

UNIVERSITY OF WEST BOHEMIA

FACULTY OF APPLIED SCIENCES

DEPARTMENT OF MECHANICS



Numerical Simulations of the Shallow Water Flow

Ph.D. thesis

submitted in partial fulfilment of
the requirements for the degree of
Doctor of Philosophy
in Applied Mechanics

Author:
Ing. Martin FIŠER

Supervisor:
Doc. Ing. Jan VIMMR, Ph.D.

Pilsen, 2016

ZÁPADOČESKÁ UNIVERZITA V PLZNI

FAKULTA APLIKOVANÝCH VĚD

KATEDRA MECHANIKY



Numerické simulace proudění mělkých vod

Dizertační práce

vypracovaná v rámci
požadavků na obdržení titulu
Ph.D. v oboru aplikovaná mechanika

Autor:
Ing. Martin FIŠER

Vedoucí práce:
Doc. Ing. Jan VIMMR, Ph.D.

Plzeň, 2016

Declaration

I declare that I have elaborated this thesis alone and all the literature which was used is written at the end of this work.

Ing. Martin Fišer
Pilsen, 2016

The Ph.D. thesis summarises the research work carried out by the author at the University of West Bohemia, which was partially supported by the project LO1506 of the Ministry of Education, Youth and Sports of Czech Republic.

Acknowledgement

Many thanks to my family for their support and patience and also many thanks to my thesis supervisor Doc. Ing. Jan Vimmr, Ph.D. and advisor Ing. Ondřej Bublík, Ph.D. for their help. And last but not least many thanks to all designers of Linux system.

Abstract

This work describes the numerical solution of the mathematical model of the Shallow Water Equations. As the source terms the bed slope source term and the bed friction term are included.

At the beginning the mathematical model and the boundary conditions are described. Mathematical model is solved by the finite volume and discontinuous Galerkin method. Unstructured mesh was used for the discretization of the computational area. The numerical flux is solved by the HLL and Roe approximative Riemann solver. Validation of described methods is at the end of this thesis.

This work is focused on the schemes providing 'C-property' condition, higher order of accuracy and schemes which can cope with wet/dry interface.

The work brings novel scheme for computing wet/dry interface within finite volume method and novel limiting process used for limiting of discontinuous Galerkin method.

Key words: bed slope source term, bed friction term, C-property condition, HLL scheme, finite volume method, discontinuous Galerkin method, limiting process, artificial viscosity, minmod limiter, linear reconstruction, Roe solver of Riemann problem, Saint-Venant equations, time integration, unstructured mesh

Abstrakt

Tato práce pojednává o numerickém řešení matematického modelu mělkých vod s uvažováním zdrojového členu od nerovnosti dna a zdrojového členu tření o dno.

Práci uvádí popis matematického modelu proudění mělkých vod a okrajových podmínek. Matematický model je řešen metodou konečných objemů a nespojitou Galerkinovou metodou. Výpočetní oblast byla diskretizována za pomoci nestrukturovných sítí. Numerický tok je simulován pomocí HLL a Roe schématu. V závěru práce je validován numerický řešič, který byl vyvinut na základě poznatků zde shrnutých.

Speciální pozornost je věnována schématům, která mají vyvážený numerickým tok a zdrojový člen dna, dále pak schématům s vyšším řádem přesnosti a simulacím na rozhraní zatopené a nezatopené oblasti.

Hlavní přínos práce je v nové metodě výpočtu zdrojového členu dna na rozhraní zatopené a nezatopené oblasti v rámci metody konečných objemů a nového limitovacího procesu aplikovatelného v rámci nespojité galerkinovy metody.

Klíčová slova: Saint-Venantovy rovnice, tření o dno, vyvážený zdrojový člen dna, HLL schéma, metoda konečných objemů, nespojitá Galerkinova metoda, limitovací proces, umělá viskozita, minmod limiter, lineární rekonstrukce, časová integrace, nestrukturované sítě

Contents

Declaration	1
1 Mathematical Model	14
1.1 Mathematical Features	15
1.2 Boundary Conditions	17
2 Finite Volume Method	19
2.1 Derivation of the FVM	19
2.2 Numerical Approximation of Inviscid Flux	21
2.2.1 HLL Scheme	21
2.2.2 Roe's Scheme	25
2.2.3 Entropy Correction	30
2.2.4 Wet/Dry Interface Treatment	31
2.3 Bed Slope Source Term Approximation	33
2.3.1 Cell-Centred Approximation	33
2.3.2 Bed Function Evaluated at the Edges	37
2.4 Bed Friction Source Term	41
2.4.1 Chezy Formulation	41
2.4.2 Manning's Formulation	42
2.4.3 Bed Friction Source Term Discretization	42
2.5 Linear Reconstruction	43
3 Discontinuous Galerkin Method	46
3.1 Derivation of the Method	46
3.2 Numerical Scheme of Discontinuous Galerkin Method in 1D	47
3.2.1 Gaussian Quadrature for Curve Integrals	50
3.2.2 Limiting Process	50
3.2.3 Bed Slope Source Term	54
3.2.4 Bed Friction Source Term	55
3.3 Numerical scheme of Discontinuous Galerkin Method in 2D	55
3.3.1 Gaussian Quadrature for 2D Triangular Elements	57
3.3.2 Limiting Process	59
3.3.3 Bed Slope Source Term	62
3.3.4 Bed Friction Source Term	64

4	Time Integration	67
4.1	Euler's Method	67
4.1.1	The Principle of Euler's Method	67
4.1.2	Application of the Euler's Method	68
4.2	Second-Order Runge-Kutta Method	68
4.2.1	Principle of the Second-Order Runge-Kutta Method	68
4.2.2	Application of the Second-Order Runge-Kutta Method	68
4.3	Fourth-Order Runge-Kutta Method	69
4.3.1	Application of the Fourth-Order Runge-Kutta Method	69
4.4	Necessary Condition of Convergence	69
5	Unstructured Meshes	71
5.1	Input data	71
5.2	EP Matrix	72
5.3	TE , ET and ETyp Matrices	73
5.4	Enx , Eny , Edl and Tzn Matrices	74
5.5	Implementation of the Unstructured-Mesh Algorithm	74
6	Parallelisation of the Code	76
7	Results	81
7.1	Comparisons with Analytical Solutions	81
7.1.1	2D Dam-break	81
7.1.2	Flow over the Bump	86
7.1.3	Planar Surface in Parabolic Bowl without Friction	93
7.2	Comparison with Experiments	94
7.2.1	Dam Break Over a Triangular Bottom Sill	98
7.2.2	H-dam Experiment	99
7.2.3	Obstacle Experiment	101

Nomenclature

The overview of the most frequent variables and constants.

- $a_{i+\frac{1}{2}}^{\pm}$... local wave propagation speed
- \mathbf{A} ... Roe's approximate matrix
- B ... function of the bottom topography
- \tilde{B} ... the function approximating bottom topography B
- c ... wave velocity
- C_f ... Manning's bed friction coefficient
- C_s ... Chezy bed friction coefficient
- d ... water surface level distance from arbitrarily chosen constant level
- \mathbf{F} ... vector function of flux in x-axis direction
- \mathbf{F}_i ... value of \mathbf{F} within finite volume Ω_i
- F_r ... Froude number
- g ... gravity
- \mathbf{G} ... vector function of flux in Y-axis direction
- \mathbf{G}_i ... value of \mathbf{G} within finite volume Ω_i
- h ... water level depth
- \mathbf{I} ... identity matrix
- \mathbf{J} ... Jacobi matrix
- K_s ... effective bed roughness height
- $|l_k|$... length of k^{th} edge
- n ... Manning's bed roughness factor
- n_b ... number of basis functions describing the solution
- $n_{b,B}$... number of basis functions describing bed function
- \vec{n}_k ... normal vector perpendicular to n -th edge of a finite volume
- n_t ... number of finite elements/volumes
- n_e ... number of edges
- \mathbf{P} ... rotation matrix
- $\mathbf{R}_{i\pm\frac{1}{2}}^{\pm}$... summand in bed slope source term, cell centered approach
- \mathbf{S}_B ... bed slope source term caused by the bottom irregularities
- \mathbf{S}_{B_i} ... approximation of the bed slope source term in the finite volume Ω_i
- \mathbf{S}_f ... bottom-friction source term
- \mathbf{S}_{f_i} ... approximation of the bed friction source term in the finite volume Ω_i
- t ... time
- t^n ... n -th time level
- u ... Cartesian component of velocity in x-axis direction
- v ... Cartesian component of velocity in y-axis direction
- w_i^j ... time dependent coefficient of the linear combination
- \mathbf{W} ... vector of conservative variables
- \mathbf{W}_0 ... initial conditions of \mathbf{W}
- \mathbf{x} ... eigenvector of matrix \mathbf{J}
- \mathbf{X} ... modal matrix assembled by eigenvectors of Jacobi matrix \mathbf{J}

- α ... attack angle at the inlet.
 Γ_i ... numerical approximation of inviscid flux in y-axis direction at the i -th edge
 Δt ... time step
 Δx ... cell width in x-axis direction
 Δy ... cell width in y-axis direction
 ϵ_v ... small coefficient used for computations of velocity
 ϵ ... limit used for decision whether a finite volume is flooded or not
 $\iota_{1/2}$... special rectangular domains in time-space plane
 λ ... eigenvalue of the matrix \mathbf{J}
 Λ^+ ... diagonal matrix containing all positive eigenvalues of matrix \mathbf{J} , zeros respectively
 Λ^- ... diagonal matrix containing all negative eigenvalues of matrix \mathbf{J} , zeros respectively
 μ_ϵ ... coefficient of entropy correction of Roe's scheme
 τ_f^i ... bed shear stress
 ϕ ... coefficient of the linear reconstruction used for control of the numerical viscosity
 Φ_i ... numerical approximation of inviscid flux in x-axis direction at the i -th edge
 φ_i^j ... the basis function of the linear approximation
 Ψ ... coefficient of the minmod limiter used for control of the numerical viscosity
 Ω ... computational domain
 Ω_i ... i -th finite volume
 $\partial\Omega$... boundary of the computational domain
 $\partial\Omega_i$... boundary of the finite volume Ω_i
 $|\Omega_i|$... measure of i -th finite volume
 ω_i^{n+1} ... integral average of \mathbf{W} at $n+1$ -th time level over special sub-domains $\iota_{1/2}$

Introduction

This work summarises recent findings in the numerical solutions of Shallow Water Equations. This mathematical model is also called Saint-Venant's Equations. The work contains the description of the mathematical model, numerical methods solving this model and novel findings in numerical schemes.

Motivation

The usage of the Shallow Water Equations has many practical applications. This mathematical model can be used for simulating floods caused by heavy torrential rains, transport of sediments or pollutant transport. The mathematical model is suitable to solve the river currents and can be used for design of the anti-flood barriers or river catchment restoration. Shallow Water Equations are also suitable for solving ocean currents or for design of the seaport anti-wave systems. Another application of Saint-Venant's Equation is connected to big shock waves like tsunami waves (e.g. Indonesian tsunami in 2004) or dam breaks caused by military actions, for example, the destruction of the Dnieprostroy Dam (140 feet high) near Kichkas on the Dnieper River in 1941 by Russian engineers, or the occasional mechanical failures, for example, the St. Francis Dam (205 feet high) in California, which collapsed in 1928 because of the defective geologic foundations (partial damming of the canyon was created by collapsed soil and not by rock). Another dam disaster was caused by hurricane Katrina, which damaged protection levees in New Orleans in 2005.

State of the Art

The first attempts of solving Shallow Water Equations were made in nineteenth century by B. Saint-Venant (1872) [1] who described the mathematical model and A. Ritter in 1892 who propounded some analytical solutions, for example the front wave velocity propagation of the dam break. This solution describes the propagation of the front wave with neglected resistances of the flume (friction, turbulence, etc.). The details of the analytical solution for rectangular dam break were summarized by Dressler in [2]. The same author has made precise experiments with different bed friction and outlined analytical solution in [3]. Another analytical solutions can be found in [4]. The most common analytical solutions of Shallow Water Equations are summarised in [5]. This article is also supplemented by free solver generating analytic and semi-analytic solutions of described solutions.

Opening chapter of present work describes the mathematical model of the Shallow Water Equations, some of its mathematical features and boundary conditions. Used mathematical model includes the bed slope source term which represents the irregularities of the bed. The description of the bed slope source can be found in [6]. Another source term considered in this work is the bed friction source term. This source term is well described in [7].

Within this work, the mathematical model is solved by the finite volume and discontinuous Galerkin method. Both methods are based on mesh which will be considered unvarying in this work. The numerical solutions are based on computation of the numerical flux between the mesh cells. The exact solution of the flux gives Riemann solver. Details of Riemann solver can be found in [8, 9]. The disadvantages of the exact Riemann solver are high computational demands and the fact that the usage of the Riemann solver is clear only if no source term is included. Another way of solving the numerical flux is usage of some approximate numerical scheme. In Chapter 2, dealing with finite volume method, two approximating schemes of the numerical flux are described. The first scheme is Roe scheme proposed by P. L. Roe in [10] and the second scheme is HLL scheme proposed in [11]. Both are based on up-wind principle. Description of another solver can be found for example in [6, 12] or [113].

Roe's approximating solver for 2D computations on orthogonal tetragonal meshes is described in [13] and the numerical scheme for unstructured triangular meshes can be found in [14–17]. HLL approximating solver for conservation laws on orthogonal tetragonal meshes can be found in [18], extension of this work for the unstructured meshes is described in [19]. In recent years there were also published articles dealing with bed slope source term for non-orthogonal meshes [14, 15, 20]. Audusse and al. described distribution of the bed source term to the cell interfaces [20]. After appropriate rotation this distribution can be used for 2D simulation on non-orthogonal meshes. Description of a suitable rotation is described for example in [21]. The approximation of the bed slope source term within finite volume method is described in Section 2.3. Section 3.2.3 describes the approximation of the bed source term within discontinuous Galerkin method. The discontinuous Galerkin method allows the polynomial description of the bed function but this function must be continuous at the edges of the mesh cells and smooth within the finite element. The finite volume method allows discontinuity at the edge of the finite volumes but the bed function can be approximated only by the linear functions. The nodes of this piecewise linear function can be evaluated either at the edge of the finite volume (for example [18, 22, 23]) or in the middle of the finite volume (for example [13, 24]). Combination of both methods is described in [21, 25]. Kesserwani [26] published investigative study for five different standard topography discretization methods that consist of the upwind scheme, the surface gradient method, the mathematically balanced set of the Shallow Water Equations, the hydrostatic reconstruction technique and the discontinuous Galerkin discretization. Another included source term is the bed friction source term representing river bed roughness. Description of this source term can be found in [16, 27–30]. Approximation of the bed friction needs special treatment when the water depth is approaching small values. This cases emerges for example during the wetting and drying processes in the computational domain and friction term can cause instability of the numerical scheme. Special limiting method for the finite volume method which handles these situations is approach described by Q. Liang and F. Marche in [31]. Another source term can be caused by the porosity of the computational are. Porosity flow is not discussed in this work, but we refer the reader to [32] for HLL solver details and to [17] to see the details on Roe scheme.

Wetting and drying processes of the cells occur in majority of the practical applications, e.g. part of the computational domain may fall dry or the domain may start initially dry and get flooded during the simulation. The interface between a wet and a dry computational area is usually referred to as the wet/dry front or wet/dry interface [33, 34]. Especially wet/dry interfaces over complex topography are known to cause numerical errors such as spurious oscillations in the flow velocity and negative water depths [20, 26, 34–40]. A positivity preserving correction of the water depth has been proposed by

Kurganov and Petrova [41]. This correction is positivity preserving, mass conservative and modifies the reconstructed velocities to avoid non-physically large velocities caused by the numerical inaccuracy of the computer. However, the C-property at the wet/dry interface, i.e. quiescent steady state, is not satisfied [33, 34]. Several other approaches for the wet/dry interface can be found in literature, such as the primitive variable reconstruction [42], which is not suitable for calculating a discontinuous solution [38] or the non-negative hydrostatic reconstruction method [20], which may fail at certain combinations of slope, water depth and mesh cell size [43]. Another approach is to limit the time step such that the cell does not 'dry out to the negative values' [35], however this approach degenerates the time step and increases computational effort. Bollermann et al. [33] modified the correction of Kurganov and Petrova [41] to obtain a C-property satisfying scheme. The scheme restricts the time step as in [35] to preserve positivity of water depth and adds water into the cell in special cases to satisfy the C-property. But, in author's opinion, this might result in loss of the mass conservation property of this scheme.

