Delta t} &= \frac{\text{-}49 \cdot u^t \text{-} 31 \cdot \frac{\partial u^t}{\partial t} \cdot \Delta t + 67 \cdot u^{t+\Delta t}}{72} \end{aligned} \tag{B.6}$$

B.3 Final form of discretized equations

The discretization process results in a final system of algebraic equations that can be (assuming uniform grid spacing) expressed in the form:

$$\begin{array}{l} a_{1,1}\cdot u_1^{t+\Delta t}+a_{1,2}\cdot u_2^{t+\Delta t}+a_{1,3}\cdot u_3^{t+\Delta t}=b_1 \\ a_{i,i\text{-}1}\cdot u_{i\text{-}1}^{t+\Delta t}+a_{i,i}\cdot u_i^{t+\Delta t}+a_{i,i+1}\cdot u_{i+1}^{t+\Delta t}=b_i; \qquad i=2,\ldots,n\text{-}1 \\ a_{n,n\text{-}2}\cdot u_{n\text{-}2}^{t+\Delta t}+a_{n,n\text{-}1}\cdot u_{n\text{-}1}^{t+\Delta t}+a_{n,n}\cdot u_n^{t+\Delta t}=b_n \end{array} \tag{B.7}$$

For every internal collocation point ($2\leq i\leq n$-1), coefficients are given:

$$a_{i,i\text{-}1}=\frac{2}{\Delta t\cdot S_i^{t+\Delta t}}\cdot\left[\begin{array}{l} 18\cdot D_{i\text{-}1}^{t+\Delta t}\cdot\Delta t+\Delta x_i\cdot\left(9\cdot\Delta t\cdot v_{i\text{-}1}^{t+\Delta t}\text{-}5\cdot\Delta x_i\right)\cdot \\ \cdot\left(D_i^{t+\Delta t}\cdot\left(36\cdot D_{i+1}^{t+\Delta t}\text{-}13\cdot\Delta x_i\cdot v_{i\text{-}1}^{t+\Delta t}\right)+\Delta x_i\cdot v_i^{t+\Delta t}\cdot\left(13\cdot D_{i+1}^{t+\Delta t}\text{-}4\cdot\Delta x_i\cdot v_{i+1}^{t+\Delta t}\right)\right) \end{array}\right] \tag{B.8}$$

$$a_{i,i}=\text{-}\frac{2}{\Delta t\cdot S_i^{t+\Delta t}}\cdot\left[\begin{array}{l} D_{i\text{-}1}^{t+\Delta t}\cdot\left(\begin{array}{l} 18\cdot D_i^{t+\Delta t}\cdot\Delta t\cdot\left(72\cdot D_{i+1}^{t+\Delta t}\text{-}31\cdot\Delta x_i\cdot v_{i+1}^{t+\Delta t}\right)+ \\ +\Delta x_i^2\cdot\left(936\cdot D_{i+1}^{t+\Delta t}+v_{i+1}^{t+\Delta t}\cdot\left(45\cdot\Delta t\cdot v_i^{t+\Delta t}\text{-}313\cdot\Delta x_i\right)\right) \end{array}\right)+ \\ +\Delta x_i\cdot v_{i\text{-}1}^{t+\Delta t}\cdot\left(\begin{array}{l} 18\cdot D_i^{t+\Delta t}\cdot\Delta t\cdot\left(31\cdot D_{i+1}^{t+\Delta t}\text{-}13\cdot\Delta x_i\cdot v_{i+1}^{t+\Delta t}\right)+ \\ +\Delta x_i\cdot\left(D_{i+1}^{t+\Delta t}\cdot\left(45\cdot\Delta t\cdot v_i^{t+\Delta t}+313\cdot\Delta x_i\right)\text{-}104\cdot\Delta x_i^2\cdot v_{i+1}^{t+\Delta t}\right) \end{array}\right) \end{array}\right] \tag{B.9}$$

$$a_{i,i+1}=\frac{2}{\Delta t\cdot S_i^{t+\Delta t}}\cdot\left[\begin{array}{l} 18\cdot D_{i+1}^{t+\Delta t}\cdot\Delta t\text{-}\Delta x_i\cdot\left(9\cdot\Delta t\cdot v_{i+1}^{t+\Delta t}+5\cdot\Delta x_i\right)\cdot \\ \cdot\left(D_i^{t+\Delta t}\cdot\left(36\cdot D_{i\text{-}1}^{t+\Delta t}+13\cdot\Delta x_i\cdot v_{i\text{-}1}^{t+\Delta t}\right)+\Delta x_i\cdot v_i^{t+\Delta t}\cdot\left(\text{-}13\cdot D_{i\text{-}1}^{t+\Delta t}\text{-}4\cdot\Delta x_i\cdot v_{i\text{-}1}^{t+\Delta t}\right)\right) \end{array}\right] \tag{B.10}$$