In the history many higher order schemes were developed. One of the method is linear reconstruction of the conservative variables which is second-order accurate. This method is described in Section 2.5. One of the first work with approximation higher than linear is parabolic approximation proposed in [44]. Higher order of approximation can be achieved by essentially non-oscillatory (ENO) schemes introduced in [45]. Better results can give the weighted essentially non-oscillatory schemes (WENO) [46–48] In Chapter 3 the discontinuous Galerkin method is described. This method has experienced a vigorous development since it's introduction in 1973 [49]. Although, the computational demands are higher than the computational demands of finite volume method, discontinuous Galerkin method provides formally arbitrarily high order of accuracy. Unfortunately this order of accuracy decreases around the shocks and discontinuities. In [50] there is suggested to use the eigenvalues correction and first-order scheme for time integration around the shocks. Even if the higher order schemes need higher order time integration, the rationale for a first-order time-stepping method is smaller spatial accuracy mentioned before. Shocks and discontinuities needs special treatment and limiting in general. Meanwhile in frame of the finite volume method each finite volume cell is limited (this process is referred as a global limiting process), in frame of discontinuous Galerkin method first the 'troubled' cells are located and these are limited (this process is referred as a local limiting process). There are various methods of defining 'troubled' cells like [51] where the criterion of the 'troubled' cell is connected by the magnitude of the solution discontinuity at the edges of the finite elements. In addition to discontinuity, Kesserwani added the criterion of solution monotonicity [52]. Limiting processes for 2D orthogonal basis functions are described in [53]. The limiting process for non-orthogonal Taylor basis can be found in [54].

The wetting and drying processes for the second order discontinuous Galerkin method are described in [55]. Another work dealing with second-order of accuracy and wet/dry interface is [36]. In [56] there are the wetting and drying processes described and the 'C-property' condition is ensured by least square projection of the bed function in to the polynomial space of basis functions. 'C-property' condition is also discussed in [57]. The numerical scheme satisfying the 'C-property' condition for the discontinuous bed function is described by Xing and Shu [58]. The same authors describe positivity-preserving limiter and the wetting and drying schemes in [59].

The time integration methods are described in Chapter 4. The time integration was done by Euler method and second and fourth order Runge-Kutta method. The discontinuous Galerkin method for the time integration wasn't chosen because of it's computational demands and no significant accuracy improvement [60].

The implemented solver was based on unstructured meshes. Even if unstructured meshes are clumsy to use, the advantages in space discretization makes unstructured meshes very useful. There are several ways of unstructured meshes implementation, one of them is described in Chapter 5. The computational performance of the numerical solver can be enhanced by the parallelisation of the code. The basics of code parallelisation are described in Chapter 6. The OpenMP library was chosen to perform the parallel computations.

At the end of this work the numerical results and validation against experiments and analytic solutions are shown. Chapter 7 is divided into two parts. In the first part, the numerical solver is validated against the analytical solutions and compared with the numerical schemes published in the literature. In the second part, the numerical results are compared with the experimental data. A lot of the experiments have been published in the literature since 1950's. One of the first published experiments was experiment conducted by R.F. Dressler [3] to validate his theoretical solution of the dam break on horizontal plane. Another dam break with flat bottom is experiment conducted by Martin and Moyce [61] where the collapse of liquid column was studied. The initial behaviour of the flow near the dam break site was investigated in [62]. Here the flow, in the flume with dimensions $15.24 \times 0.4 \times 0.4$ m, was recorded on a CCD video camera giving a field of view of about 0.83 m height and 1.0 m length. The flow was captured at different positions with the frame rate 25 FPS. In the experiment, there can be seen the 'mushroom like' flow patterns which occurs near the dam site at the beginning of the experiment and which cannot be captured by the Shallow Water Equations model. L.S. Nanía studied the flow division in steep street crossings [63]. In this work there was a set-up of the 'crossroad' of four flumes with angle of 90° between two adjacent canals. Steady inflows were set in two flumes and another two flumes were used as the outlets. Under the steady conditions the hydraulic jumps, discharges and the water depths in the flumes were captured. One of the experiments with the irregular bed was the experiment conducted by S. Soares-Frazão [64]. Here, the dynamics of the dam break wave over a triangular bottom sill was studied. The same author conducted an experiment of the dam break flow against an isolated obstacle [65]. Here the effects of a single building on a dam-break flow are presented. The building wasn't set neither in the centre of the channel, nor aligned with the flow direction. Measurements were done at five different locations. The water level evolution was measured by means of water level gauges and the velocity was measured by means of an acoustic Doppler velocimeter. The experiment of the dam break against an isolated obstacle was also measured by Maritime Research Institute Netherlands [66]. Except for the water depths, the impact pressure was measured. Another experiment, where the impact pressure was measured, is for example experiment of V. Bukreev and V. Zykov [67], here the force of the bore impacting a narrow vertical plate was investigated. In [68] authors conducted experimental measurements of a dam break wave impacting the downstream vertical wall. During the experiment, the pressure loads on the wall were measured. For the experiment, a prismatic tank with inner dimensions of $1610 \times 600 \times 150$ mm was used. Soares-Frazão and Zech [69] made experiment of the dam break in the 36 m long and 3.6 m wide flume. The initial water depth in the dam, which is 6.75 m long, is 0.4 m and the initial water depth in the rest of the flume is 0.011 m. In the front of the water gate, there were fixed square tubes simulating porous area. Except for the water depth, the velocity of the flow was measured. This velocity was measured by the particles in the water which were recorded by the camera with the recording rate 40 FPS. But as the authors say, the measured velocity does not give a good agreement with the numerical prediction where the velocity is depth averaged. In study of Alcrudo et al. [70], the flood in Toce river was simulated. The model was made in the scale 1:100 and in the flood area, the isolated buildings

were set. The length of the model is 50 m. The set-up of the buildings, simulated by the concrete cuboids, was aligned and staggered. In aligned set-up the buildings were placed in rows approximately parallel to the main axis of the valley. In staggered case, the buildings are placed in a checker board configuration.

Main Objectives of the Work

The aim of this work is description of the mathematical model and creation of the software which can be used for the simulation of floods and river flows. The main attention was given to the interface between flooded and dry area (wet/dry interface). Special bench mark for the numerical simulation of the Shallow Water Equations is 'C-property' condition. This condition was tested in fully flooded domain, but there is a lack of research made at this interface. There is also lack of the numerical schemes for discontinuous Galerkin method which can handle the higher orders of accuracy and wet/dry interface in the computational domain. So description of the discontinuous Galerkin method will also contain this topic. In the limiting processes, described by other authors, various constants are used to find the 'troubled cells'. But there is big uncertainty in magnitude of these constants (compare [51] and [52]). Thus another aim of this work is invention of the 'constantless' limiting process.

Chapter 1

Mathematical Model

Mathematical model of the shallow water was described by Saint-Venant [1]. This model is known as Saint-Venant's Equations or **Shallow Water Equations (SWE)**. The model is based on basic physical laws, namely the law of mass conservation and the law of momentum conservation, for details see [9] or [71]. The mathematical model is suitable for the free surface flows if the horizontal scales of water mass are much bigger than the vertical scale and the flow in vertical direction is negligible. The water density in our model is considered to be constant and Coriolis force is neglected. As the source terms the bed slope term and bed friction term are considered. Written as a compact vector equation, The differential conservation law form of two-dimensional SWE can be expressed as

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{S}_B + \mathbf{S}_f, \quad (1.1)$$

where \mathbf{W} is the vector of conservative variables

$$\mathbf{W} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad (1.2)$$

\mathbf{F} is vector function of inviscid flux in x-direction

$$\mathbf{F} = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}, \quad (1.3)$$

\mathbf{G} is vector function of inviscid flux in y-direction

$$\mathbf{G} = \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}, \quad (1.4)$$

\mathbf{S}_B is the source term caused by bottom irregularities

$$\mathbf{S}_B = \begin{bmatrix} 0 \\ -gh \frac{\partial B(x,y)}{\partial x} \\ -gh \frac{\partial B(x,y)}{\partial y} \end{bmatrix} \quad (1.5)$$

and \mathbf{S}_f is term caused by bottom-friction during the flow

$$\mathbf{S}_f = \begin{bmatrix} 0 \\ -\tau_f^x \\ -\tau_f^y \end{bmatrix}. \quad (1.6)$$

In this equations, h is the water depth, g is gravity, u, v respectively, is the velocity of the flow in x -direction, y -direction respectively. τ_f^x and τ_f^y are the bed shear stress and function $B(x,y)$ is the function describing the bottom topography which is considered to be constant in time within this work.

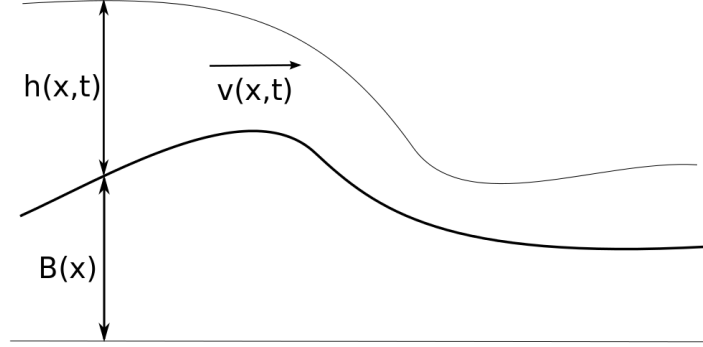


Figure 1.1: 1D shallow water flow over the bed function.

In 1D case the system (1.1) is simplified to

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \mathbf{S}_B + \mathbf{S}_f \quad (1.7)$$

where

$$\mathbf{W} = \begin{bmatrix} h \\ hu \end{bmatrix}, \mathbf{F} = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix}, \mathbf{S}_B = \begin{bmatrix} 0 \\ -gh \frac{\partial B(x)}{\partial x} \end{bmatrix}, \mathbf{S}_f = \begin{bmatrix} 0 \\ -\tau_f \end{bmatrix}. \quad (1.8)$$

1.1 Mathematical Features

The eigenvalues of the system of SWE play a fundamental role in understanding the mathematical character of this system and free-surface waves spreading. Equation (1.7) without the source terms can be written in quasi-linear form [6]

$$\frac{\partial}{\partial t} \mathbf{W} + \mathbf{J} \frac{\partial}{\partial x} \mathbf{W} = \mathbf{0}, \quad (1.9)$$

where the Jacobi matrix \mathbf{J} of inviscid flux $\mathbf{F} = [F_1, F_2]^T$ is defined as

$$\mathbf{J} = \frac{\partial \mathbf{F}}{\partial \mathbf{W}} = \begin{bmatrix} \frac{\partial F_1}{\partial W_1} & \frac{\partial F_1}{\partial W_2} \\ \frac{\partial F_2}{\partial W_1} & \frac{\partial F_2}{\partial W_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ c^2 - u^2 & 2u \end{bmatrix}, \quad (1.10)$$

where $[W_1, W_2]^T = \mathbf{W}$ and c is wave velocity equal to

$$c = \sqrt{gh}. \quad (1.11)$$

The eigenvalues of the Jacobi matrix (1.10) can be computed as

$$\mathbf{J} - \lambda \mathbf{I} = \mathbf{0}, \quad (1.12)$$

where \mathbf{I} is identity matrix. Equation (1.12) has two non-singular solutions, thus Jacobi matrix (1.10) has two eigenvalues

$$\lambda_1 = u - c, \quad \lambda_2 = u + c. \quad (1.13)$$

The equations

$$\mathbf{J}\mathbf{x}_1 = \lambda_1\mathbf{x}_1, \quad \mathbf{J}\mathbf{x}_2 = \lambda_2\mathbf{x}_2 \quad (1.14)$$

yields corresponding right eigenvectors

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ \lambda_1 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 1 \\ \lambda_2 \end{bmatrix}. \quad (1.15)$$

Definition 1. (*Hyperbolic system*) A system $\frac{\partial}{\partial t}\mathbf{W} + \mathbf{J}\frac{\partial}{\partial x}\mathbf{W} + \mathbf{B} = \mathbf{0}$ is said to be hyperbolic at a point (x, t) if \mathbf{J} has m real eigenvalues $\lambda_1, \dots, \lambda_m$ and a corresponding set of m linearly independent right eigenvectors $\mathbf{x}_1, \dots, \mathbf{x}_m$. The system is said to be strictly hyperbolic if the eigenvalues λ_i are all distinct [12].

When this system is strictly hyperbolic, i.e. when $h > 0$, the matrix \mathbf{J} is diagonalizable in the following way

$$\mathbf{X}^{-1}\mathbf{J}\mathbf{X} = \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \quad (1.16)$$

where $\mathbf{\Lambda}$ is spectral matrix and \mathbf{X} is modal matrix. The columns of the modal matrix are created by the eigenvectors i.e.

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2] = \begin{bmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{bmatrix}. \quad (1.17)$$

In 2D case the system (1.1) without the bed slope source term and the friction term can be also rewritten into the quasi-linear form:

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{J}_F \frac{\partial \mathbf{W}}{\partial x} + \mathbf{J}_G \frac{\partial \mathbf{W}}{\partial y} = \mathbf{0}. \quad (1.18)$$

There are two Jacobi matrices, connected to the system of Saint-Venant equations, in 2D space:

$$\mathbf{J}_F = \frac{\partial \mathbf{F}}{\partial \mathbf{W}} = \begin{bmatrix} 0 & 1 & 0 \\ c^2 - u^2 & 2u & 0 \\ -uv & v & u \end{bmatrix}, \quad \mathbf{J}_G = \frac{\partial \mathbf{G}}{\partial \mathbf{W}} = \begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ c^2 - v^2 & 0 & 2v \end{bmatrix}. \quad (1.19)$$

Their eigenvalues are

$$\lambda_1^F = u - c, \quad \lambda_2^F = u, \quad \lambda_3^F = u + c, \quad (1.20)$$

and

$$\lambda_1^G = v - c, \quad \lambda_2^G = v, \quad \lambda_3^G = v + c. \quad (1.21)$$

Corresponding eigenvectors are

$$\mathbf{x}_1^F = \begin{bmatrix} 1 \\ u - c \\ v \end{bmatrix}, \quad \mathbf{x}_2^F = \begin{bmatrix} 0 \\ 0 \\ c \end{bmatrix}, \quad \mathbf{x}_3^F = \begin{bmatrix} 1 \\ u + c \\ v \end{bmatrix}, \quad (1.22)$$

$$\mathbf{x}_1^G = \begin{bmatrix} 1 \\ u \\ v - c \end{bmatrix}, \quad \mathbf{x}_2^G = \begin{bmatrix} 0 \\ -c \\ 0 \end{bmatrix}, \quad \mathbf{x}_3^G = \begin{bmatrix} 1 \\ u \\ v + c \end{bmatrix}. \quad (1.23)$$

And modal matrices corresponding to the Jacobi matrices are:

$$\mathbf{X}^F = \begin{bmatrix} 1 & 0 & 1 \\ u - c & 0 & u + c \\ v & c & v \end{bmatrix} \quad \text{and} \quad \mathbf{X}^G = \begin{bmatrix} 1 & 0 & 1 \\ u & -c & u \\ v - c & 0 & v + c \end{bmatrix}. \quad (1.24)$$

1.2 Boundary Conditions

Boundary value problem is set of partial differential equations together with boundary conditions. The solution must satisfy the system of differential equations but also meet the boundary conditions at the boundary of the computational domain Ω . This boundary will be referred to as $\partial\Omega$.

There are three classical boundary conditions in theory of mathematics. The condition setting the normal derivative of a variable is called **Neumann boundary condition**. The condition setting a value to a variable is called **Dirichlet boundary condition**. **Cauchy boundary condition** is a set of conditions which determine the value and derivative of a variable simultaneously.

The number of variables, which have to be extrapolated to the boundary from the computational domain is equal to the number of characteristic curves which intersect the boundary $\partial\Omega$. The rest of values must be set as a boundary condition. Let x_0 be a chosen point in computational domain sufficiently close to the boundary. The characteristic curves of system (1.7) are

$$x = x_0 + \lambda_1 t \quad \text{and} \quad x = x_0 + \lambda_2 t, \quad (1.25)$$

where λ_1, λ_2 are eigenvalues (1.13). Both characteristic curves intersecting $\partial\Omega$ are depicted in Figure

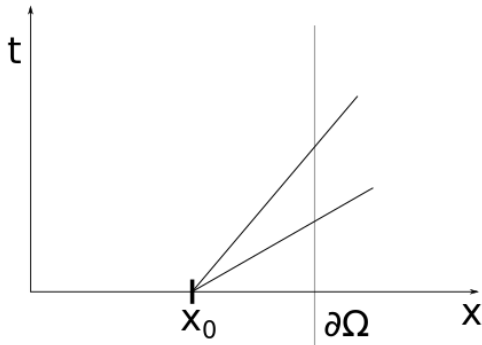


Figure 1.2: Two characteristics curves intersecting the boundary of the computational domain.

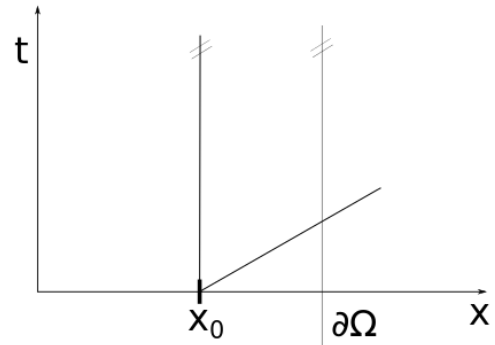


Figure 1.3: One characteristics curve intersecting the boundary of the computational domain.