$$b_i=\text{-}\frac{\Delta x_i^2}{\Delta t\cdot S_i^{t+\Delta t}}\cdot \left[\begin{array}{l} u_{i\text{-}1}^t\cdot\left[10\cdot D_i^{t+\Delta t}\cdot\left(36\cdot D_{i+1}^{t+\Delta t}\text{-}13\cdot\Delta x_i\cdot v_{i+1}^{t+\Delta t}\right)+10\cdot\Delta x_i\cdot v_i^{t+\Delta t}\cdot\left(13\cdot D_{i+1}^{t+\Delta t}\text{-}4\cdot\Delta x_i\cdot v_{i+1}^{t+\Delta t}\right)\right]+ \\ +u_i^t\cdot\left[2\cdot D_{i\text{-}1}^{t+\Delta t}\cdot\left(936\cdot D_{i+1}^{t+\Delta t}\text{-}313\cdot\Delta x_i\cdot v_{i+1}^{t+\Delta t}\right)+2\cdot\Delta x_i\cdot v_{i\text{-}1}^{t+\Delta t}\cdot\left(313\cdot D_{i+1}^{t+\Delta t}\text{-}104\cdot\Delta x_i\cdot v_{i+1}^{t+\Delta t}\right)\right]+ \\ +u_{i+1}^t\cdot\left[10\cdot\left(D_{i\text{-}1}^{t+\Delta t}\cdot\left(36\cdot D_i^{t+\Delta t}\text{-}13\cdot\Delta x_i\cdot v_i^{t+\Delta t}\right)+13\cdot D_i^{t+\Delta t}\cdot\Delta x_i\cdot v_{i\text{-}1}^{t+\Delta t}\text{-}4\cdot\Delta x_i^2\cdot v_{i\text{-}1}^{t+\Delta t}\cdot v_i^{t+\Delta t}\right)\right]+ \\ +\frac{\partial u_{i\text{-}1}^t}{\partial t}\cdot\left[5\cdot\Delta t\cdot\left(D_i^{t+\Delta t}\cdot\left(36\cdot D_{i+1}^{t+\Delta t}\text{-}13\cdot\Delta x_i\cdot v_{i+1}^{t+\Delta t}\right)+\Delta x_i\cdot v_i^{t+\Delta t}\cdot\left(13\cdot D_{i+1}^{t+\Delta t}\text{-}4\cdot\Delta x_i\cdot v_{i+1}^{t+\Delta t}\right)\right)\right]+ \\ +\frac{\partial u_i^t}{\partial t}\cdot\left[\Delta t\cdot\left(D_{i\text{-}1}^{t+\Delta t}\cdot\left(936\cdot D_{i+1}^{t+\Delta t}\text{-}313\cdot\Delta x_i\cdot v_{i+1}^{t+\Delta t}\right)+2\cdot\Delta x_i\cdot v_{i\text{-}1}^{t+\Delta t}\cdot\left(313\cdot D_{i+1}^{t+\Delta t}\text{-}104\cdot\Delta x_i\cdot v_{i+1}^{t+\Delta t}\right)\right)\right]+ \\ +\frac{\partial u_{i+1}^t}{\partial t}\cdot\left[5\cdot\Delta t\cdot\left(D_{i\text{-}1}^{t+\Delta t}\cdot\left(36\cdot D_i^{t+\Delta t}\text{-}13\cdot\Delta x_i\cdot v_i^{t+\Delta t}\right)+13\cdot D_i^{t+\Delta t}\cdot\Delta x_i\cdot v_{i\text{-}1}^{t+\Delta t}\text{-}4\cdot\Delta x_i^2\cdot v_{i\text{-}1}^{t+\Delta t}\cdot v_i^{t+\Delta t}\right)\right] \end{array}\right] \tag{B.11}$$

In the case of a boundary conditions defined by (5.5), the boundary equation with Dirichlet boundary condition produces:

$$a_{1,1}=1,\quad a_{1,2}=0,\quad a_{1,3}=0 \tag{B.12}$$

$$b_1=G^{t+\Delta t} \tag{B.13}$$

while for a boundary equation with Neumann boundary condition:

$$a_{n,n\text{-}2}=\frac{\Delta x_n}{\Delta t\cdot S_n}\cdot\left[10\cdot D_n^{t+\Delta t}\cdot\left(18\cdot D_{n\text{-}2}^{t+\Delta t}\cdot\Delta t+\Delta x_n\cdot\left(9\cdot\Delta t\cdot v_{n\text{-}2}^{t+\Delta t}\text{-}5\cdot\Delta x_n\right)\right)\right] \tag{B.14}$$

$$a_{n,n\text{-}1}=\text{-}\frac{\Delta x_n}{S_n}\cdot\left[36\cdot D_n^{t+\Delta t}\cdot\left(36\cdot D_{n\text{-}2}^{t+\Delta t}+13\cdot\Delta x_n\cdot v_{n\text{-}2}^{t+\Delta t}\right)\right] \tag{B.15}$$

$$a_{n,n}=\frac{\Delta x_n}{\Delta t\cdot S_n}\cdot\left[2\cdot\left(D_{n\text{-}2}^{t+\Delta t}\cdot\left(558\cdot D_n^{t+\Delta t}\cdot\Delta t+313\cdot\Delta x_n^2\right)+\Delta x_n\cdot v_{n\text{-}2}^{t+\Delta t}\cdot\left(189\cdot D_n^{t+\Delta t}\cdot\Delta t+104\cdot\Delta x_n^2\right)\right)\right] \tag{B.16}$$

$$b_n=H^{t+\Delta t}+\frac{\Delta x_n^2}{\Delta t\cdot S_n}\cdot\left[\begin{array}{l}\text{-}u_{n\text{-}2}^{t+\Delta t}\cdot\left(50\cdot D_n^{t+\Delta t}\cdot\Delta x_n\right)+u_n^{t+\Delta t}\cdot 2\cdot\left(313\cdot D_{n\text{-}2}^{t+\Delta t}+104\cdot\Delta x_n\cdot v_{n\text{-}2}^{t+\Delta t}\right)\\ \text{-}\frac{\partial u_{n\text{-}2}^{t+\Delta t}}{\partial t}\cdot 25\cdot\Delta t\cdot\Delta x_n+\frac{\partial u_n^{t+\Delta t}}{\partial t}\cdot\Delta t\cdot\Delta x_n\cdot\left(313\cdot D_{n\text{-}2}^{t+\Delta t}+104\cdot\Delta x_n\cdot v_{n\text{-}2}^{t+\Delta t}\right)\end{array}\right] \tag{B.17}$$

Bibliography

[1] S. V. Patankar, *Numerical heat transfer and fluid flow*. Hemisphere Pub. Corp., 1980, p. 197, ISBN: 9780891165224.

[2] J. H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics*. Springer, 2002, p. 423, ISBN: 978-3-540-78091-5. DOI: 10.1007/978-3-642-98037-4. arXiv: arXiv:1011.1669v3. [Online]. Available: http://link.springer.com/10.1007/978-3-642-98037-4.

[3] K.-J. Bathe, *Finite Element Procedures*. Jan. 2006, vol. 2, ISBN: 9780979004902.

[4] O. C. Zienkiewicz, R. L. R. L. Taylor, and J. Z. Zhu, *Finite element method : its basis and fundamentals*. Butterworth-Heinemann, 2013, ISBN: 9780080951355.

[5] O. C. Zienkiewicz, R. L. R. L. Taylor, and P. Nithiarasu, *The finite element method for fluid dynamics*, p. 544, ISBN: 9780080951379.

[6] T. J. Hughes, J. A. Cottrell, and Y. Bazilevs, "Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement", *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 39-41, pp. 4135–4195, 2005, ISSN: 00457825. DOI: 10.1016/j.cma.2004.10.008.