1.2. This means that no boundary condition is set and both variables (water depth h and velocity u) are extrapolated from the computational domain Ω . In Figures 1.3 and 1.4 only one characteristic curve is intersecting $\partial\Omega$ so one arbitrary variable will be extrapolated and second will be set by boundary condition. And situation in Figure 1.5 needs both variables to be set by boundary condition.

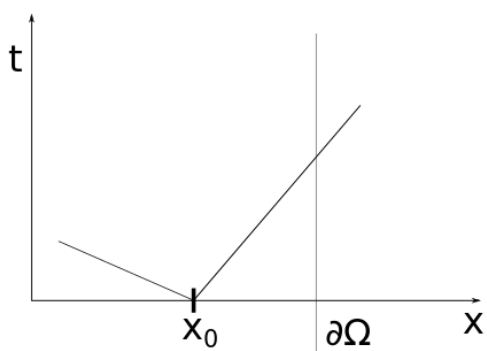


Figure 1.4: One characteristics curve intersecting the boundary of the computational domain.

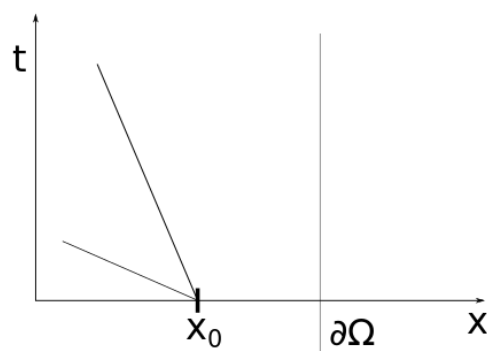


Figure 1.5: No characteristics curve intersecting the boundary of the computational domain.

Chapter 2

Finite Volume Method

In this chapter, the spatial discreteness will be implemented by the **Finite Volume Method** (FVM). The finite volume method is suitable for solving non-linear hyperbolic partial differential equations and can be implemented on both structured and unstructured meshes. These meshes can be created by arbitrary polygons. Another advantage of FVM is the capability to deal with discontinuous solutions. The most common polygons for meshing are triangles and quadrilaterals.

At the beginning the computational domain Ω is divided into discrete mesh of finite volumes $\Omega_i, i = 1, 2, \dots, n_t$. An example of a mesh can be seen in Figure 2.1.

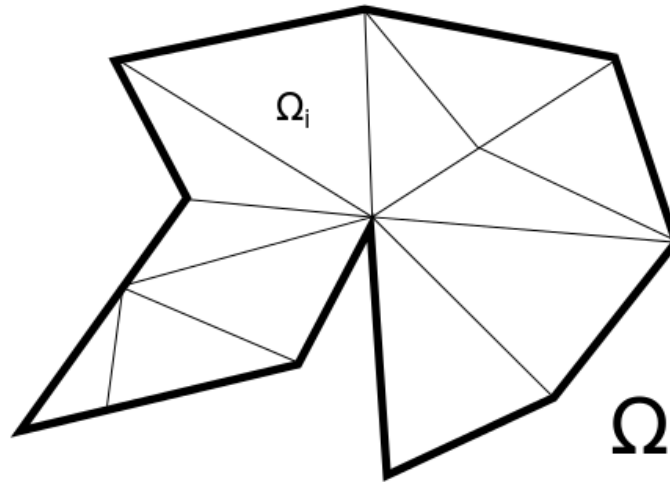


Figure 2.1: Computational domain Ω divided into the finite volumes Ω_i .

2.1 Derivation of the FVM

Derivation of the FVM can be described for the system of equations (1.1), i.e.

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{S}_b + \mathbf{S}_f,$$

as follows. The system of equations (1.1) are integrated over the finite volume Ω_i

$$\iint_{\Omega_i} \frac{\partial \mathbf{W}}{\partial t} dx dy + \iint_{\Omega_i} \left[\frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} \right] dx dy = \iint_{\Omega_i} [\mathbf{S}_b + \mathbf{S}_f] dx dy. \quad (2.1)$$

With the assumption of constant boundary of the finite volume Ω_i , the order of integration and derivation in the integral $\iint_{\Omega_i} \frac{\partial \mathbf{W}}{\partial t} dx dy$ can be changed. The integral is approximated by the integral average \mathbf{W}_i

$$\mathbf{W}_i(t) = \frac{\iint_{\Omega_i} \mathbf{W}(t) dx dy}{|\Omega_i|} \Rightarrow \iint_{\Omega_i} \mathbf{W}(t) dx dy = \mathbf{W}_i(t) |\Omega_i|, \quad (2.2)$$

where $|\Omega_i|$ is the measure of the finite volume. Similar treatment can be done with the right hand side of the equation (2.1)

$$\mathbf{S}_{b/f_i}(t) = \frac{\iint_{\Omega_i} \mathbf{S}_{b/f}(t) dx dy}{|\Omega_i|} \Rightarrow \iint_{\Omega_i} \mathbf{S}_{b/f}(t) dx dy = \mathbf{S}_{b/f_i}(t) |\Omega_i|. \quad (2.3)$$

In the next approach the Green's theorem is used.

Definition 2. *Green's theorem* is the relationship between a line integral around a simple closed curve and a double integral over the plane region bounded by the curve, which may be stated as [72]

$$\iint_{\Omega_i} \left(\frac{\partial f(x, y)}{\partial x} + \frac{\partial g(x, y)}{\partial y} \right) dx dy = \oint_{\partial \Omega_i} f(x, y) dx + g(x, y) dy. \quad (2.4)$$

Second integral in (2.1) can be modified using (2.4)

$$\iint_{\Omega_i} \left[\frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} \right] dx dy = \oint_{\partial \Omega_i} (\mathbf{F} n_x + \mathbf{G} n_y) dl, \quad (2.5)$$

where n_x, n_y are the components of the outer normal vector of the finite volume boundary $\partial \Omega_i$. Substitution of (2.2), (2.3) and (2.5) into (2.1) yields

$$\frac{\partial}{\partial t} \mathbf{W}_i |\Omega_i| + \oint_{\partial \Omega_i} (\mathbf{F} n_x + \mathbf{G} n_y) dl = \mathbf{S}_{b_i} |\Omega_i| + \mathbf{S}_{f_i} |\Omega_i|. \quad (2.6)$$

With the assumption of the triangular shape of the finite volumes, the components of the inviscid fluxes \mathbf{F} and \mathbf{G} can be replaced by summation of the numerical fluxes through all edges

$$\oint_{\partial \Omega_i} (\mathbf{F} n_x + \mathbf{G} n_y) dl = \sum_{k=1}^3 (\Phi_k n_{x_k} + \Gamma_k n_{y_k}) |l_k|, \quad (2.7)$$

where Φ_k (respectively Γ_k) is the value of the numerical flux at the edge k of the finite volume Ω_i in the x (respectively y) direction and $|l_k|$ is length of the corresponding edge of the Ω_i . Approximation of the numerical fluxes Φ_k/Γ_k and source terms $\mathbf{S}_{b_i}/\mathbf{S}_{f_i}$ is described in the following chapters.

The substitution into (2.6) yields the semi-discrete scheme of the finite volume method

$$\frac{d}{dt} \mathbf{W}_i = - \frac{1}{|\Omega_i|} \sum_{k=1}^3 (\Phi_k n_{x_k} + \Gamma_k n_{y_k}) |l_k| + \mathbf{S}_{b_i} + \mathbf{S}_{f_i}. \quad (2.8)$$

In 1D case, scheme (2.8) is simplified to

$$\frac{d}{dt} \mathbf{W}_i = -\frac{1}{|\Omega_i|} \sum_{k=1}^2 (\Phi_k^{n_{xk}}) |l_k| + \mathbf{S}_{bi} + \mathbf{S}_{fi}. \quad (2.9)$$

Both schemes (2.8) and (2.9) can be written in compact form

$$\frac{d}{dt} \mathbf{W}_i = \mathbf{RHS}_i + \mathbf{S}_{fi}. \quad (2.10)$$

The time integration of

$$\frac{d}{dt} \mathbf{W}_i = \mathbf{RHS}_i \quad (2.11)$$

is described in Section 4. The time integration of the bed friction source term \mathbf{S}_{fi} needs a special treatment which is described in Section 2.4.

2.2 Numerical Approximation of Inviscid Flux

This chapter is focused on numerical approximation of the inviscid flux \mathbf{F} from (1.7). Roe and HLL approximative schemes are described.

2.2.1 HLL Scheme

HLL scheme belongs to the group of Godunov-type schemes. The approach published in [18] is followed in this section.

Considering Saint-Venant equations in 1D case (1.7) without the source terms, i.e.

$$\frac{\partial \mathbf{W}(x, t)}{\partial t} + \frac{\partial \mathbf{F}(x, t)}{\partial x} = \mathbf{0}, \quad (2.12)$$

integrating this system over i -th control volume and a single time step yields the following identity

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{t^n}^{t^{n+1}} \left(\frac{\partial \mathbf{W}(x, t)}{\partial t} + \frac{\partial \mathbf{F}(x, t)}{\partial x} \right) dt dx = \mathbf{0}. \quad (2.13)$$

The integrals are separated

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{t^n}^{t^{n+1}} \frac{\partial \mathbf{W}(x, t)}{\partial t} dt dx = - \int_{t^n}^{t^{n+1}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial \mathbf{F}(x, t)}{\partial x} dx dt \quad (2.14)$$

and left-hand side is integrated over the time step and right-hand side is integrated over the finite volume

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (\mathbf{W}(x, t^{n+1}) - \mathbf{W}(x, t^n)) dx = - \int_{t^n}^{t^{n+1}} (\mathbf{F}(x_{i+\frac{1}{2}}, t) - \mathbf{F}(x_{i-\frac{1}{2}}, t)) dt. \quad (2.15)$$

Integral on the left-hand side can be also separated

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{W}(x, t^{n+1}) dx - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{W}(x, t^n) dx = - \int_{t^n}^{t^{n+1}} (\mathbf{F}(x_{i+\frac{1}{2}}, t) - \mathbf{F}(x_{i-\frac{1}{2}}, t)) dt. \quad (2.16)$$

Substitution of $\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{W}(x, t^{n+1}) dx$ for the integral average $\mathbf{W}_i(t^{n+1}) = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{W}(x, t^{n+1}) dx$, where $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$, and implementation of the nomenclature $\mathbf{W}_i(t^{n+1}) = \mathbf{W}_i^{n+1}$, respectively $\mathbf{W}(x, t^n) = \mathbf{W}^n(x)$, yields the solution in i^{th} finite volume in $(n+1)^{\text{th}}$ time level

$$\mathbf{W}_i^{n+1} = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{W}^n(x) dx - \frac{1}{\Delta x} \int_{t^n}^{t^{n+1}} (\mathbf{F}(x_{i+\frac{1}{2}}, t) - \mathbf{F}(x_{i-\frac{1}{2}}, t)) dt. \quad (2.17)$$

In general case vector \mathbf{W} in n^{th} time level is described by piecewise polynomial reconstruction

$$\mathbf{W}_i^n(x, t) = \mathbf{p}_i^n(x), \quad x_{i-\frac{1}{2}} < x < x_{i+\frac{1}{2}} \quad \forall i. \quad (2.18)$$

Vector \mathbf{p}^n is assembled by arbitrary polynomial functions. In general cases there are discontinuities at the points $x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}$, see Figure 2.2. Values of the vector of the conservative variables in n^{th} time level on the left and right side of the finite volume edge will be represented by vectors \mathbf{W}_i^W and \mathbf{W}_i^E . These vectors are defined as

$$\mathbf{W}_i^W = \mathbf{p}_i^n(x_{i-\frac{1}{2}}) \quad \text{and} \quad \mathbf{W}_i^E = \mathbf{p}_i^n(x_{i+\frac{1}{2}}). \quad (2.19)$$

Discontinuities at point $x_{i+\frac{1}{2}}$ propagate with right and left-sided local speeds which can be estimated by eigenvalues of the system (2.12)

$$\begin{aligned} a_{i+\frac{1}{2}}^- &= \min \{ \lambda_1(\mathbf{W}_{i+1}^W), \lambda_1(\mathbf{W}_i^E), 0 \}, \\ a_{i+\frac{1}{2}}^+ &= \max \{ \lambda_2(\mathbf{W}_{i+1}^W), \lambda_2(\mathbf{W}_i^E), 0 \}. \end{aligned} \quad (2.20)$$

In the following non-equal rectangular domains in time-space plane are considered:

$$\iota_1 = [x_{i-\frac{1}{2},r}^n, x_{i+\frac{1}{2},l}^n] \times [t^n, t^{n+1}] \quad \text{and} \quad \iota_2 = [x_{i+\frac{1}{2},l}^n, x_{i+\frac{1}{2},r}^n] \times [t^n, t^{n+1}], \quad (2.21)$$

where $x_{i+\frac{1}{2},l}^n = x_{i+\frac{1}{2}} + \Delta t a_{i+\frac{1}{2}}^-$ and $x_{i+\frac{1}{2},r}^n = x_{i+\frac{1}{2}} + \Delta t a_{i+\frac{1}{2}}^+$. The vector of conservative variables \mathbf{W} is continuous within the domain ι_1 . On the contrary this vector is discontinuous within domain ι_2 , see figure 2.2. Integral average of \mathbf{W} in time level $(n+1)$ over domain ι_1, ι_2 respectively, is referred to be $\omega_i^{n+1}, \omega_{i+\frac{1}{2}}^{n+1}$ respectively. Considering (2.17) this averages can be computed as

$$\omega_i^{n+1} = \frac{1}{x_{i+\frac{1}{2},l}^n - x_{i-\frac{1}{2},r}^n} \left[\int_{x_{i-\frac{1}{2},r}^n}^{x_{i+\frac{1}{2},l}^n} \mathbf{p}_i^n(x) dx - \int_{t^n}^{t^{n+1}} \left(\mathbf{F}(\mathbf{W}^n(x_{i+\frac{1}{2},l}^n, t)) - \mathbf{F}(\mathbf{W}^n(x_{i-\frac{1}{2},r}^n, t)) \right) dt \right], \quad (2.22)$$

and

$$\omega_{i+\frac{1}{2}}^{n+1} = \frac{1}{x_{i+\frac{1}{2},r}^n - x_{i+\frac{1}{2},l}^n} \left[\int_{x_{i+\frac{1}{2},l}^n}^{x_{i+\frac{1}{2}}} \mathbf{p}_i^n(x) dx + \int_{x_{i+\frac{1}{2}}}^{x_{i+\frac{1}{2},r}^n} \mathbf{p}_{i+1}^n(x) dx - \int_{t^n}^{t^{n+1}} \left(\mathbf{F}(\mathbf{W}^n(x_{i+\frac{1}{2},r}^n, t)) - \mathbf{F}(\mathbf{W}^n(x_{i+\frac{1}{2},l}^n, t)) \right) dt \right], \quad (2.23)$$

The integral average of the vector \mathbf{W}_i in time level $(n+1)$ over the finite volume Ω_i is obtained as

$$\mathbf{W}_i^{n+1} = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \omega^{n+1}(x) dx. \quad (2.24)$$

Aforesaid method describes the basic principle of the HLL scheme. The semi-discrete HLL scheme is derived in the following.