[7] K. Höllig, *Finite Element Methods with B-Splines*. Society for Industrial and Applied Mathematics (SIAM; 3600 Market Street; Floor 6; Philadelphia; PA 19104), 2003, p. 145, ISBN: 978-0-89871-699-3. DOI: 10.1137/1.9780898717532. [Online]. Available: http://epubs.siam.org/doi/book/10.1137/1.9780898717532.

[8] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, "Isogeometric Analysis Toward Intergration of CAD and FEA", p. 335, 2009.

[9] R. V. Rvačev V.L., "Pro odnu finitnu funkciju", *DAN URSR. Ser. A (in Russian)*, no. 6, pp. 705–707. 1971.

[10] B. Gotovac and V. Kozulic, "On a selection of basis functions in numerical analyses of engineering problems", *International Journal for Engineering Modelling*, vol. 12, no. 1-4, pp. 25–41, 1999, ISSN: 13301365.

[11] H. Gotovac, R. Andricevic, and B. Gotovac, "Multi-resolution adaptive modeling of groundwater flow and transport problems", *Advances in Water Resources*, vol. 30, no. 5, pp. 1105–1126, 2007, ISSN: 03091708. DOI: 10.1016/j.advwatres.2006.10.007.

[12] N. Brajčić Kurbaša, "Eksponencijalne atomske bazne funkcije : razvoj i primjena : disertacija", PhD thesis, 2016. [Online]. Available: https://dr.nsk.hr/islandora/object/gradst{\%}3A898.

[13] N. B. Kurbaša, B. Gotovac, and V. Kozulić, "Atomic exponential basis function Eup(x, ω) - Development and application", *CMES - Computer Modeling in Engineering and Sciences*, vol. 111, no. 6, pp. 493–530, 2016, ISSN: 15261492.

[14] J. A. Cottrell, T. J. R. Hughes, and A. Reali, "Studies of refinement and continuity in isogeometric structural analysis", *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 41-44, pp. 4160–4183, 2007, ISSN: 00457825. DOI: 10.1016/j.cma.2007.04.007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782507001703.

[15] D. C. Thomas, M. A. Scott, J. A. Evans, K. Tew, and E. J. Evans, "Bézier projection: A unified approach for local projection and quadrature-free refinement and coarsening of NURBS and T-splines with particular application to isogeometric design and analysis", *Computer Methods in Applied Mechanics and Engineering*, vol. 284, pp. 55–105, 2015, ISSN: 00457825. DOI: 10.1016/j.cma.2014.07.014. arXiv: 1404.7155. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782514002448.

[16] Y. W. Bekele, T. Kvamsdal, A. M. Kvarving, and S. Nordal, "Adaptive isogeometric finite element analysis of steady-state groundwater flow", *International Journal for Numerical and Analytical Methods in Geomechanics*, vol. 40, no. 5, pp. 738–765, 2016, ISSN: 10969853. DOI: 10.1002/nag.2425. arXiv: nag.2347 [10.1002]. [Online]. Available: http://doi.wiley.com/10.1002/nag.2425.

[17] G. Lorenzo, M. Scott, K. Tew, T. Hughes, and H. Gomez, "Hierarchically refined and coarsened splines for moving interface problems, with particular application to phase-field models of prostate tumor growth", *Computer Methods in Applied Mechanics and Engineering*, vol. 319, pp. 515–548, 2017, ISSN: 00457825. DOI: 10.1016/j.cma.

2017.03.009. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0045782516318254.

[18] A.-V. Vuong, C. Giannelli, B. Jüttler, and B. Simeon, "A hierarchical approach to adaptive local refinement in isogeometric analysis", *Computer Methods in Applied Mechanics and Engineering*, vol. 200, no. 49-52, pp. 3554–3567, 2011, ISSN: 0045-7825. DOI: 10.1016/J.CMA.2011.09.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782511002933.

[19] D. Schillinger, J. A. Evans, A. Reali, M. A. Scott, and T. J. Hughes, "Isogeometric collocation: Cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations", *Computer Methods in Applied Mechanics and Engineering*, vol. 267, no. February, pp. 170–232, 2013, ISSN: 00457825. DOI: 10.1016/j.cma.2013.07.017.

[20] C. Manni, A. Reali, and H. Speleers, "Isogeometric collocation methods with generalized B-splines", in *Computers and Mathematics with Applications*, vol. 70, Pergamon, 2015, pp. 1659–1675. DOI: 10.1016/j.camwa.2015.03.027. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0898122115001443.

[21] A. Bellin and Y. Rubin, "Hydro_gen: A spatially distributed random field generator for correlated properties", *Stochastic Hydrology and Hydraulics*, vol. 10, no. 4, pp. 253–278, 1996, ISSN: 1435-151X. DOI: 10.1007/BF01581869. [Online]. Available: https://doi.org/10.1007/BF01581869.

[22] V. Cvetkovic, H. Cheng, and X. H. Wen, "Analysis of nonlinear effects on tracer migration in heterogeneous aquifers using lagrangian travel time statistics", *Water Resources Research*, vol. 32, no. 6, pp. 1671–1680, 1996. DOI: 10.1029/96WR00278. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/96WR00278. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/96WR00278.

[23] J.-R. de Dreuzy, A. Beaudoin, and J. Erhel, "Asymptotic dispersion in 2d heterogeneous porous media determined by parallel numerical simulations", *Water Resources Research*, vol. 43, no. 10, 2007. DOI: 10.1029/2006WR005394. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2006WR005394. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2006WR005394.

[24] H. Gotovac, V. Cvetković, and R. Andričević, "Adaptive Fup multi-resolution approach to flow and advective transport in highly heterogeneous porous media: Methodology, accuracy and convergence", *Advances in Water Resources*, vol. 32, no. 6, pp. 885–905, 2009, ISSN: 03091708. DOI: 10.1016/j.advwatres.2009.02.013.

[25] H. Gomez and L. D. Lorenzis, "The variational collocation method", *Computer Methods in Applied Mechanics and Engineering*, vol. 309, pp. 152 –181, 2016, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2016.06.003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782516305035.

[26] L. Malenica, H. Gotovac, G. Kamber, S. Simunovic, S. Allu, and V. Divic, "Groundwater flow modeling in karst aquifers: Coupling 3d matrix and 1d conduit flow via control volume isogeometric analysis—experimental verification with a 3d physical model", *Water*, vol. 10, no. 12, 2018, ISSN: 2073-4441. DOI: 10.3390/w10121787. [Online]. Available: http://www.mdpi.com/2073-4441/10/12/1787.

[27] D. Ford and P. D. Williams, *Karst hydrogeology and geomorphology*, 2007.

[28] M. Bakalowicz, "Karst groundwater: A challenge for new resources", *Hydrogeology Journal*, vol. 13, no. 1, pp. 148–160, 2005, ISSN: 14312174. DOI: 10.1007/s10040-004-0402-9. [Online]. Available: http://link.springer.com/10.1007/s10040-004-0402-9.

[29] N. Goldscheider and D. D. P. Drew, *Methods in karst hydrogeology*. Taylor & Francis, 2007, p. 264, ISBN: 0415428734.

[30] R. Ghasemizadeh, F. Hellweger, C. Butscher, I. Padilla, D. Vesper, M. Field, and A. Alshawabkeh, "Review: Groundwater flow and transport modeling of karst aquifers, with particular reference to the North Coast Limestone aquifer system of Puerto Rico", *Hydrogeology Journal*, vol. 20, no. 8, pp. 1441–1461, 2012, ISSN: 1431-2174. DOI: 10.1007/s10040-012-0897-4. arXiv: NIHMS150003. [Online]. Available: http://link.springer.com/10.1007/s10040-012-0897-4.

[31] D. Jukić and V. Denić-Jukić, "Groundwater balance estimation in karst by using a conceptual rainfall–runoff model", *Journal of Hydrology*, vol. 373, no. 3-4, pp. 302–315, 2009, ISSN: 0022-1694. DOI: 10.1016/J.JHYDROL.2009.04.035. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022169409002868.

[32] L. Xing, L. Huang, G. Chi, L. Yang, C. Li, X. Hou, L. Xing, L. Huang, G. Chi, L. Yang, C. Li, and X. Hou, "A Dynamic Study of a Karst Spring Based on Wavelet Analysis and the Mann-Kendall Trend Test", *Water*, vol. 10, no. 6, p. 698, 2018, ISSN: 2073-4441. DOI: 10.3390/w10060698. [Online]. Available: http://www.mdpi.com/2073-4441/10/6/698.

[33] N. Dvory, A. Ronen, Y. Livshitz, E. Adar, M. Kuznetsov, A. Yakirevich, N. Z. Dvory, A. Ronen, Y. Livshitz, E. Adar, M. Kuznetsov, and A. Yakirevich, "Quantification of Groundwater Recharge from an Ephemeral Stream into a Mountainous Karst Aquifer", *Water*, vol. 10, no. 1, p. 79, 2018, ISSN: 2073-4441. DOI: 10.3390/w10010079. [Online]. Available: http://www.mdpi.com/2073-4441/10/1/79.

[34] A. Kovács and M. Sauter, "Modelling karst hydrodynamics", *Methods in Karst Hydrogeology*, no. November, pp. 201–222, 2007. DOI: 10.1201/9781482266023-16. [Online]. Available: https://www.taylorfrancis.com/books/e/9781482266023/chapters/10.1201{\%}2F9781482266023-16Biblioth{\'{e}}que{\%}5CnISSKA{\%}5Cn429.

[35] A. Hartmann, N. Goldscheider, T. Wagener, J. Lange, and M. Weiler, *Karst water resources in a changing world: Review of hydrological modeling approaches*, 2014. DOI: 10.1002/2013RG000443. [Online]. Available: http://doi.wiley.com/10.1002/2013RG000443.

[36] R. Liedl, M. Sauter, D. Hückinghaus, T. Clemens, and G. Teutsch, "Simulation of the development of karst aquifers using a coupled continuum pipe flow model", *Water Resources Research*, vol. 39, no. 3, 2003, ISSN: 00431397. DOI: 10.1029/2001WR001206. [Online]. Available: http://doi.wiley.com/10.1029/2001WR001206.

[37] L. Kiraly, *Modelling karst aquifers by the combined discrete channel and continuum approach*, 1998. DOI: PNR61. [Online]. Available: http://doc.rero.ch/record/260628.

[38] M. G. Király, "Remarques sur hydrogramme des sources karstiques simulé par modèles mathématiques (Remarks on karst springs hydrographs simulated with mathematical models).", *Bulletin d'Hydrogéologie de l'Université de Neuchâtel.*, 1:37–60. 1976.

[39] L. Kiraly, "FEM 301-a three dimensional model for groundwater flow simulation", Tech. Rep., 1985. [Online]. Available: https://inis.iaea.org/search/search.aspx?orig{_}q=RN:18068625.

[40] T. Clemens, D. Hückinghaus, R. Liedl, and M. Sauter, "Simulation of the development of karst aquifers: Role of the epikarst", *International Journal of Earth Sciences*, vol. 88, no. 1, pp. 157–162, 1999, ISSN: 14373254. DOI: 10.1007/s005310050252. [Online]. Available: http://link.springer.com/10.1007/s005310050252.

[41] S. Bauer, R. Liedl, and M. Sauter, "Modelling of karst development considering conduit-matrix exchange flow.", *Calibration and reliability in groundwater modelling: coping with uncertainty. Proceedings of the ModelCARE'99 conference held in Zurich, Switzerland, 20-23 September 1999*, pp. 10–15, 2000. [Online]. Available: https://www.cabdirect.org/cabdirect/abstract/20013045602.

[42] S. Bauer, R. Liedl, and M. Sauter, "Modeling of karst aquifer genesis: Influence of exchange flow", *Water Resources Research*, vol. 39, no. 10, 2003, ISSN: 00431397. DOI: 10.1029/2003WR002218. [Online]. Available: http://doi.wiley.com/10.1029/2003WR002218.

[43] S. Birk, R. Liedl, M. Sauter, and G. Teutsch, "Hydraulic boundary conditions as a controlling factor in karst genesis: A numerical modeling study on artesian conduit development in gypsum", *Water Resources Research*, vol. 39, no. 1, SBH 2–1–SBH 2–14, 2003, ISSN: 00431397. DOI: 10.1029/2002WR001308. [Online]. Available: http://doi.wiley.com/10.1029/2002WR001308.