Time derivation of the vector \mathbf{W}_i along with (2.24) yields

$$\frac{d}{dt} \mathbf{W}_i(t) = \lim_{\Delta t \rightarrow 0} \frac{\mathbf{W}_i^{n+1} - \mathbf{W}_i^n}{\Delta t} = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \left[\frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \omega^{n+1}(x) dx - \mathbf{W}_i^n \right], \quad (2.25)$$

where Δt is the time step. Integral (2.24) can be replaced by following sums

$$\mathbf{W}_i^{n+1} = \frac{\Delta t a_{i-\frac{1}{2}}^+}{\Delta x} \omega_{i-\frac{1}{2}}^{n+1} + \frac{x_{i+\frac{1}{2},l} - x_{i-\frac{1}{2},r}}{\Delta x} \omega_i^{n+1} - \frac{\Delta t a_{i+\frac{1}{2}}^-}{\Delta x} \omega_{i+\frac{1}{2}}^{n+1}. \quad (2.26)$$

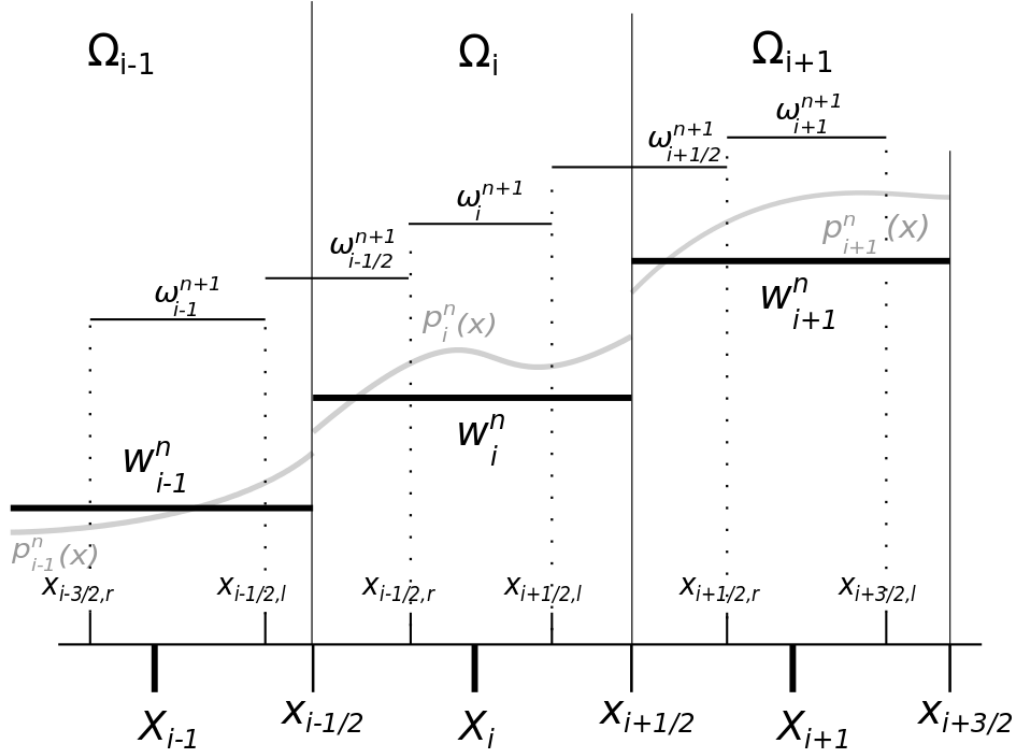


Figure 2.2: Principle of HLL scheme.

Substitution into (2.25) yields

$$\frac{d}{dt} \mathbf{W}_i(t) = \frac{a_{i-\frac{1}{2}}^+}{\Delta x} \lim_{\Delta t \rightarrow 0} \omega_{i-\frac{1}{2}}^{n+1} + \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \left(\frac{x_{i+\frac{1}{2},l} - x_{i-\frac{1}{2},r}}{\Delta x} \omega_i^{n+1} - \mathbf{W}_i^n \right) - \frac{a_{i+\frac{1}{2}}^-}{\Delta x} \lim_{\Delta t \rightarrow 0} \omega_{i+\frac{1}{2}}^{n+1}. \quad (2.27)$$

This three limits are computed separately. Second limit can be computed with aid of (2.22) as

$$\begin{aligned} & \lim_{\Delta t \rightarrow 0} \left(\frac{x_{i+\frac{1}{2},l} - x_{i-\frac{1}{2},r}}{\Delta x} \omega_i^{n+1} - \mathbf{W}_i^n \right) = \\ & \lim_{\Delta t \rightarrow 0} \left\{ \frac{1}{\Delta x} \frac{x_{i+\frac{1}{2},l} - x_{i-\frac{1}{2},r}}{x_{i+\frac{1}{2},l} - x_{i-\frac{1}{2},r}} \left[\int_{x_{i-\frac{1}{2},r}}^{x_{i+\frac{1}{2},l}} \mathbf{p}_i(x) dx - \int_{t^n}^{t^{n+1}} \left(\mathbf{F}(\mathbf{W}(x_{i+\frac{1}{2},l}, t)) - \mathbf{F}(\mathbf{W}(x_{i-\frac{1}{2},r}, t)) \right) dt \right] - \mathbf{W}_i^n \right\} = \\ & \lim_{\Delta t \rightarrow 0} \left[\frac{\int_{x_{i-\frac{1}{2},r}}^{x_{i+\frac{1}{2},l}} \mathbf{p}_i(x) dx}{\Delta x} - \mathbf{W}_i^n - \frac{\int_{t^n}^{t^{n+1}} \left(\mathbf{F}(\mathbf{W}(x_{i+\frac{1}{2},l}, t)) - \mathbf{F}(\mathbf{W}(x_{i-\frac{1}{2},r}, t)) \right) dt}{\Delta x} \right]. \end{aligned} \quad (2.28)$$

It should be noticed that $\lim_{\Delta t \rightarrow 0} x_{i+\frac{1}{2},l} = x_{i+\frac{1}{2}} - \delta$ and $\lim_{\Delta t \rightarrow 0} x_{i+\frac{1}{2},r} = x_{i+\frac{1}{2}} + \delta$ for $\delta \rightarrow 0^+$. Then

$$\lim_{\Delta t \rightarrow 0} \left(\frac{x_{i+\frac{1}{2},l} - x_{i-\frac{1}{2},r}}{\Delta x} \omega_i^{n+1} - \mathbf{W}_i^n \right) = \frac{a_{i+\frac{1}{2}}^- \mathbf{W}_i^E - a_{i+\frac{1}{2}}^+ \mathbf{W}_i^W}{\Delta x} - \frac{\mathbf{F}(\mathbf{W}_i^E) - \mathbf{F}(\mathbf{W}_i^W)}{\Delta x}. \quad (2.29)$$

Equation 2.23 results in

$$\lim_{\Delta t \rightarrow 0} \omega_{i+\frac{1}{2}}^{n+1} = \frac{a_{i+\frac{1}{2}}^+ \mathbf{W}_{i+1}^W - a_{i+\frac{1}{2}}^- \mathbf{W}_i^E}{a_{i+\frac{1}{2}}^+ - a_{i+\frac{1}{2}}^-} - \frac{\mathbf{F}(\mathbf{W}_{i+1}^W) - \mathbf{F}(\mathbf{W}_i^E)}{a_{i+\frac{1}{2}}^+ - a_{i+\frac{1}{2}}^-}. \quad (2.30)$$

Substituting these limits into (2.27) leads to the final semi-discrete HLL scheme

$$\frac{d}{dt} \mathbf{W}_i(t) = - \frac{\Phi_{i+\frac{1}{2}}(t) - \Phi_{i-\frac{1}{2}}(t)}{\Delta x}, \quad (2.31)$$

where the numerical flux across the edge of finite volume $\Phi_{i+\frac{1}{2}}$ is expressed as follows [18]

$$\Phi_{i+\frac{1}{2}} = \frac{a_{i+\frac{1}{2}}^+ \mathbf{F}(\mathbf{W}_i^E) - a_{i+\frac{1}{2}}^- \mathbf{F}(\mathbf{W}_{i+1}^W)}{a_{i+\frac{1}{2}}^+ - a_{i+\frac{1}{2}}^-} + \frac{a_{i+\frac{1}{2}}^+ a_{i+\frac{1}{2}}^-}{a_{i+\frac{1}{2}}^+ - a_{i+\frac{1}{2}}^-} [\mathbf{W}_{i+1}^W - \mathbf{W}_i^E]. \quad (2.32)$$

2.2.1.1 2D Extension

In the following section, HLL scheme is applied to the 2D problem of the Shallow Water Equations. Let us consider an edge with the normal vector $\vec{n} = [n_x, n_y]^T$ which connects the finite volumes Ω_L and Ω_R (see Figure 2.3) and $|\vec{n}| = 1$. In the first step the Riemann's problem is transformed from the

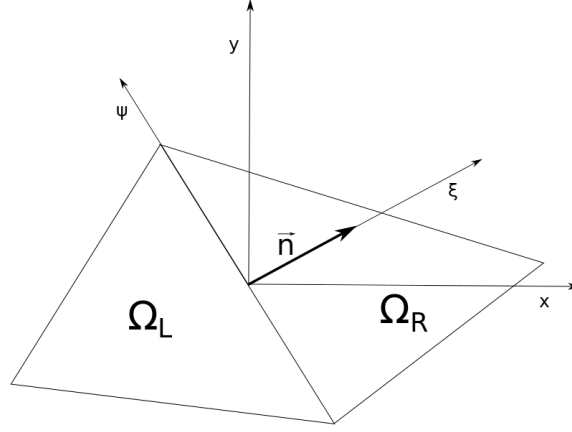


Figure 2.3: An edge connecting the finite volumes Ω_L and Ω_R with the normal vector \vec{n} .

global coordinate system (x, y) to the local coordinate system (ξ, ψ) where ξ is the direction of the normal vector and ψ is the transverse direction. The rotation from the global coordinate system to the local system can be expressed by matrix \mathbf{P} . The rotation affects the discharges but does not affect water depth, thus the rotational matrix has the form

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & n_x & -n_y \\ 0 & n_y & n_x \end{bmatrix}. \quad (2.33)$$

Numerical flux Φ is computed using the following procedure. Firstly, the Riemann problem in the global coordinate system (x, y) is transformed to the local coordinate system (ξ, ψ) via a left-hand multiplication by \mathbf{P}^{-1} . Secondly, the local Riemann problem is solved using HLL Riemann solver described previously, this yields the numerical flux Φ^* . Finally, the flux is transformed back to the (x, y) coordinate system by

$$\Phi = \mathbf{P}\Phi^*. \quad (2.34)$$

The system of equations (2.12) can be now solved in directional way and have the form

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}^*}{\partial \xi} = \mathbf{0} \quad (2.35)$$

with

$$\mathbf{W} = \begin{bmatrix} h \\ hu_n \\ hu_t \end{bmatrix}, \quad \mathbf{F}^* = \begin{bmatrix} hu_n \\ hu_n^2 + \frac{1}{2}gh^2 \\ hu_nu_t \end{bmatrix} \quad (2.36)$$

where u_n , u_t are velocities in the normal and tangent direction.

Mass discharge and normal momentum flux computations. The continuity equation, normal momentum flux respectively, is solved directly by the scheme (2.32). First, second respectively, component of the vector of the conservative variables and flux defined in (2.36) are used for the computations in (2.32). It can be noticed that

$$\begin{aligned} u_n &= u \cdot n_x + v \cdot n_y, \\ u_t &= -u \cdot n_y + v \cdot n_x. \end{aligned} \quad (2.37)$$

Tangent momentum flux computations. The tangent momentum flux is computed by the third components of (2.36). Within the finite volume, there is no solution discontinuity and no need to use the approximative Riemann solver. Thus only upwind scheme is used. If the velocity normal velocity is positive then for the computation $u_{t,L}$ is used, otherwise $u_{t,R}$ is used. Following formulae accounts for both situations [32]

$$hu_nu_t = \frac{h_L u_{n,L} + |h_L u_{n,L}|}{2} u_{t,L} + \frac{h_R u_{n,R} - |h_R u_{n,R}|}{2} u_{t,R}. \quad (2.38)$$

2.2.2 Roe's Scheme

The Roe solver of the Riemann's problem was published in [10] and [73]. Since then this solver has been improved and used in wide range of physical problems. It's fundamental idea is based on replacement of original non-linear system of differential equations by new linearised system of differential equations with constant coefficients. The solver for inviscid flux is well described for both structured [13] and unstructured meshes [14, 15]. The Roe scheme is of first-order accuracy only. The accuracy of the numerical method can be increased by the combination of Roe scheme with Lax-Wendroff scheme as described in [13]. Another way how to achieve higher accuracy is linear reconstruction of the vector of conservative variables described in Section 2.5. Influence of min-mod, superbee and beta limiter on the solution, when linear reconstruction is implemented, is described in [14].

2.2.2.1 Linearisation of Riemann's problem

Riemann's problem, is based on finding solution of the set of differential equations

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0, \quad (2.39)$$

with initial condition

$$\mathbf{W}(x, 0) = \begin{cases} \mathbf{W}_L, & \forall x < 0, \\ \mathbf{W}_R, & \forall x > 0. \end{cases} \quad (2.40)$$

Using the Jacobi matrix $\mathbf{J}(\mathbf{W}) = \frac{\partial \mathbf{F}}{\partial \mathbf{W}}$, see equation (1.10), this system can be rewritten into the form

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{F}}{\partial \mathbf{W}} \frac{\partial \mathbf{W}}{\partial x} = \frac{\partial \mathbf{W}}{\partial t} + \mathbf{J}(\mathbf{W}) \frac{\partial \mathbf{W}}{\partial x} = 0. \quad (2.41)$$

Roe, in his work, replaced non-linear matrix $\mathbf{J}(\mathbf{W})$ by the matrix \mathbf{A} with constant coefficients. This new matrix is a function of the vector of conservative variables on the left and right side of the discontinuity, $\mathbf{A} = \mathbf{A}(\mathbf{W}_L, \mathbf{W}_R)$. Thus it simplifies the original system (2.39) to system of differential equations with constant coefficients. Riemann's problem (2.39) is now replaced by approximate Riemann's problem for the system of partial differential equations

$$\begin{aligned} \frac{\partial \mathbf{W}}{\partial t} + \mathbf{A} \frac{\partial \mathbf{W}}{\partial x} &= 0, \text{ with initial condition} \\ \mathbf{W}(x, 0) &= \begin{cases} \mathbf{W}_L, & \forall x < 0, \\ \mathbf{W}_R, & \forall x > 0, \end{cases} \end{aligned} \quad (2.42)$$

which can be solved exactly.

The matrix \mathbf{A} given above has to satisfy special requirements [12]:

1. Eigenvalues of \mathbf{A} must be real with the set of linearly independent eigenvectors
2. Equality $\mathbf{J}(\mathbf{W}) = \mathbf{A}(\mathbf{W}_L, \mathbf{W}_R)$ for $\mathbf{W}_L = \mathbf{W}_R$
3. $\mathbf{F}(\mathbf{W}_L) - \mathbf{F}(\mathbf{W}_R) = \mathbf{A}(\mathbf{W}_L, \mathbf{W}_R) \times (\mathbf{W}_L - \mathbf{W}_R)$

It is obvious that the first requirement has to be kept to preserve the hyperbolic character of the original set of non-linear differential equations. Second requirement ensures consistency with conservation laws. And the last requirement ensures conservation and exact recognition of isolated discontinuities. It means that if the data W_L , W_R are connected by a single, isolated discontinuity, then the approximate Riemann solver solves (2.42) exactly [12].

First two requirements can be easily satisfied for example by taking $\mathbf{A} = \frac{1}{2}(\mathbf{J}(\mathbf{W}_L) + \mathbf{J}(\mathbf{W}_R))$ or $\mathbf{A} = \mathbf{J}(\frac{\mathbf{W}_L + \mathbf{W}_R}{2})$. But these matrices do not fulfil the third requirement, i.e. do not provide exact solution of (2.42).