[44] T. Reimann, T. Geyer, W. B. Shoemaker, R. Liedl, and M. Sauter, "Effects of dynamically variable saturation and matrix-conduit coupling of flow in karst aquifers", *Water Resources Research*, vol. 47, no. 11, 2011, ISSN: 00431397. DOI: 10.1029/2011WR010446. [Online]. Available: http://doi.wiley.com/10.1029/2011WR010446.

[45] T. Reimann, M. Giese, T. Geyer, R. Liedl, J. C. Maréchal, and W. B. Shoemaker, "Representation of water abstraction from a karst conduit with numerical discrete-continuum models", *Hydrology and Earth System Sciences*, vol. 18, no. 1, pp. 227–241, 2014, ISSN: 10275606. DOI: 10.5194/hess-18-227-2014. [Online]. Available: https://www.hydrol-earth-syst-sci.net/18/227/2014/.

[46] M. Giese, T. Reimann, V. Bailly-Comte, J.-C. Maréchal, M. Sauter, and T. Geyer, "Turbulent and Laminar Flow in Karst Conduits Under Unsteady Flow Conditions: Interpretation of Pumping Tests by Discrete Conduit-Continuum Modeling", *Water Resources Research*, vol. 54, no. 3, pp. 1918–1933, 2018, ISSN: 00431397. DOI: 10.1002/

2017WR020658. [Online]. Available: http://doi.wiley.com/10.1002/2017WR020658.

[47] Y. Xue, T. Teng, L. Zhu, M. He, J. Ren, X. Dong, F. Liu, Y. Xue, T. Teng, L. Zhu, M. He, J. Ren, X. Dong, and F. Liu, "Evaluation of the Non-Darcy Effect of Water Inrush from Karst Collapse Columns by Means of a Nonlinear Flow Model", *Water*, vol. 10, no. 9, p. 1234, 2018, ISSN: 2073-4441. DOI: 10.3390/w10091234. [Online]. Available: http://www.mdpi.com/2073-4441/10/9/1234.

[48] W. B. Shoemaker, E. L. Kuniansky, S. Birk, S. Bauer, and E. D. Swain, "Documentation of a Conduit Flow Process (CFP) for MODFLOW-2005", Tech. Rep., 2005, p. 50. [Online]. Available: https://pubs.usgs.gov/tm/tm6a24/pdf/tm6-A24.pdf.

[49] A. W. Harbaugh, *MODFLOW-2005, The U.S. Geological Survey Modular Ground-Water Model—the Ground-Water Flow Process*, 2005. [Online]. Available: http://water.usgs.gov/software/ground{_}water.html/..

[50] M. A. Celia, E. T. Bouloutas, and R. L. Zarba, *A general mass conservative numerical solution for the unsaturated flow equation*, 1990. DOI: 10.1029/WR026i007p01483.

[51] R. de Rooij, "Towards improved numerical modeling of karst aquifers : coupling turbulent conduit flow and laminar matrix flow under variably saturated conditions", no. December 2007, p. 130, 2008. DOI: MilandreOFEV,PNR61.

[52] R. de Rooij, P. Perrochet, and W. Graham, "From rainfall to spring discharge: Coupling conduit flow, subsurface matrix flow and surface flow in karst systems using a discrete–continuum model", vol. 61, 29–41, Nov. 2013.

[53] Öllős and Németh, "Szakvélemény a repedezett kőzetekben lejátszódó folyamatok kisminta vizsgálatáról (Study of the physical modelling of the flow processes in fractured rock)", *Technical University Budapest*, 1960.

[54] Y. Wu and D. Hunkeler, "Hyporheic exchange in a karst conduit and sediment system - A laboratory analog study", *Journal of Hydrology*, vol. 501, pp. 125–132, 2013, ISSN: 00221694. DOI: 10.1016/j.jhydrol.2013.07.040. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022169413005672.

[55] J. Faulkner, B. X. Hu, S. Kish, and F. Hua, "Laboratory analog and numerical study of groundwater flow and solute transport in a karst aquifer with conduit and matrix domains", *Journal of Contaminant Hydrology*, vol. 110, no. 1-2, pp. 34–44, 2009, ISSN: 01697722. DOI: 10.1016/j.jconhyd.2009.08.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169772209001107.

[56] R. B. P. Castro, *Statistical Analysis of Karst Aquifer Pollution, Karst Flow Model Validation at Laboratory Scale, and Development of Seepage Meter*, 2017. [Online]. Available: https://fsu.digital.flvc.org/islandora/object/fsu{\%}3A552112.

[57] A. Furman, "Modeling Coupled Surface–Subsurface Flow Processes: A Review", *Vadose Zone Journal*, vol. 7, no. 2, p. 741, 2008, ISSN: 1539-1663. DOI: 10.2136/vzj2007.0065. [Online]. Available: https://www.soils.org/publications/vzj/abstracts/7/2/741.

[58] T. P. Clement, W. R. Wise, and F. J. Molz, "A physically based, two-dimensional, finite-difference algorithm for modeling variably saturated flow", *Journal of Hydrology*, vol. 161, no. 1-4, pp. 71–90, 1994, ISSN: 00221694. DOI: 10.1016/0022-1694(94)90121-X.

[59] A. Szymkiewicz, "Modelling Water Flow in Unsaturated Porous Media: Accounting for Nonlinear Permeability and Material Heterogeneity", *GeoPlanet: Earth and Planetary Sciences*, GeoPlanet: Earth and Planetary Sciences, vol. 9, p. 1, 2013, ISSN: 21905193. DOI: 10.1007/978-3-642-23559-7. [Online]. Available: http://link.springer.com/10.1007/978-3-642-23559-7.

[60] P. Y. Jeannin, "Modeling flow in phreatic and epiphreatic karst conduits in the Hölloch cave (Muotatal, Switzerland)", *Water Resources Research*, vol. 37, no. 2, pp. 191–200, 2001, ISSN: 00431397. DOI: 10.1029/2000WR900257. [Online]. Available: http://doi.wiley.com/10.1029/2000WR900257.

[61] W. B. White, "Karst hydrology: Recent developments and open questions", *Engineering Geology*, vol. 65, no. 2-3, pp. 85–105, 2002, ISSN: 00137952. DOI: 10.1016/S0013-7952(01)00116-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0013795201001168.

[62] V. Jovic, "Analysis and Modelling of Non-Steady Flow in Pipe", 2013. [Online]. Available: https://www.wiley.com/en-us/Analysis+and+Modelling+of+Non+Steady+Flow+in+Pipe+and+Channel+Networks-p-9781118532140.

[63] R. Szymkiewicz, *Numerical modeling in open channel hydraulics*. Springer, 2010, p. 419, ISBN: 9048136733.

[64] E. G. Lappala, R. W. Healy, and E. P. Weeks, "Documentation of Computer Program VS2D to Solve the Equations of Fluid Flow in Variably Saturated Porous Media - Water-Resources Investigations Report 83-4099", *Water-Resources Investigations Report*, p. 193, 1987. [Online]. Available: https://pubs.er.usgs.gov/publication/wri834099.

[65] W. Durner, "Hydraulic conductivity estimation for soils with heterogeneous pore structure", *Water Resources Research*, vol. 30, no. 2, pp. 211–223, 1994, ISSN: 19447973. DOI: 10.1029/93WR02676. [Online]. Available: http://doi.wiley.com/10.1029/93WR02676.

[66] E. Priesack and W. Durner, "Closed-Form Expression for the Multi-Modal Unsaturated Conductivity Function", *Vadose Zone Journal*, vol. 5, no. 1, p. 121, 2006, ISSN: 1539-1663. DOI: 10.2136/vzj2005.0066. [Online]. Available: https://www.soils.org/publications/vzj/abstracts/5/1/121.

[67] M. T. van Genuchten, "A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils1", *Soil Science Society of America Journal*, vol. 44, no. 5, p. 892, 1980, ISSN: 0361-5995. DOI: 10.2136/sssaj1980.03615995004400050002x. [Online]. Available: https://www.soils.org/publications/sssaj/abstracts/44/5/SS0440050892.

[68] D. M. Bjerklie, S. L. Dingman, and C. H. Bolster, "Comparison of constitutive flow resistance equations based on the Manning and Chezy equations applied to natural rivers", *Water Resources Research*, vol. 41, no. 11, 2005, ISSN: 1944-7973. DOI: 10.1029/2004WR003776. [Online]. Available: http:https://doi.org/10.1029/2004WR003776.

[69] A. M. Jaeschke, "Isogeometric Analysis for Compressible Flows with Application in Turbomachinery", *MSc Thesis, TU Delft*, 2015. [Online]. Available: http://repository.tudelft.nl/..

[70] M. Vauclin, D. Khanji, and G. Vachaud, "Experimental and numerical study of a transient, two-dimensional unsaturated-saturated water table recharge problem", *Water Resources Research*, vol. 15, no. 5, pp. 1089–1101, 1979, ISSN: 19447973. DOI: 10.1029/WR015i005p01089. [Online]. Available: http://doi.wiley.com/10.1029/WR015i005p01089.

[71] L. A. Rossman, "Storm Water Management Model Quality Assurance Report: Dynamic Wave Flow Routing", *Storm Water Management Model Quality Assurance Report*, no. EPA/600/R-06/097, pp. 1–115, 2006. [Online]. Available: https://cfpub.epa.gov/si/si{_}public{_}record{_}report.cfm?dirEntryId=158255http://www.epa.gov/water-research/storm-water-management-model-swmm.

[72] L. Rossman, "Storm Water Management Model Reference Manual Volume II – Hydraulics", Environmental Protection Agency, EPA/600/R-17/111, 2017., Tech. Rep., 2017. [Online]. Available: https://nepis.epa.gov/Exe/ZyPDF.cgi?Dockey=P100S9AS.pdf.

[73] L. Dalcin, N. Collier, P. Vignal, A. Côrtes, and V. Calo, "PetIGA: A framework for high-performance isogeometric analysis", *Computer Methods in Applied Mechanics and Engineering*, vol. 308, pp. 151–181, 2016, ISSN: 0045-7825. DOI: 10.1016/J.CMA.2016.05.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045782516303401.

[74] S. Balay and J. Brown, "{PETS}c {U}sers {M}anual", Tech. Rep. ANL-95/11 - Revision 3.3, 2012. [Online]. Available: www.anl.gov.

[75] O. Bonacci, *Karst Hydrology*, ser. Springer Series in Physical Environment. Berlin, Heidelberg: Springer Berlin Heidelberg, 1987, vol. 2, p. 184, ISBN: 978-3-642-83167-6. DOI: 10.1007/978-3-642-83165-2. [Online]. Available: http://link.springer.com/10.1007/978-3-642-83165-2.

[76] A. Brandt, "Multi-Level Adaptive Solutions to Boundary-Value Problems", *Mathematics of Computation*, vol. 31, no. 138, p. 333, 1977, ISSN: 00255718. DOI: 10.2307/2006422. [Online]. Available: http://www.jstor.org/stable/2006422?origin=crossref.

[77] I Babushka and W. C. Rheinboldt, "Error Estimates for Adaptive Finite Element Computation", *SIAM J. Num. Anal.*, vol. 15, no. 4, pp. 736–754, 1978.

[78] L. Demkowicz, P. Devloo, and J. Oden, "On an h-type mesh-refinement strategy based on minimization of interpolation errors", *Computer Methods in Applied Mechanics and Engineering*, vol. 53, no. 1, pp. 67–89, 1985, ISSN: 0045-7825. DOI: 10.1016/0045-7825(85)90076-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0045782585900763.

[79] M. J. Berger and P. Colella, "Local adaptive mesh refinement for shock hydrodynamics", *Journal of Computational Physics*, vol. 82, no. 1, pp. 64–84, 1989, ISSN: 10902716. DOI: 10.1016/0021-9991(89)90035-1. arXiv: 1607.02212.

[80] L. Demkowicz, J. Oden, W. Rachowicz, and O. Hardy, "Toward a universal h-p adaptive finite element strategy, part 1. Constrained approximation and data structure", *Computer Methods in Applied Mechanics and Engineering*, vol. 77, no. 1-2, pp. 79–112, 1989, ISSN: 0045-7825. DOI: 10.1016/0045-7825(89)90129-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0045782589901291.