Matrix \mathbf{A} can be found as follows. Considering the parameter $\Theta \in \langle 0, 1 \rangle$ along the curve connecting \mathbf{W}_L a \mathbf{W}_R so that

$$\mathbf{W}(\Theta) = \mathbf{W}_L + \Theta(\mathbf{W}_R - \mathbf{W}_L), \quad d\mathbf{W} = (\mathbf{W}_R - \mathbf{W}_L)d\Theta, \quad (2.43)$$

then

$$\mathbf{F}(\mathbf{W}_R) - \mathbf{F}(\mathbf{W}_L) = \int_0^1 \frac{d\mathbf{F}}{d\Theta} d\Theta = \int_0^1 \frac{d\mathbf{F}}{d\mathbf{W}} d\Theta \frac{d\mathbf{W}}{d\Theta} = \int_0^1 \mathbf{J}(\Theta) d\Theta \frac{d\mathbf{W}}{d\Theta} = \int_0^1 \mathbf{J}(\Theta) d\Theta (\mathbf{W}_R - \mathbf{W}_L). \quad (2.44)$$

This implies

$$\mathbf{A} = \int_0^1 \mathbf{J}(\Theta) d\Theta. \quad (2.45)$$

Unfortunately this integral cannot be evaluated directly. Therefore construction of matrix \mathbf{A} must be done in another way. In the following let Δ mean $\Delta(\cdot) = (\cdot)_R - (\cdot)_L$. For the system of Saint-Venant equations the parametric vector \mathbf{z} [10]

$$\mathbf{z} = \begin{bmatrix} \sqrt{h} \\ \sqrt{hu} \end{bmatrix} \quad (2.46)$$

can be considered and the third requirement, i.e. $\Delta \mathbf{F} = \mathbf{A} \Delta \mathbf{W}$, can be expressed as

$$\begin{aligned} \underbrace{\mathbf{C} \Delta \mathbf{z}}_{\Delta \mathbf{F}} &= \mathbf{A} \underbrace{\mathbf{B} \Delta \mathbf{z}}_{\Delta \mathbf{W}} \\ &\Downarrow \\ \mathbf{A} &= \mathbf{C} \mathbf{B}^{-1} \end{aligned} \quad (2.47)$$

The following identity will be used in process of finding matrices \mathbf{B} and \mathbf{C} [12]

$$\Delta(pq) = \bar{p}\Delta q + \bar{q}\Delta p, \quad (2.48)$$

where \bar{p} (respectively \bar{q}) denotes arithmetic mean of a quantity. Considering this identity and supposing non-negative h , $\Delta\mathbf{W}$ can be expressed as

$$\Delta\mathbf{W} = \begin{bmatrix} \Delta h \\ \Delta(hu) \end{bmatrix} = \begin{bmatrix} \Delta(\sqrt{h}\sqrt{h}) \\ \Delta(\sqrt{h} \cdot \sqrt{hu}) \end{bmatrix} = \begin{bmatrix} 2\Delta\sqrt{h} \cdot \sqrt{h} \\ \sqrt{h}\Delta(\sqrt{hu}) + \sqrt{hu}\Delta\sqrt{h} \end{bmatrix}, \quad (2.49)$$

and matrix \mathbf{B} for Shallow Water Equations can be derived

$$\underbrace{\begin{bmatrix} \Delta h \\ \Delta(hu) \end{bmatrix}}_{\Delta\mathbf{W}} = \underbrace{\begin{bmatrix} 2\sqrt{h} & 0 \\ \sqrt{hu} & \sqrt{h} \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} \Delta\sqrt{h} \\ \Delta(\sqrt{hu}) \end{bmatrix}}_{\Delta\mathbf{z}} \quad (2.50)$$

Matrix \mathbf{C} can be computed similarly

$$\underbrace{\begin{bmatrix} \Delta(hu) \\ \Delta(hu^2 + \frac{1}{2}gh^2) \end{bmatrix}}_{\Delta\mathbf{F}} = \underbrace{\begin{bmatrix} \sqrt{hu} & \sqrt{h} \\ 2g\sqrt{h} & \sqrt{h}^2 \end{bmatrix}}_{\mathbf{C}} \underbrace{\begin{bmatrix} \Delta\sqrt{h} \\ \Delta(\sqrt{hu}) \end{bmatrix}}_{\Delta\mathbf{z}}. \quad (2.51)$$

Then accordingly to (2.47)

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ g\frac{h_L+h_R}{2} - \left(\frac{u_L\sqrt{h_L}+u_R\sqrt{h_R}}{\sqrt{h_L}+\sqrt{h_R}}\right)^2 & 2\frac{u_L\sqrt{h_L}+u_R\sqrt{h_R}}{\sqrt{h_L}+\sqrt{h_R}} \end{bmatrix}. \quad (2.52)$$

By introducing Roe's averaged variables of flux celerity $u_{\frac{1}{2}}$ and celerity of information spreading $c_{\frac{1}{2}}$

$$u_{\frac{1}{2}} = \frac{u_L\sqrt{h_L}+u_R\sqrt{h_R}}{\sqrt{h_L}+\sqrt{h_R}}, \quad (2.53)$$

$$c_{\frac{1}{2}} = \sqrt{g\frac{h_L+h_R}{2}},$$

matrix \mathbf{A} can be written as

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ c_{\frac{1}{2}}^2 - u_{\frac{1}{2}}^2 & 2u_{\frac{1}{2}} \end{bmatrix}. \quad (2.54)$$

It's eigenvalues are

$$\lambda_1 = u_{\frac{1}{2}} - c_{\frac{1}{2}}, \quad \lambda_2 = u_{\frac{1}{2}} + c_{\frac{1}{2}} \quad (2.55)$$

and corresponding eigenvectors are

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ \lambda_1 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 1 \\ \lambda_2 \end{bmatrix}. \quad (2.56)$$

2.2.2.2 Numerical Approximation of the Flux

As described in Section 1.1, Jacobi matrix \mathbf{J} can be diagonalised and expressed as $\mathbf{J} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$, where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ is the modal matrix, it's columns contain corresponding eigenvectors $\mathbf{x}_1, \dots, \mathbf{x}_n$, and $\mathbf{\Lambda} = \text{diag}[\lambda_1, \lambda_2, \dots, \lambda_n]$ is the spectral matrix containing eigenvalues. New matrices \mathbf{J}^+ and \mathbf{J}^- are defined as

$$\mathbf{J}^+ = \mathbf{X}\mathbf{\Lambda}^+\mathbf{X}^{-1}, \quad \mathbf{J}^- = \mathbf{X}\mathbf{\Lambda}^-\mathbf{X}^{-1}. \quad (2.57)$$

In these relations

$$\begin{aligned}\mathbf{A}^+ &= \text{diag}(\lambda_1^+, \lambda_2^+, \dots, \lambda_n^+), \quad \text{where } \lambda_i^+ = \max(\lambda_i, 0), \\ \mathbf{A}^- &= \text{diag}(\lambda_1^-, \lambda_2^-, \dots, \lambda_n^-), \quad \text{where } \lambda_i^- = \min(\lambda_i, 0).\end{aligned}\quad (2.58)$$

The flux of the system of linear differential equations in the finite volume is expressed as $\mathbf{F} = \mathbf{A}\mathbf{U}$. Accordingly to the theory of up-wind schemes, applied to the systems of differential equations with constant coefficients, the numerical flux through the edge of the finite volume can be computed as [74]

$$\mathbf{F}_{1/2} = \mathbf{F}^+ + \mathbf{F}^-, \quad (2.59)$$

where \mathbf{F}^+ and \mathbf{F}^- can be expressed as

$$\mathbf{F}^+ = \mathbf{J}^+ \mathbf{W}_L, \quad \mathbf{F}^- = \mathbf{J}^- \mathbf{W}_R. \quad (2.60)$$

Instead of the original Jacobi matrix \mathbf{J} , matrix \mathbf{A} is used in approximate Roe solver of Riemann's problem (2.42). Thus the numerical flux $\Phi_{i-\frac{1}{2}}$ through edge of finite volume is computed as

$$\Phi_{i-\frac{1}{2}} = \Phi_{i-1}^+ + \Phi_i^- = \mathbf{A}^+ \mathbf{W}_{i-1} + \mathbf{A}^- \mathbf{W}_i. \quad (2.61)$$

Matrices \mathbf{A}^+ and \mathbf{A}^- were derived from \mathbf{A} in the same way as $\mathbf{J}^{+/-}$ in (2.57), i.e.

$$\mathbf{A}^+ = \mathbf{X}_a \mathbf{\Lambda}_a^+ \mathbf{X}_a^{-1}, \quad \mathbf{A}^- = \mathbf{X}_a \mathbf{\Lambda}_a^- \mathbf{X}_a^{-1}, \quad (2.62)$$

where index a means that the spectral and modal matrices belongs to the matrix \mathbf{A} . For the sake of simplicity, this index will be omitted in the following.

Hereafter $|\mathbf{A}|$ is defined as

$$|\mathbf{A}| = \mathbf{X} |\mathbf{\Lambda}| \mathbf{X}^{-1} \quad \text{where } |\mathbf{\Lambda}| = \text{diag}(|\lambda_1|, |\lambda_2|, \dots, |\lambda_n|). \quad (2.63)$$

It can be easily proofed that

$$\begin{aligned}\mathbf{A} &= \mathbf{A}^+ + \mathbf{A}^-, \\ |\mathbf{A}| &= \mathbf{A}^+ - \mathbf{A}^-, \\ \mathbf{A}^+ &= \frac{\mathbf{A} + |\mathbf{A}|}{2}, \quad \mathbf{A}^- = \frac{\mathbf{A} - |\mathbf{A}|}{2}.\end{aligned}\quad (2.64)$$

With this formulas the numerical flux (2.61) can be expressed as

$$\Phi_{i-\frac{1}{2}} = \frac{\mathbf{A} + |\mathbf{A}|}{2} \mathbf{W}_{i-1} + \frac{\mathbf{A} - |\mathbf{A}|}{2} \mathbf{W}_i = \frac{\mathbf{A} \mathbf{W}_{i-1} + \mathbf{A} \mathbf{W}_i}{2} - \frac{1}{2} |\mathbf{A}| (\mathbf{W}_i - \mathbf{W}_{i-1}) = \frac{\mathbf{F}_{i-1} + \mathbf{F}_i}{2} - \frac{1}{2} |\mathbf{A}| (\mathbf{W}_i - \mathbf{W}_{i-1}). \quad (2.65)$$

Taking in account the requirement of linear independence of eigenvectors, the difference $\mathbf{W}_i - \mathbf{W}_{i-1}$ can be computed as linear combination

$$\mathbf{W}_i - \mathbf{W}_{i-1} = \sum_{p=1}^2 \alpha_p \mathbf{x}_p. \quad (2.66)$$

Coefficients of this linear combination are [13]

$$(\alpha_{1,2})_{i-\frac{1}{2}} = \frac{(h_i u_i - h_{i-1} u_{i-1}) + (-u_{i-\frac{1}{2}} \mp c_{i-\frac{1}{2}})(h_i - h_{i-1})}{\mp 2c_{i-\frac{1}{2}}}. \quad (2.67)$$

The relation between matrix \mathbf{A} , it's eigenvalue λ_p and corresponding eigenvector \mathbf{x}_p is

$$\mathbf{A} \mathbf{x}_p = \lambda_p \mathbf{x}_p. \quad (2.68)$$

Similar relation can be also derived for the matrix $|\mathbf{A}|$:

$$\begin{aligned} |\mathbf{A}| &= \mathbf{X}|\mathbf{\Lambda}|\mathbf{X}^{-1} \\ &\Downarrow \\ |\mathbf{A}|\mathbf{X} &= \mathbf{X}|\mathbf{\Lambda}|. \end{aligned} \quad (2.69)$$

This relation yields matrices

$$\left[|\mathbf{A}| \mathbf{x}_1 \quad |\mathbf{A}| \mathbf{x}_2 \quad \cdots \quad |\mathbf{A}| \mathbf{x}_n \right] = \left[\mathbf{x}_1 |\lambda_1| \quad \mathbf{x}_2 |\lambda_2| \quad \cdots \quad \mathbf{x}_n |\lambda_n| \right] \quad (2.70)$$

By comparison of the columns we get

$$|\mathbf{A}|\mathbf{x}_p = |\lambda_p|\mathbf{x}_p. \quad (2.71)$$

And the final form of the numerical flux of the Roe's scheme is

$$\Phi_{i-\frac{1}{2}} = \frac{(\mathbf{F}_{i-1} + \mathbf{F}_i)}{2} - \frac{1}{2} \sum_{p=1}^2 [|\lambda_p| \alpha_p \mathbf{x}_p]_{i-\frac{1}{2}}. \quad (2.72)$$

2.2.2.3 2D Extension

In the previous section, the Roe's approximating solver of the Riemann's problem for 1D conservation laws was derived. In this section the Roe's scheme for 2D problem of the Shallow Water Equations is described. Let us consider an edge with the normal vector $\vec{n} = [n_x, n_y]^T$ which connects the finite volumes Ω_L and Ω_R (see Figure 2.3) and length of the normal vector $|\vec{n}| = 1$. The numerical flux is computed as [75]

$$\begin{aligned} \Phi_{L,R} &= \frac{1}{2} (\mathbf{f}(\mathbf{W}_L) + \mathbf{f}(\mathbf{W}_R)) - |\mathbf{A}| (\mathbf{W}_R - \mathbf{W}_L), \\ |\mathbf{A}| &= \mathbf{R}|\mathbf{\Lambda}|\mathbf{L} \end{aligned} \quad (2.73)$$

where

$$\begin{aligned} \mathbf{f}(\mathbf{W}) &= \mathbf{F}(\mathbf{W})n_x + \mathbf{G}(\mathbf{W})n_y = \begin{bmatrix} hv_n \\ huv_n + \frac{1}{2}gh^2n_x \\ hvv_n + \frac{1}{2}gh^2n_y \end{bmatrix}, \\ v_n &= un_x + vn_y \end{aligned} \quad (2.74)$$

and Roe matrix

$$\mathbf{A} = \frac{\partial(\mathbf{f} \cdot \vec{n})}{\partial \mathbf{W}} = \begin{bmatrix} 0 & n_x & n_y \\ (c^2 - u^2)n_x - uvn_y & 2un_x + vn_y & un_y \\ -uvn_x + (c^2 - v^2)n_y & vn_x & un_x + 2vn_y \end{bmatrix}. \quad (2.75)$$

The eigenvalues of (2.75) are

$$\lambda_1 = un_x + vn_y, \quad \lambda_2 = un_x + vn_y - c, \quad \lambda_3 = un_x + vn_y + c, \quad (2.76)$$

the right eigenvector matrix is given by

$$\mathbf{R} = \begin{bmatrix} 0 & 1 & 1 \\ n_y & u - cn_x & u + cn_x \\ -n_x & v - cn_y & v + cn_y \end{bmatrix} \quad (2.77)$$

and the left eigenvector matrix is given by

$$\mathbf{L} = \begin{bmatrix} -(un_y - vn_x) & n_y & -n_x \\ (un_x + vn_y)/(2c) + \frac{1}{2} & -n_x/(2c) & -n_y/(2c) \\ -(un_x + vn_y)/(2c) & n_x/(2c) & n_y/(2c) \end{bmatrix}. \quad (2.78)$$

The matrix $|\mathbf{A}|$ is computed as

$$|\mathbf{A}| = \begin{bmatrix} |\lambda_1| & 0 & 0 \\ 0 & |\lambda_2| & 0 \\ 0 & 0 & |\lambda_3| \end{bmatrix} \quad (2.79)$$

and for the computation the Roe's averaged variables are used

$$\begin{aligned} c &= \sqrt{g \frac{h_L + h_R}{2}}, \\ u &= \frac{\sqrt{h_L} u_L + \sqrt{h_R} u_R}{\sqrt{h_L} + \sqrt{h_R}}, \\ v &= \frac{\sqrt{h_L} v_L + \sqrt{h_R} v_R}{\sqrt{h_L} + \sqrt{h_R}}. \end{aligned} \quad (2.80)$$

2.2.3 Entropy Correction

Schemes based on up-wind principle suffer from lack of the numerical viscosity and give non-physical solution when Froude number takes on specific value. In the case of SWE, for flux-difference schemes such critical values are the sonic ones corresponding to $F_r = 1$, while for flux-splitting schemes the critical points correspond to $F_r = 1/\sqrt{2}$ [24]. For these critical values one of the eigenvalues of Jacobi matrix vanishes, thus producing a lack of the numerical viscosity. There are several techniques used to add artificial viscosity to the scheme in these cases [76]. One of the most used technique is the regularization of the eigenvalues proposed by Harten [11].

Definition 3. *Froude number* is a dimensionless number defined by William Froude and is expressed as

$$F_r = \frac{u}{\sqrt{gl}}, \quad (2.81)$$

where u is velocity of the fluid flow, g is gravity and l is characteristics length. The term \sqrt{gl} means wave velocity. For the equations of shallow water, this velocity is expressed by (1.11) i.e. \sqrt{gh} .

Aforesaid schemes do not simulate the flux correctly when one of eigenvalues is vanishing, i.e. when Froude number in computational domain is equal to one. This numerical error causes non-physical shock wave. Figure 2.4 displays the numerical results of dam-break problem computed by Roe scheme at time 0.1 s for the computational domain of 3 m length with the initial conditions

$$h(x, 0) = \begin{cases} 8 \text{ m}, & \text{for } x \leq 1.5 \text{ m}, \\ 0.5 \text{ m}, & \text{for } x > 1.5 \text{ m} \end{cases} \quad (2.82)$$

and $\Delta x = 0.03 \text{ m}$. There can be seen non-physical behaviour of numerical solver in the domain where $F_r = 1$.

To avoid this phenomena we need to ensure non-zero size of eigenvalues, thus correct the entropy. The size of eigenvalues can be modified as follows [24]

$$|\lambda_i|_\epsilon = |\lambda_i| + \frac{1}{2}(1 + \text{sgn}(\epsilon - |\lambda_i|)) \left(\frac{\lambda_i^2 + \epsilon^2}{2\epsilon} - |\lambda_i| \right), \quad i = 1, 2, \quad (2.83)$$

where $\epsilon = \mu_\epsilon \sqrt{gh}$ and μ_ϵ is small positive parameter smaller than one which has to be set for each particular case. Simplified way of the eigenvalue correction is [24]

$$|\lambda_i|_\epsilon = \max(|\lambda_i|, \epsilon). \quad (2.84)$$

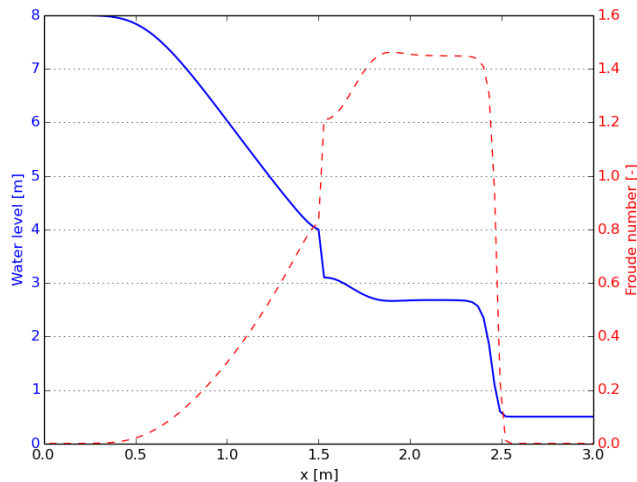


Figure 2.4: Numerical results of dam-break problem computed by Roe solver when $F_r = 1$.

Previous relations from the sections 2.2.1 and 2.2.2 have to be computed with these new eigenvalues. The simulation of the dam-break described by initial conditions (2.82) with $\mu_\epsilon = 0.2$ is shown in Figure 2.5.

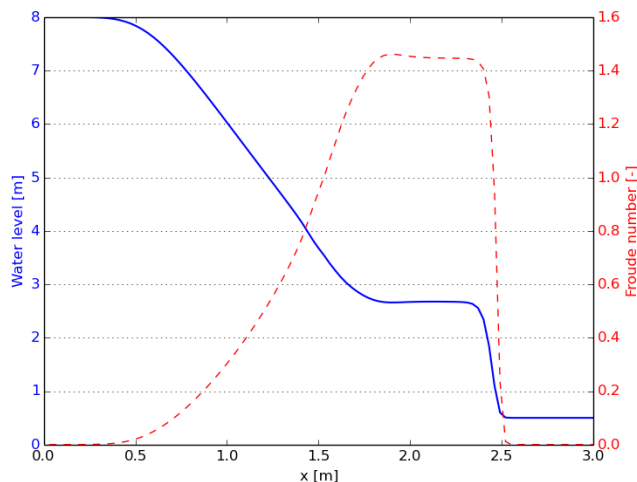


Figure 2.5: Simulation of dam-break computed by Roe solver with entropy correction.

The HLL scheme does not suffer from this phenomena so much, but the correction can improve the numerical solution around the point with $F_r = 1$ as shown in Figure 2.6.

2.2.4 Wet/Dry Interface Treatment

Computations of the numerical flux near the wet/dry interface needs special attention. Basic problem is computation of velocity when water depth h approaches zero value. The velocity is computed from the vector of conservative variables $\mathbf{W} = [h, hu]^T$ as $u = \frac{(hu)}{h}$. Numerical inaccuracy of the computers are causing that the velocity can reach non-physically large values when $h \rightarrow 0$.

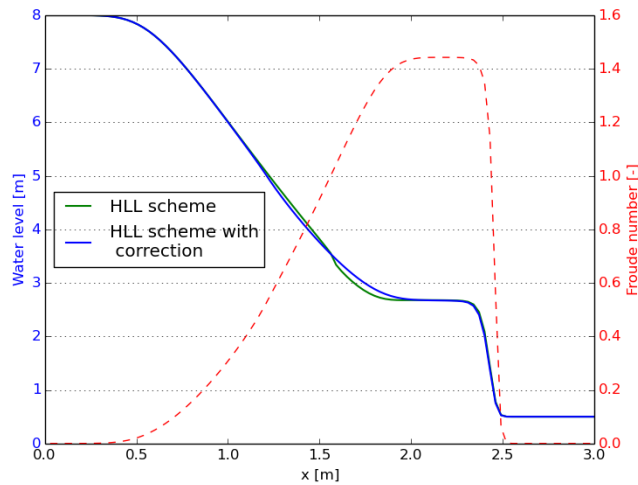


Figure 2.6: Numerical results of dam-break problem computed by HLL solver with and without the entropy correction when $F_r = 1$.

The simplest method dealing with this problem is to set some limit ϵ which is small but larger than the numerical accuracy of the computer. This tolerance defines whether a finite volume is flooded or not. All values of the vector of conservative variables are considered to be zero when water depth is smaller than this constant. According to [16] small changes of this value do not affect numerical results a lot, when computing frictionless cases. But the model becomes more sensitive to changes in ϵ for problems with bed friction parametrized by the Manning expression (see chapter 2.4 for details on Manning expression). This is because the Manning formula requires division by h , which can result in an unrealistically large prediction of the shear stress in shallow regions near wet/dry boundaries [16].

Another way how to cope with this phenomena is to set the celerity of the flow to zero in the finite volume which abuts wet/dry interface [22]. But this can hypothetically cause problems when there is no wet/dry interface in the computational domain and this domain is covered by the tiny layer of water.

The last way mentioned in this study is to use following relation [41]

$$u = \frac{h(hu)}{h^2 + \epsilon_v}, \quad (2.85)$$

where ϵ_v is small positive constant. This relation is simple and computationally undemanding, but for computations when $h > \epsilon_v$ the velocity is computed with error depending on ratio of h^2 and ϵ_v . Another similar modification of velocity computation can be [77]

$$u = \frac{\sqrt{2}h(hu)}{\sqrt{h^4 + \max(h^4, \epsilon_v)}}, \quad (2.86)$$

which computes the velocity precisely for $h^4 \geq \epsilon_v$ and in case of small water depth the velocity is altered to

$$u = \frac{hu}{h} \cdot \frac{\sqrt{2}}{\sqrt{1 + \frac{\epsilon_v}{h^4}}}. \quad (2.87)$$

When h is small, the entire algorithm remains consistent only if we recompute the discharge (hu) using

$$(hu) = h \cdot u, \quad (2.88)$$

where u on the right-hand side of the equation (2.88) is computed with the aid of the relation (2.85) or (2.86). It was proved by numerical experiments of A. Kurganova et al. [41] that otherwise the water depth can attain negative values.

Comparison of aforesaid modifications is shown and discussed in Section 7.1.1.3.

Note 1. *When none of the mentioned modifications is used the time step will degenerate because of too large local velocity at the wet/dry interface.*

2.3 Bed Slope Source Term Approximation

In this section different approaches of the bed slope source term approximation are described. Only constant cell width Δx is considered.

Function describing bed topography $B(x)$ can be approximated by piece-wise linear function. There are two possibilities where this approximate function can be evaluated. Either in the center (Figure 2.7) or at the edge (Figure 2.8) of the finite volume.

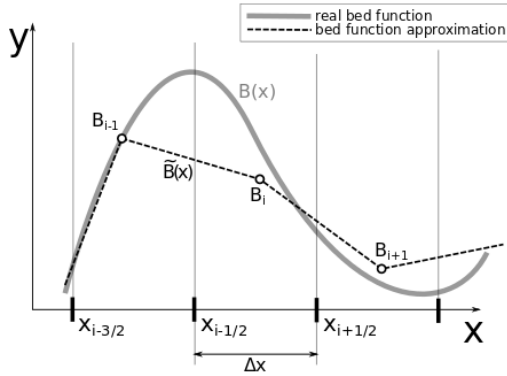


Figure 2.7: The approximation of the bed function $B(x)$ evaluated in the centre of the finite volume.

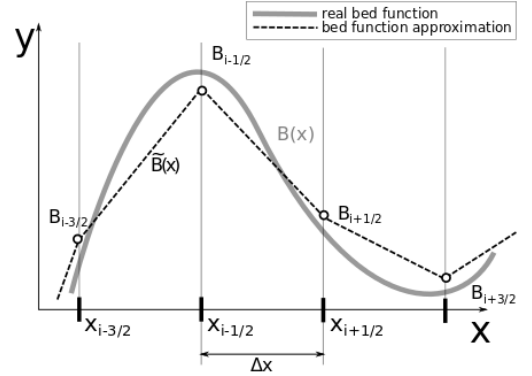


Figure 2.8: The approximation of the bed function $B(x)$ evaluated at the edges of the finite volume.

Definition 4. ('C-property' condition) *In the case when there are no other source terms, except of the bed slope source term, and for initial conditions*

$$\begin{aligned} h(x, 0) + B(x) &= d, & x \in \Omega, & d \in \mathbb{R}_{>0} \\ u(x, 0) &= 0, & x \in \Omega, \end{aligned} \quad (2.89)$$

a scheme satisfies the 'C-property' condition when

$$u(x, t) = 0 \quad \forall t \in \langle 0, \infty \rangle, \quad \forall x \in \Omega. \quad (2.90)$$

2.3.1 Cell-Centred Approximation

In this section the cell-centred approximation is taken under the consideration. Function $B(x)$, describing the bed, is approximated by the piecewise linear function $\tilde{B}(x)$ defined as

$$\tilde{B}(x) = B_i + \frac{B_{i+1} - B_i}{\Delta x} (x - x_i), \quad \forall x: x_i \leq x \leq x_{i+1}. \quad (2.91)$$

Function $\tilde{B}(x)$ is from the space of C^0 functions and point B_i which is located in the centre of the finite volume is computed in the following way

$$B_i = \frac{\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} B(x) dx}{\Delta x}. \quad (2.92)$$

2.3.1.1 Cell Centered Approximation for Roe Scheme

The bed slope source term at the finite volume edge can be computed in sense of (1.5) and spatial derivative of the bed function is approximated by differential formula using values from Ω_i and Ω_{i-1} . The bed slope source term at the edge is expressed as

$$\mathbf{R}_{i-\frac{1}{2}} = \begin{bmatrix} 0 \\ -(c_{i-\frac{1}{2}})^2 \frac{(B_i - B_{i-1})}{\Delta x} \end{bmatrix}. \quad (2.93)$$

Total bed slope source term \mathbf{S}_{b_i} in (2.9) is computed as [78]

$$\mathbf{S}_{b_i} = \mathbf{R}_{i+\frac{1}{2}}^- + \mathbf{R}_{i-\frac{1}{2}}^+. \quad (2.94)$$

Using relations from Section 2.2.2.2, the summands in this relation can be expressed as [13]

$$\begin{aligned} \mathbf{R}_{i-\frac{1}{2}}^+ &= (\mathbf{A}^+ \mathbf{A}^{-1} \mathbf{R})_{i-\frac{1}{2}} = \frac{1}{2} (\mathbf{X} (\mathbf{I} + \mathbf{\Lambda}^{-1} |\mathbf{\Lambda}|) \mathbf{X}^{-1} \mathbf{R})_{i-\frac{1}{2}}, \\ \mathbf{R}_{i+\frac{1}{2}}^- &= (\mathbf{A}^- \mathbf{A}^{-1} \mathbf{R})_{i+\frac{1}{2}} = \frac{1}{2} (\mathbf{X} (\mathbf{I} - \mathbf{\Lambda}^{-1} |\mathbf{\Lambda}|) \mathbf{X}^{-1} \mathbf{R})_{i+\frac{1}{2}}. \end{aligned} \quad (2.95)$$

Then (2.95) can be computed using $\beta_{i-\frac{1}{2}} = \mathbf{X}_{i-\frac{1}{2}}^{-1} \mathbf{R}_{i-\frac{1}{2}} = [\text{DOPOCITAT}] \begin{bmatrix} 0 \\ -(c_{i-\frac{1}{2}})^2 \frac{(B_i - B_{i-1})}{\Delta x} \end{bmatrix} = \left[\frac{c_{i-\frac{1}{2}}(B_i - B_{i-1})}{2\Delta x}, -\frac{c_{i-\frac{1}{2}}(B_i - B_{i-1})}{2\Delta x} \right]^T$ and eigenvalues (2.55) with corresponding eigenvectors (2.56) as [13]

$$\begin{aligned} \mathbf{R}_{i-\frac{1}{2}}^+ &= \frac{1}{2} \sum_{p=1}^2 [\beta_p \mathbf{x}_p (1 + \text{sgn}(\lambda_p))]_{i-\frac{1}{2}}, \\ \mathbf{R}_{i+\frac{1}{2}}^- &= \frac{1}{2} \sum_{p=1}^2 [\beta_p \mathbf{x}_p (1 - \text{sgn}(\lambda_p))]_{i+\frac{1}{2}}. \end{aligned} \quad (2.96)$$

This approximation of the source term is well balanced with Roe scheme and meets 'C-property' condition in the flooded domain. However the scheme has to be altered to satisfy 'C-property' condition also at the wet/dry interface.

Proof. The proof that Roe scheme satisfies 'C-property' in flooded computational domain is following. The condition which has to be satisfied is the equilibrium of components of numerical flux and bed slope source term, i.e. zero time change of equation (2.9).

In case of zero velocity of the flow the component (2.67) is simplified to

$$(\alpha_{1,2})_{i-1/2} = \frac{h_i - h_{i-1}}{2} \quad (2.97)$$

and the eigenvalues (2.55) are simplified to

$$(\lambda_1)_{i-1/2} = -c_{i-1/2}, \quad (\lambda_2)_{i-1/2} = c_{i-1/2}. \quad (2.98)$$

Froude number Fr is equal to zero, in case of constant water depth with $u = 0$, thus the entropy correction can be omitted. It must be proved that

$$-\frac{1}{\Delta x} (\Phi_{i+1/2} - \Phi_{i-1/2}) + (\mathbf{R}_{i+1/2}^- + \mathbf{R}_{i-1/2}^+) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (2.99)$$

The simplified relations (2.98), (2.97) and corresponding eigenvectors are substituted into the relations (2.72) and (2.96). After division by Δx , (2.99) yields

$$\begin{aligned} & \begin{bmatrix} \frac{1}{2}c_{i+1/2}(h_{i+1} - h_i) \\ -\frac{1}{4}g(h_{i+1}^2 + h_i^2) \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}c_{i-1/2}(h_i - h_{i-1}) \\ \frac{1}{4}g(h_i^2 + h_{i-1}^2) \end{bmatrix} + \\ & + \begin{bmatrix} \frac{1}{2}c_{i-1/2}(h_i - h_{i-1}) \\ -\frac{1}{4}g(h_{i-1} + h_i)(B_i - B_{i-1}) \end{bmatrix} + \begin{bmatrix} \frac{1}{2}c_{i+1/2}(h_i - h_{i-1}) \\ -\frac{1}{4}g(h_{i+1} + h_i)(B_{i+1} - B_i) \end{bmatrix}. \end{aligned} \quad (2.100)$$

In Figure 2.9 it can be seen that

$$h_i + B_i = D = h_{i-1} + B_{i-1} \quad \Rightarrow \quad B_i - B_{i-1} = h_{i-1} - h_i. \quad (2.101)$$

Now (2.100) can be rewritten into

$$\begin{bmatrix} \frac{1}{2}c_{i+1/2}(h_{i+1} - h_i) \\ -\frac{1}{4}g(h_{i+1}^2 + h_i^2) \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}c_{i-1/2}(h_i - h_{i-1}) \\ \frac{1}{4}g(h_i^2 + h_{i-1}^2) \end{bmatrix} + \begin{bmatrix} \frac{1}{2}c_{i-1/2}(h_i - h_{i-1}) \\ -\frac{1}{4}g(h_{i-1}^2 - h_i^2) \end{bmatrix} + \begin{bmatrix} \frac{1}{2}c_{i+1/2}(h_i - h_{i-1}) \\ \frac{1}{4}g(h_{i+1}^2 - h_i^2) \end{bmatrix}. \quad (2.102)$$

And the summation yields the satisfaction of the equilibrium (2.99).

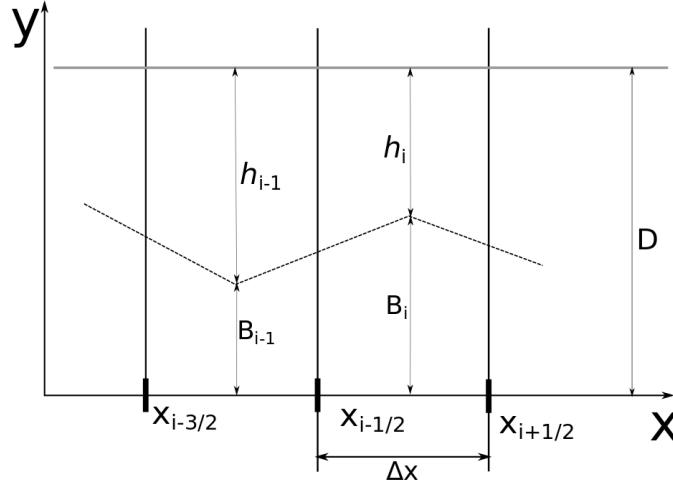


Figure 2.9: Steady state of water level with zero velocity.