[81] J. Oden, L. Demkowicz, W. Rachowicz, and T. Westermann, "Toward a universal h-p adaptive finite element strategy, part 2. A posteriori error estimation", *Computer Methods in Applied Mechanics and Engineering*, vol. 77, no. 1-2, pp. 113–180, 1989, ISSN: 0045-7825. DOI: 10.1016/0045-7825(89)90130-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0045782589901308.

[82] W. Rachowicz, J. Oden, and L. Demkowicz, "Toward a universal h-p adaptive finite element strategy part 3. design of h-p meshes", *Computer Methods in Applied Mechanics and Engineering*, vol. 77, no. 1-2, pp. 181–212, 1989, ISSN: 0045-7825. DOI: 10.1016/0045-7825(89)90131-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/004578258990131X.

[83] A. Harten, *Adaptive Multiresolution Schemes for Shock Computations*, 1994. DOI: 10.1006/jcph.1994.1199. [Online]. Available: http://dx.doi.org/10.1006/jcph.1994.1199.

[84] A. Harten, "Multiresolution algorithms for the numerical solution of hyperbolic conservation laws", *Communications on Pure and Applied Mathematics*, vol. 48, no. 12, pp. 1305–1342, 1995, ISSN: 10970312. DOI: 10.1002/cpa.3160481201.

[85] S. Bertoluzza and G. Naldi, "A Wavelet Collocation Method for the Numerical Solution of Partial Differential Equations", *Applied and Computational Harmonic Analysis*, vol. 3, no. 1, pp. 1–9, 1996, ISSN: 10635203. DOI: 10.1006/acha.1996.0001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1063520396900019.

[86] M. Holmström, "Solving Hyperbolic PDEs Using Interpolating Wavelets", *SIAM Journal on Scientific Computing*, vol. 21, no. 2, pp. 405–420, 1999, ISSN: 1064-8275. DOI: 10.1137/S1064827597316278. [Online]. Available: http://epubs.siam.org/doi/10.1137/S1064827597316278http://epubs.siam.org/doi/abs/10.1137/S1064827597316278.

[87] A. Nejadmalayeri, A. Vezolainen, E. Brown-Dymkoski, and O. V. Vasilyev, "Parallel adaptive wavelet collocation method for PDEs", *Journal of Computational Physics*, vol. 298, pp. 237–253, 2015, ISSN: 10902716. DOI: 10.1016/j.jcp.2015.05.028. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999115003629.

[88] O. V. Vasilyev and S. Paolucci, "A Fast Adaptive Wavelet Collocation Algorithm for Multidimensional PDEs", *Journal of Computational Physics*, vol. 138, no. 1, pp. 16–56, 1997, ISSN: 00219991. DOI: 10.1006/jcph.1997.5814. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0021999197958149.

[89] O. V. Vasilyev and C. Bowman, "Second-Generation Wavelet Collocation Method for the Solution of Partial Differential Equations", *Journal of Computational Physics*, vol. 165, no. 2, pp. 660–693, 2000, ISSN: 00219991. DOI: 10.1006/jcph.2000.6638. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0021999100966385.

[90] R. Wang, P. Keast, and P. Muir, "A high-order global spatially adaptive collocation method for 1-D parabolic PDEs", *Applied Numerical Mathematics*, vol. 50, no. 2, pp. 239–260, 2004, ISSN: 01689274. DOI: 10.1016/j.apnum.2003.12.023. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168927403002344.

[91] K. Alhumaizi, "A moving collocation method for the solution of the transient convection–diffusion-reaction problems", *Journal of Computational and Applied Mathematics*, vol. 193, no. 2, pp. 484–496, 2006,

ISSN: 0377-0427. DOI: 10.1016/J.CAM.2005.03.079. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377042705004279.

[92] V. Cvetkovic and H. Gotovac, "On the upscaling of chemical transport in fractured rock", *Water Resources Research*, vol. 50, no. 7, pp. 5797–5816, 2014, ISSN: 19447973. DOI: 10.1002/2014WR015505. [Online]. Available: http://doi.wiley.com/10.1002/2014WR015505.

[93] A. Fiori, A. Zarlenga, H. Gotovac, I. Jankovic, E. Volpi, V. Cvetkovic, and G. Dagan, "Advective transport in heterogeneous aquifers: Are proxy models predictive?", *Water Resources Research*, vol. 51, no. 12, pp. 9577–9594, 2015, ISSN: 00431397. DOI: 10.1002/2015WR017118. [Online]. Available: http://doi.wiley.com/10.1002/2015WR017118.

[94] S. Osher and R. Sanders, "Numerical Approximations to Nonlinear Conservation Laws With Locally Varying Time and Space Grids", *Mathematics of Computation*, vol. 41, no. 164, pp. 321–336, 1983, ISSN: 0025-5718. DOI: 10.1090/S0025-5718-1983-0717689-8. [Online]. Available: http://www.ams.org/journals/mcom/1983-41-164/S0025-5718-1983-0717689-8/S0025-5718-1983-0717689-8.pdf{\%}5Cnhttp://dx.doi.org/10.1090/S0025-5718-1983-0717689-8-Seemoreat:http://www.ams.org/journals/mcom/1983-41-164/S0025-5718-1983-0717689-8/{\#}sthash.jwxmj.

[95] M. J. Berger and O. Joseph, "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations", *Journal of Computational Physics*, vol. 53, no. 3, pp. 484–512, 1984.

[96] S. Müller and Y. Stiriba, "Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping", *Journal of Scientific Computing*, vol. 30, no. 3, pp. 493–531, 2007, ISSN: 08857474. DOI: 10.1007/s10915-006-9102-z. [Online]. Available: http://link.springer.com/10.1007/s10915-006-9102-z.

[97] E. M. Constantinescu and A. Sandu, "Multirate timestepping methods for hyperbolic conservation laws", *Journal of Scientific Computing*, vol. 33, no. 3, pp. 239–278, 2007, ISSN: 08857474. DOI: 10.1007/s10915-007-9151-y.

[98] A. Sandu and E. M. Constantinescu, "Multirate explicit adams methods for time integration of conservation laws", *Journal of Scientific Computing*, vol. 38, no. 2, pp. 229–249, 2009, ISSN: 08857474. DOI: 10.1007/s10915-008-9235-3.