□

2.3.1.1.1 Computations at the Wet and Dry Interface

Robust scheme has to keep the 'C-property' condition not only at the computational domain which is fully flooded but also at the wet/dry interface when only part of the finite volume is flooded. Approximation of the bed slope source term described previously is not able to meet this demand. The computation of the bed slope source term at this interface has to be modified to improve the scheme. Let wet/dry interface to be set between a dry cell Ω_i and a wet cell Ω_{i-1} . Part of the numerical bed

slope source term (2.94) will be then redefined in the following way [13]

$$R_{i-\frac{1}{2}} = \begin{cases} \begin{bmatrix} 0 \\ -c_{i-\frac{1}{2}}^2 \frac{h_{i-1}}{\Delta x} \end{bmatrix}, & \text{when } h_{i-1} < B_i - B_{i-1}, \\ \begin{bmatrix} 0 \\ c_{i-\frac{1}{2}}^2 \frac{h_i}{\Delta x} \end{bmatrix}, & \text{when } h_i < B_{i-1} - B_i, \\ \begin{bmatrix} 0 \\ -c_{i-\frac{1}{2}}^2 \frac{B_i - B_{i-1}}{\Delta x} \end{bmatrix} & \text{other cases.} \end{cases} \quad (2.103)$$

All computations remain unchanged except the computation of $\beta_{1,2}$. It is computed as

$$(\beta_{1,2})_{i-\frac{1}{2}} = \pm \frac{c_{i-\frac{1}{2}} \Delta B}{2\Delta x}, \quad (2.104)$$

where

$$\Delta B = \begin{cases} h_{i-1}, & \text{when } h_{i-1} < B_i - B_{i-1} \\ -h_i, & \text{when } h_i < B_{i-1} - B_i \\ B_i - B_{i-1} & \text{other cases.} \end{cases} \quad (2.105)$$

Physical meaning of this modification is sketched in Figures 2.10 and 2.11. This redefined bed slope source term doesn't produce any spurious oscillations at the wet/dry interface.

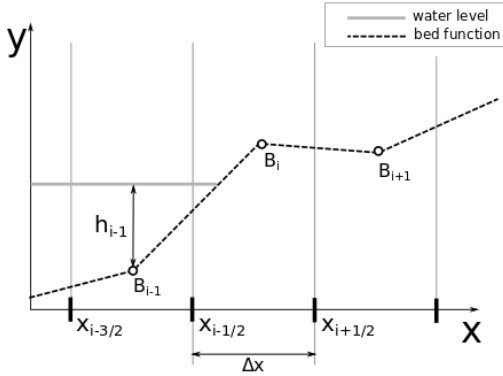


Figure 2.10: Wet/dry interface before the correction when $h_{i-1} < B_i - B_{i-1}$.

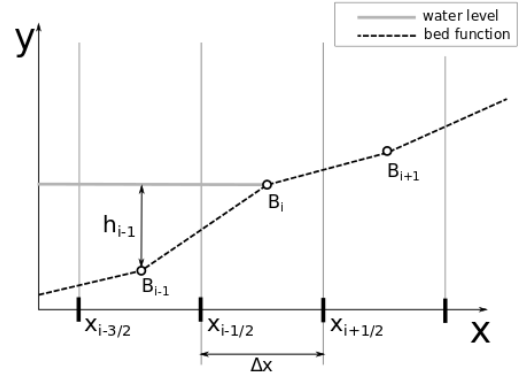


Figure 2.11: Wet/dry interface after the correction: $\Delta B = h_{i-1}$.

The Roe solver works very well within flooded domains but in certain situations, when a finite volume is getting dry, the solver can produce negative water depth. This was discussed in [22] or in [114]. One way of dealing with this problem is to compute the exact Riemann solution. But this is straightforward only for the homogeneous Shallow Water Equations when the bottom source is neglected.

Fraccarollo and Toro constructed special Riemann method for this case [79]. And Nicholas Dodd modified Roe's method to treat this deficiency correctly [80]. However, all these corrections were found to have a minor influence on the resulting solution [16].

Another way of solution is to consider the zero velocity in the finite volume which is adjacent to a dry finite volume i.e. $\mathbf{W}_{i-1} = [h_{i-1}, 0]^T$. Although this attempt is not physically correct, it can guarantee that in computational domain will not show up negative water depth caused by numerical inaccuracy [13].

2.3.1.2 Cell Centered Approximation for HLL Scheme

Aforesaid bed slope source approximation isn't well balanced with HLL flux approximation and therefore doesn't satisfy 'C-property' condition. In this work a novel computation of local wave propagation speeds (2.20) is suggested.

Instead of direct computation of eigenvalues

$$\lambda_{1,2i} = u_i \mp \sqrt{gh_i}, \quad (2.106)$$

the eigenvalues are divided into dynamic and static part and then used within the computations of local wave propagation speeds. Dynamic part λ_D containing velocity remains unchanged $(\lambda_{1,2i})_D = u_i$ meanwhile static part λ_S is computed as $(\lambda_{1,2i})_S = \mp \sqrt{g \frac{h_i + h_{i+1}}{2}}$. The modified eigenvalues

$$(\lambda_{1,2i})_m = (\lambda_{1,2i})_D + (\lambda_{1,2i})_S = u_i \mp \sqrt{g \frac{h_i + h_{i+1}}{2}} \quad (2.107)$$

are then used for the computation of local propagation speeds (2.20) in HLL scheme. This proposed modification satisfies 'C-property' condition in fully flooded domain as well as at the wet/dry interface while the dynamics of the flow is resolved with the same precision as within the original scheme as show in Table 7.1.

The usage of Roe's averaged variables (2.53), for the computation of the eigenvalues, was also tested. As a testing benchmark, 2D dam-break described in 7.1.1 was chosen. But it brought no improvement and results were slightly worse in comparison with the method suggested in this work.

2.3.2 Bed Function Evaluated at the Edges

Following approach is valid for HLL scheme and wasn't tested for Roe scheme. Bed function is approximated by the piece-wise linear function $\tilde{B}(x)$ as shown in Figure 2.8. This function is in Ω_i defined as

$$\tilde{B}(x) = B_{i-\frac{1}{2}} + \frac{B_{i+\frac{1}{2}} - B_{i-\frac{1}{2}}}{\Delta x} (x - x_{i-\frac{1}{2}}), \quad \forall x : x_{i-\frac{1}{2}} \leq x \leq x_{i+\frac{1}{2}}. \quad (2.108)$$

$\tilde{B}(x)$ is also from the space of C^0 functions. The point $B_{i+\frac{1}{2}}$ can be approximated as

$$B_{i+\frac{1}{2}} = \frac{\int_{x_i}^{x_{i+1}} B(x) dx}{\Delta x}. \quad (2.109)$$

The bed slope source term \mathbf{S}_i in (2.9) is then computed in every finite volume by simple differencing formula

$$\mathbf{S}_i = \begin{bmatrix} 0 \\ -gh_i \frac{B_{i+\frac{1}{2}} - B_{i-\frac{1}{2}}}{\Delta x} \end{bmatrix}. \quad (2.110)$$

Proof. Proof that HLL scheme meets 'C-property' condition is following. Again equilibrium between numerical flux and bed slope source term, i.e. zero time change of equation (2.9), is to be proofed.

Zero velocity u and constant water level D is considering in Ω . Water level values at the finite volume $h_{i\pm\frac{1}{2}}^{W/E}$ are shown in Figure 2.12. In the following h_{i-1}^E and h_i^W , which are identical, are denoted as $h_{i-\frac{1}{2}}$, similarly $h_i^E = h_{i+1}^W$ are denoted as $h_{i+\frac{1}{2}}$. It can be seen that $h_{i+\frac{1}{2}} = D - B_{i+\frac{1}{2}}$ and $h_{i-\frac{1}{2}} = D - B_{i-\frac{1}{2}}$. Components (2.20) are, in case of zero celerities, simplified to

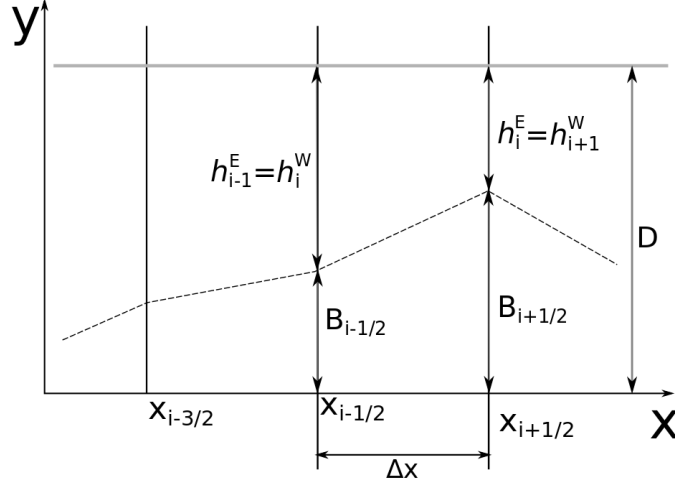


Figure 2.12: Steady state of water level. Bed approximated at the edges of finite volume

$$\begin{aligned} a_{i+\frac{1}{2}}^+ &= \max \left\{ \sqrt{gh_{i+\frac{1}{2}}^-}, \sqrt{gh_{i-\frac{1}{2}}^+}, 0 \right\} = \sqrt{gh_{i+\frac{1}{2}}}, \\ a_{i+\frac{1}{2}}^- &= \min \left\{ -\sqrt{gh_{i+\frac{1}{2}}^-}, -\sqrt{gh_{i-\frac{1}{2}}^+}, 0 \right\} = -\sqrt{gh_{i+\frac{1}{2}}}. \end{aligned} \quad (2.111)$$

Flux function (1.3) at the edge of the finite volume is

$$\mathbf{F}_{i+\frac{1}{2}} = \begin{bmatrix} 0 \\ \frac{1}{2}gh_{i+\frac{1}{2}}^2 \end{bmatrix}. \quad (2.112)$$

Substitution into (2.32) yields

$$\Phi_{i+\frac{1}{2}} = \frac{\sqrt{gh_{i+\frac{1}{2}}} \begin{bmatrix} 0 \\ \frac{1}{2}gh_{i+\frac{1}{2}}^2 \end{bmatrix} + \sqrt{gh_{i+\frac{1}{2}}} \begin{bmatrix} 0 \\ \frac{1}{2}gh_{i+\frac{1}{2}}^2 \end{bmatrix}}{2\sqrt{gh_{i+\frac{1}{2}}}} + \frac{-gh_{i+\frac{1}{2}}}{2\sqrt{gh_{i+\frac{1}{2}}}} \begin{bmatrix} h_{i+\frac{1}{2}} - h_{i+\frac{1}{2}} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{2}gh_{i+\frac{1}{2}}^2 \end{bmatrix}, \quad (2.113)$$

similarly

$$\Phi_{i-\frac{1}{2}} = \begin{bmatrix} 0 \\ \frac{1}{2}gh_{i-\frac{1}{2}}^2 \end{bmatrix}. \quad (2.114)$$

Evaluated bed slope source term (2.110) in i -th finite volume is

$$\begin{aligned} \mathbf{S}_i &= \begin{bmatrix} 0 \\ -g \left(D - \frac{B_{i+\frac{1}{2}} + B_{i-\frac{1}{2}}}{2} \right) \frac{B_{i+\frac{1}{2}} - B_{i-\frac{1}{2}}}{\Delta x} \end{bmatrix} = \begin{bmatrix} 0 \\ -g \left(\frac{D - B_{i+\frac{1}{2}} + D - B_{i-\frac{1}{2}}}{2} \right) \frac{B_{i+\frac{1}{2}} - B_{i-\frac{1}{2}}}{\Delta x} \end{bmatrix} = \\ & \begin{bmatrix} 0 \\ -g \left(\frac{h_{i+\frac{1}{2}} + h_{i-\frac{1}{2}}}{2} \right) \frac{B_{i+\frac{1}{2}} - B_{i-\frac{1}{2}}}{\Delta x} \end{bmatrix}. \end{aligned} \quad (2.115)$$

By substitution into (2.9) we get

$$\frac{\partial}{\partial t} \mathbf{W}_i = - \frac{\begin{bmatrix} 0 \\ \frac{1}{2} g h_{i+\frac{1}{2}}^2 \end{bmatrix} - \begin{bmatrix} 0 \\ \frac{1}{2} g h_{i-\frac{1}{2}}^2 \end{bmatrix}}{\Delta x} + \begin{bmatrix} 0 \\ -g \left(\frac{h_{i+\frac{1}{2}} + h_{i-\frac{1}{2}}}{2} \right) \frac{B_{i+\frac{1}{2}} - B_{i-\frac{1}{2}}}{\Delta x} \end{bmatrix}. \quad (2.116)$$

Considering

$$B_{i+\frac{1}{2}} + h_{i+\frac{1}{2}} = D = B_{i-\frac{1}{2}} + h_{i-\frac{1}{2}} \quad \Rightarrow \quad B_{i+\frac{1}{2}} - B_{i-\frac{1}{2}} = h_{i-\frac{1}{2}} - h_{i+\frac{1}{2}}, \quad (2.117)$$

the time equation can be expressed as

$$\frac{\partial}{\partial t} \mathbf{W}_i = \frac{1}{2\Delta x} g \begin{bmatrix} 0 \\ h_{i-\frac{1}{2}}^2 - h_{i+\frac{1}{2}}^2 \end{bmatrix} - \frac{1}{2\Delta x} g \begin{bmatrix} 0 \\ h_{i-\frac{1}{2}}^2 - h_{i+\frac{1}{2}}^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (2.118)$$

which means that the derivation of the vector of conservative variables is zero and HLL scheme satisfies the 'C-property' condition in flooded computational domain. \square

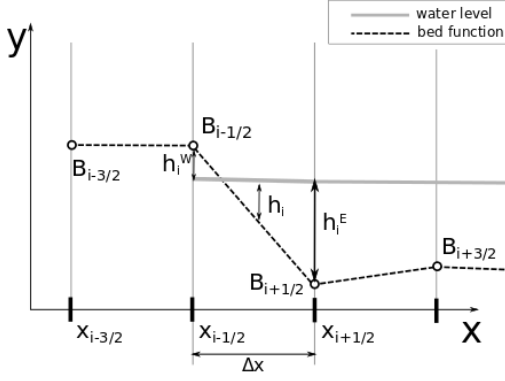
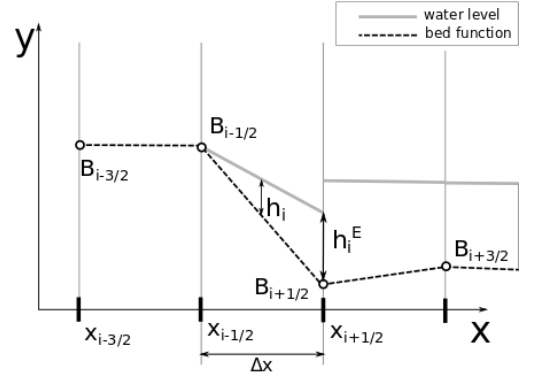
2.3.2.1 Computations at the Wet and Dry Interface

Aforesaid approach 2.3.2 can lead to the negative water depth h_i^W at the finite volume edge as shown in Figure 2.13. The value h_i^W is needed for computations of the numerical flux through the edge. To deal with this problem Kurganov and Petrova [41] suggest replacing the constant water depth by linear function at the wet/dry interface. This function is defined as follows

$$\text{if } h_i^W < 0 \Rightarrow \begin{cases} h_i^W = 0 \\ h_i^E = 2h_i \end{cases}, \quad \text{or} \quad (2.119)$$

$$\text{if } h_i^E < 0 \Rightarrow \begin{cases} h_i^W = 2h_i \\ h_i^E = 0 \end{cases}.$$

New corrected water level is shown in Figure 2.14. This scheme is not only positivity preserving but also mass conservative. The proof of the positivity preserving property of the scheme is given in [41]. But the correction causes non-physical discontinuity at the point $x_{i+\frac{1}{2}}$ and scheme doesn't satisfy 'C-property' condition [115, 119], thus in frame of this work a novel modification of the bed slope source term approximation, which will meet 'C-property' condition, is proposed.

Figure 2.13: Negative water depth h_i^W .Figure 2.14: Corrected water level with $h_i^W = 0$.

The flux across the edge $x_{i-\frac{1}{2}}$ is considered to be zero which means that total flux in the finite volume Ω_i is flux across edge $x_{i+\frac{1}{2}}$ which is, considering zero velocity and the constant water level (see Figure 2.13), equal to

$$\Phi_{i+\frac{1}{2}} = \begin{bmatrix} 0 \\ \frac{1}{2\Delta x} g (h_i^E)^2 \end{bmatrix}. \quad (2.120)$$

Bed slope source term (2.110) is equal to

$$\mathbf{S}_i = \begin{bmatrix} 0 \\ -g h_i \frac{\Delta B}{\Delta x} \end{bmatrix}. \quad (2.121)$$

The component ΔB is chosen to be modified in this work. Following equation is solved considering $h_i = \frac{h_i^W + h_i^E}{2}$

$$\begin{bmatrix} 0 \\ -g h_i \frac{\Delta B}{\Delta x} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{2\Delta x} g (h_i^E)^2 \end{bmatrix} \Rightarrow \Delta B = -\frac{(h_i^E)^2}{h_i^W + h_i^E}. \quad (2.122)$$

With this modification the bed slope source term and the flux term are well balanced and negative water depth h^W doesn't have to be changed and numerical flux $\Phi_{i-\frac{1}{2}}$ is considered to be zero. Similarly for the case when numerical flux $\Phi_{i+\frac{1}{2}}$ is considered to be zero and h^E is negative the equation (2.122) can be computed and conditions for ΔB can be summarised as

$$\Delta B = \begin{cases} -\frac{(h_i^E)^2}{(h_i^W + h_i^E)} & \text{when } h_i^W < 0, \\ \frac{(h_i^W)^2}{(h_i^W + h_i^E)} & \text{when } h_i^E < 0, \\ B_{i+\frac{1}{2}} - B_{i-\frac{1}{2}} & \text{other cases.} \end{cases} \quad (2.123)$$

Although the scheme (2.123) satisfies the 'C-property' condition, the modification is not mass conservative.