[99] M. O. Domingues, S. M. Gomes, O. Roussel, and K. Schneider, "An adaptive multiresolution scheme with local time stepping for evolutionary PDEs", *Journal of Computational Physics*, vol. 227, no. 8, pp. 3758–3780, 2008, ISSN: 00219991. DOI: 10.1016/j.jcp.2007.11.046.

[100] M. O. Domingues, S. M. Gomes, O. Roussel, and K. Schneider, "Space-time adaptive multiresolution methods for hyperbolic conservation laws: Applications to compressible Euler equations", *Applied Numerical Mathematics*, vol. 59, no. 9, pp. 2303–2321, 2009, ISSN: 01689274. DOI: 10.1016/j.apnum.2008.12.018. [Online]. Available: http://dx.doi.org/10.1016/j.apnum.2008.12.018.

[101] L. Krivodonova, "An efficient local time-stepping scheme for solution of nonlinear conservation laws", *Journal of Computational Physics*, vol. 229, no. 22, pp. 8537–8551, 2010, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2010.07.037. [Online]. Available: http://dx.doi.org/10.1016/j.jcp.2010.07.037.

[102] L. Lapidus and F. G Pinder, *Numerical Solution of Partial Differential Equations in Science and Engineering*, 4. Wiley, 1983, vol. 64, p. 49, ISBN: 0471098663. DOI: 10.1029/EO064i004p00026-01.

[103] T. J. Hughes, "Unconditionally stable algorithms for nonlinear heat conduction", *Computer Methods in Applied Mechanics and Engineering*, vol. 10, no. 2, pp. 135–139, 1977, ISSN: 0045-7825. DOI: 10.1016/0045-7825(77)90001-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0045782577900019.

[104] R. J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations*. Society for Industrial and Applied Mathematics, 2007, pp. 1–144, ISBN: 978-0-89871-629-0. DOI: 10.1137/1.9780898717839.

[105] O. V. Vasilyev, S. Paolucci, and M. Sen, "A Multilevel Wavelet Collocation Method for Solving Partial Differential Equations in a Finite Domain", *Journal of Computational Physics*, vol. 120, no. 1, pp. 33–47, 1995, ISSN: 00219991. DOI: 10.1006/jcph.1995.1147. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999185711473http://linkinghub.elsevier.com/retrieve/pii/S0021999185711473.

[106] A. Cohen, S. M. Kaber, S. Müller, and M. Postel, "Fully adaptive multiresolution finite volume schemes for conservation laws", *Mathematics of Computation*, vol. 72, no. 241, pp. 183–226, 2001, ISSN: 0025-5718. DOI: 10.1090/S0025-5718-01-01391-6.

[107] S. H. Lee and S. S. Hsieh, "Expedient implicit integration with adaptive time stepping algorithm for nonlinear transient analysis", *Computer Methods in Applied Mechanics and Engineering*, vol. 81, no. 2, pp. 151–172, 1990, ISSN: 0045-7825. DOI: 10.1016/0045-7825(90)90107-W. [Online]. Available: https://www.sciencedirect.com/science/article/pii/004578259090107W.

[108] D. Kavetski, P. Binning, and S. W. Sloan, "Adaptive backward Euler time stepping with truncation error control for numerical modelling of unsaturated fluid flow", *International Journal for Numerical Methods in Engineering*, vol. 53, no. 6, pp. 1301–1322, 2002, ISSN: 00295981. DOI: 10.1002/nme.329.

[109] M. O. Domingues, O Roussel, and K Schneider, "An adaptive multiresolution method for parabolic PDEs with time-step control", *International Journal for Numerical Methods in Engineering*, vol. 78, no. 6, pp. 652–670, 2009, ISSN: 00295981. DOI: 10.1002/nme.2501. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.540.8264{\&}rep=rep1{\&}type=pdf.

[110] A. Ogata and R. B. Banks, "A solution of the differential equation of longitudinal dispersion in porous media", *Geological Survey (U.S.).; Professional paper*, A1–A7, 1961, ISSN: 2330-7102. [Online]. Available: http://pubs.er.usgs.gov/publication/pp411A.

[111] J. M. Alam, N. K. Kevlahan, and O. V. Vasilyev, "Simultaneous space-time adaptive wavelet solution of nonlinear parabolic differential equations", *Journal of Computational Physics*, vol. 214, no. 2, pp. 829–857, 2006, ISSN: 00219991. DOI: 10.1016/j.jcp.2005.10.009.

[112] H. Gotovac, V. Kozulić, and B. Gotovac, "Space-Time Adaptive Fup Multi-Resolution Approach for Boundary-Initial Value Problems", *Computers, materials & continua*, vol. 15, no. 3, pp. 173–198, 2010. DOI: 10.3970/CMC.2010.015.173. [Online]. Available: https://www.bib.irb.hr/476414.

[113] M. R. Kirkland, R. G. Hills, and P. J. Wierenga, "Algorithms for solving Richards' equation for variably saturated soils", *Water Resources Research*, vol. 28, no. 8, pp. 2049–2058, 1992, ISSN: 19447973. DOI: 10.1029/92WR00802. [Online]. Available: http://doi.wiley.com/10.1029/92WR00802.

[114] H. B. Fischer, E. J. List, R. C. Y. Koh, J Imberger, and N. H. Brooks, "Mixing in inland and coastal waters, Academic, New York", p. 483, 1979.

[115] D. L. Stefanovic and H. G. Stefan, "Two-Dimensional Water Quality Model for Unsteady Advection-Diffusion of Nonconservative Substances", 1997. [Online]. Available: https://conservancy.umn.edu/handle/11299/112965.

[116] X. Wei, Y. Zhang, T. J. Hughes, and M. A. Scott, "Truncated hierarchical catmull–clark subdivision with local refinement", *Computer Methods in Applied Mechanics and Engineering*, vol. 291, pp. 1 –20, 2015, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2015.03.019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045782515001292.

[117] G. Kamber, H. Gotovac, V. Kozulić, L. Malenica, and B. Gotovac, "Adaptive isogeometric analysis based on truncated hierarchical basis functions", *Book of Abstracts of the 9th International Congress of Croatian Society of Mechanics*, p. 3, 2019. [Online]. Available: https://www.bib.irb.hr/980767.