The aforementioned reconstruction methods (2.119) and (2.123) differ only in bed slope source term computations and conservative variables, which are used in the computation of the numerical flux. Herein, a novel reconstruction method is proposed, which is essentially switching between both schemes, depending whether positivity or C-property needs to be preserved and therefore combines

the advantages of both methods while overcoming their limitations. In this section a new predictor is introduced. This predictor governs the numerical approximation and chooses suitable approach for the computations at the wet/dry interface.

The predictor is based on the continuity equation which governs the mass flow. In the one-dimensional case, the continuity equation is given by

$$h_i^{n+1} = h_i^n - \frac{\Delta t}{\Delta x_i} \left(\Phi_{i+\frac{1}{2}} - \Phi_{i-\frac{1}{2}} \right) \quad (2.124)$$

where n is the time level and $\Phi_{i\pm\frac{1}{2}}$ are the first components of the numerical fluxes. At the wet/dry interface one of the numerical fluxes is zero. For the conditions illustrated in Figure 2.13 and 2.14, the flux $\Phi_{i-\frac{1}{2}}$ is zero and Equation 2.124 results in

$$h_i^{n+1} = h_i^n - \frac{\Delta t}{\Delta x_i} F_{i+\frac{1}{2}}. \quad (2.125)$$

The water depth in the $n + 1^{th}$ time level must be non-negative

$$h_i^n - \frac{\Delta t}{\Delta x_i} \Phi_{i+\frac{1}{2}} \geq 0 \quad (2.126)$$

and Equation 2.126 results in the final predictor which defines whether the bed source term and conservative variables are computed by the C-property or positivity preserving reconstruction:

$$\Phi_{i+\frac{1}{2}}, \mathbf{S}_{b_i} = \begin{cases} \text{C-property approach (2.119) if } h_i^n \Delta x_i \geq \Delta t \Phi_{i+\frac{1}{2}}, \\ \text{positivity preserving approach (2.123) in other cases.} \end{cases} \quad (2.127)$$

The case, when h_i^E is negative, follows by symmetry. The resulting reconstruction is positivity preserving and satisfies the C-property condition.

2.4 Bed Friction Source Term

Another source term which plays important role in the simulations of the real flows is the bed friction source term. This term is caused by the roughness of the bed. The friction term is described by the equation (1.6), i.e.

$$\mathbf{S}_f = \begin{bmatrix} 0 \\ -\tau_f^x \\ -\tau_f^y \end{bmatrix}$$

where τ_f^x and τ_f^y are the bed shear stresses. These stresses were firstly derived for the pipe and open channel flows but also were implemented in Shallow Water Equations. The relations are rather empirical than analytical and usually have to be obtained experimentally. It is assumed that the bed shear stress can be expressed as a quadratic function of the depth-averaged velocity multiplied by bed friction coefficient. In this study, the usage of the Manning's and Chezy bed drag formulations are investigated.

2.4.1 Chezy Formulation

Chezy formulation has origin in 1769 and was derived for steady, turbulent open channel flows when velocity across the channel was assumed to be constant [27]. According to Chezy formulation the bed shear stress is defined as [29]

$$\tau_f^i = \frac{1}{C_s^2} u^i \sqrt{u^i u^i}, \quad i = x, y \quad (2.128)$$

where C_s is Chezy bed friction coefficient which is related to the effective bed roughness height, K_s , as follows [29]

$$C_s = 5.75 \log \left(12 \frac{h}{K_s} \right). \quad (2.129)$$

2.4.2 Manning's Formulation

This formulation was introduced by Robert Manning in 1889 as an alternative to the Chezy formulation. He did a lot of experiments to determine the relationship between the various factors that effect water as it runs down hill in a flume. Manning suggests to compute bed shear stress as follows [30]

$$\tau_f^i = C_f u^i \sqrt{u^i u^i}, \quad i = x, y \quad (2.130)$$

where C_f is Manning's bed friction coefficient. This coefficient is calculated using Manning's equation [28]

$$C_f = g \frac{n^2}{\sqrt[3]{h}}, \quad (2.131)$$

where n is the Manning's bed roughness factor which varies over the computational domain.

There is considerable uncertainty in choosing effective bed roughness height K_s and Manning's bed roughness factor n . In case of the river flows, these coefficients vary not only by the reach of the river but also depend on many other factors. Some of these factors can be sinuosity of the river (i.e. fraction of the river and valley length), vegetation (thus season of the year) or speed of the flow. Some description of these coefficients can be found in [27]. Within this work the bed friction coefficients are considered to be the same in X and Y direction.

2.4.3 Bed Friction Source Term Discretization

Regardless of the chosen formulation the bed shear stress can be described as

$$\tau_f^i = C u^i \sqrt{u^i u^i}, \quad (2.132)$$

where C is the the constant dependent on the used formulation

$$C = \begin{cases} \frac{1}{C_s^2}, & \text{in case of Chezy formulation,} \\ C_f, & \text{in case of Manning's formulation.} \end{cases} \quad (2.133)$$

When wetting and drying processes appear in the numerical simulation, the bed friction source term dominates the stability of the numerical scheme due to the water height in the denominator. In [31] the authors solve the bed friction source term by splitting method. The following derivation of the friction source term for the 1D Shallow Water Equation (1.7) follows their approach. The time change of the vector of the conservative variables must be in balance with bed friction source term, i.e.

$$\frac{d\mathbf{W}}{dt} = \mathbf{S}_f \quad (2.134)$$

The friction term doesn't affect the continuity equation. So, with the substitution $hu = q$, the equation (2.134) can be written as

$$\frac{q^{n+1} - q^n}{\Delta t} = -\tau_f^{n+1} \quad (2.135)$$

and the friction term τ_f can be approximated by the Taylor series

$$\tau_f^{n+1} = \tau_f^n + \left(\frac{\partial \tau_f}{\partial q} \right)^n \Delta q + \mathcal{O}(\Delta q^2), \quad (2.136)$$

where $\Delta q = q^{n+1} - q^n$. Ignoring the higher-order terms in (2.136) and its substitution into (2.135) leads to

$$\frac{q^{n+1} - q^n}{\Delta t} = -\tau_f^n - \left(\frac{\partial \tau_f}{\partial q} \right)^n (q^{n+1} - q^n) \quad (2.137)$$

And after some rearrangement

$$q^{n+1} = q^n - \Delta t \left(\frac{\tau_f}{1 + \Delta t \frac{\partial \tau_f}{\partial q}} \right)^n = q^n - \Delta t \underbrace{\left(\frac{Cu|u|}{1 + 2\Delta t C|q|h^{-2}} \right)^n}_{S_c^n} = q^n - \Delta t S_c^n. \quad (2.138)$$

As mentioned earlier the bed friction source term needs special treatment in the cases when the water depth approaches zero value during the wetting and drying processes. In these cases we have to use following limiter [31]

$$S_c^n = \frac{q^n}{\Delta t} \quad \text{if} \quad |\Delta t S_c^n| > |q^n|. \quad (2.139)$$

If the absolute value of S_c^n is bigger than the absolute value of (hu) then the flow is stopped. Without the limiter the friction term should non-physically change the flow direction.

2.5 Linear Reconstruction

In the following section the linear reconstruction for FVM is described. The linear reconstruction improves the order of accuracy by using linear functions for conservative variables description. Higher accuracy order increases precision of results within the same mesh resolution. The reconstruction is described only for bed function approximated at the edges of finite volumes but with small adaptations works also in case of the cell-centred approximation.

Within the finite volume, the conservative variables are described by constant function (integral average). In the following this constant values is approximated by linear function i.e. for approximating polynomials (2.18) the polynomials of degree one are chosen. Conservative variables described by piece-wise constant functions are shown in Figure 2.15. Piece-wise linear functions approximating the variables are shown in Figure 2.16.

The **Surface Gradient Method** published in [81] is used to meet 'C-property' condition. This method replaces the conservative variables vector $\mathbf{W} = [h, hu]^T$ by new vector $\mathbf{W}^* = [d, hu]^T$ where d means water surface level defined as $d = h + B$. Approximation of the variables by piece-wise linear function is based on \mathbf{W}^* and then the original variable \mathbf{W} is recomputed.

This piece-wise linear function is computed as follows

$$\mathbf{W}_i^* = \mathbf{W}_i^* + (\mathbf{W}_x^*)_i (x - x_i) \quad \forall x : x_{i-\frac{1}{2}} < x < x_{i+\frac{1}{2}} \quad (2.140)$$

where the spatial derivatives

$$(\mathbf{W}_x^*)_i = [d_x, (hu)_x]_i^T \quad (2.141)$$

are at least of first order of accuracy. Vectors of the conservative variables at the edges, \mathbf{W}^{*E} and \mathbf{W}^{*W} , are then expressed as

$$\mathbf{W}_{i+\frac{1}{2}}^{*E} = \mathbf{W}_i^* + \frac{\Delta x}{2} (\mathbf{W}_x^*)_i, \quad (2.142)$$

$$\mathbf{W}_{i-\frac{1}{2}}^{*W} = \mathbf{W}_i^* - \frac{\Delta x}{2} (\mathbf{W}_x^*)_i. \quad (2.143)$$

The reconstruction increases formal order of numerical schemes by one which means that the local error of our scheme decays from $\mathcal{O}(\Delta x)$ to $\mathcal{O}(\Delta x^2)$.

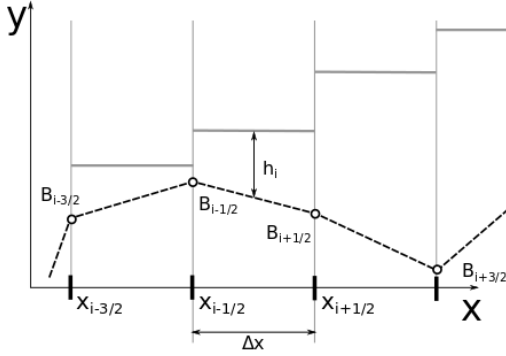


Figure 2.15: Piece-wise constant approximation of the water depth.

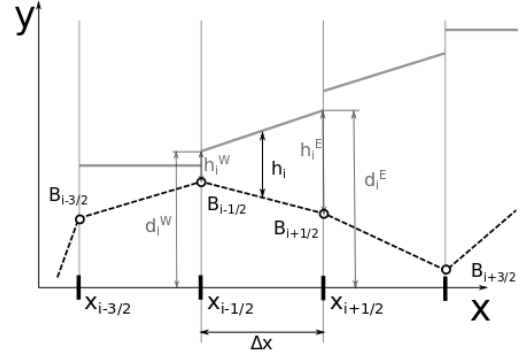


Figure 2.16: Piece-wise linear approximation of the water depth.

Note 2. *The order of accuracy of used time integration must be at least of the same order as the order of accuracy of the numerical scheme approximating the numerical flux. So Euler's method, for the time-integration, isn't suitable any more.*

Definition 5. Total variation diminishing A scheme is said to be TVD or Total Variation Diminishing if it does not increase the total variation of the solution, i.e.

$$(TV)(h^{n+1}) \leq (TV)(h^n). \quad (2.144)$$

The total variation of a grid function is defined as [82]

$$TV(u) = \sum_j |u_{j+1} - u_j|. \quad (2.145)$$

The solution is more oscillatory because of this reconstruction [81]. Thus some limiter must be used to satisfy TVD condition. One of possible limiters is 'minmod' limiter. Derivatives (2.141) are computed with the aid of this limiter [83]

$$(\mathbf{W}_x^*)_i = \text{minmod} \left(\Psi \frac{\mathbf{W}_i^* - \mathbf{W}_{i-1}^*}{\Delta x}, \frac{\mathbf{W}_{i+1}^* - \mathbf{W}_{i-1}^*}{2\Delta x}, \Psi \frac{\mathbf{W}_{i+1}^* - \mathbf{W}_i^*}{\Delta x} \right), \quad 1 < \Psi < 2. \quad (2.146)$$

Minmod function is defined as

$$\text{minmod}(z_1, z_2, z_3) = \begin{cases} \min(z_1, z_2, z_3), & \text{if } z_i > 0 \text{ for } i = 1, 2, 3, \\ \max(z_1, z_2, z_3), & \text{if } z_i < 0 \text{ for } i = 1, 2, 3, \\ 0, & \text{otherwise.} \end{cases} \quad (2.147)$$

Sometimes the derivatives are computed only by down-wind and up-wind derivatives

$$(\mathbf{W}_x^*)_i = \text{minmod} \left(\Psi \frac{\mathbf{W}_i^* - \mathbf{W}_{i-1}^*}{\Delta x}, \Psi \frac{\mathbf{W}_{i+1}^* - \mathbf{W}_i^*}{\Delta x} \right), \quad 1 < \Psi < 2. \quad (2.148)$$

with the minmod limiter defined as [84]

$$\text{minmod}(z_1, z_2) = \frac{1}{2} [\text{sgn}(z_1) + \text{sgn}(z_2)] \cdot \min(|z_1|, |z_2|). \quad (2.149)$$

Parameter Ψ can be used to control the numerical viscosity [41]. Another way of numerical viscosity controlling is the usage of the parameter ϕ in (2.140): $\mathbf{W}_i^* = \mathbf{W}_i^* + \phi \cdot (\mathbf{W}_x^*)_i(x - x_i)$. Viscosity is smaller with larger Ψ/ϕ but the solution is more oscillating.

Before the computations of the numerical fluxes the water level $h = d - B$ is recomputed. In case of bed function approximated at the edges of finite volume, the values $\mathbf{W}_i^W, \mathbf{W}_i^E$ are obtained as

$$\begin{aligned}\mathbf{W}_i^W &= [d_i^W - B_{i-\frac{1}{2}}, (hu)_i^W]^T = [h_i^W, (hu)_i^W]^T, \\ \mathbf{W}_i^E &= [d_i^E - B_{i+\frac{1}{2}}, (hu)_i^E]^T = [h_i^E, (hu)_i^E]^T.\end{aligned}\tag{2.150}$$

Reconstruction with d instead of h guaranties 'C-property' condition in flooded part of Ω , because derivatives (2.141) remain zero, in case of constant water level, and the reconstruction has no influence.

Chapter 3

Discontinuous Galerkin Method

In this chapter, the spatial discreteness will be implemented by the discontinuous Galerkin method. This method is also suitable for solving non-linear hyperbolic partial differential equations and can be implemented on both structured and unstructured meshes.

3.1 Derivation of the Method

In the following the discontinuous Galerkin method is derived. Let us consider a 1D computational area $\Omega = [a, b] \subset \mathbb{R}$ and let $t \in [0, T]$. Let $\mathcal{T} = \{\Omega_i\}_{i=1}^{n_t}$ be the discretization of the computational area where $\Omega_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ satisfies $\Omega_n \cap \Omega_m = \emptyset$ for $m \neq n$ and $\cup_i \Omega_i = \Omega$.

The solution of (1.7) is from the space $\mathbf{S}_h = S_h \oplus S_h$ such that

$$S_h = \{q \in L^2([a, b]) : q|_{\Omega_i} \in P^p(\Omega_i), i = 1, 2, \dots, n_t\}, \quad (3.1)$$

here n_t means the number of the finite elements, $P^p(\Omega_i)$ is the space of polynomials with degree less than or equal to p in Ω_i . The set $\{\varphi_i^j\}_{j=1}^{n_b}$ is a basis of the space $P^p(\Omega_i)$. 2D shallow water equations can be multiplied by the testing function $\mathbf{v}(x)$ (1.1), i.e.

$$\frac{\partial \mathbf{W}}{\partial t} \mathbf{v} + (\nabla \cdot \mathbf{F}) \mathbf{v} = \mathbf{S}_B \mathbf{v} + \mathbf{S}_f \mathbf{v}. \quad (3.2)$$

The Equation 3.2 is integrated over the subinterval Ω_i

$$\int_{\Omega_i} \frac{\partial \mathbf{W}}{\partial t} \mathbf{v} \, d\Omega + \int_{\Omega_i} (\nabla \cdot \mathbf{F}) \mathbf{v} \, d\Omega = \int_{\Omega_i} \mathbf{S}_B \mathbf{v} \, d\Omega + \int_{\Omega_i} \mathbf{S}_f \mathbf{v} \, d\Omega. \quad (3.3)$$

The second integral in (3.3) is altered in aid of integration-by-parts

$$\int_{\Omega_i} \frac{\partial \mathbf{W}}{\partial t} \mathbf{v} \, d\Omega + \oint_{\partial \Omega_i} \mathbf{F} \mathbf{v} \, \mathbf{n} \, ds - \int_{\Omega_i} \mathbf{F} \nabla \mathbf{v} \, d\Omega = \int_{\Omega_i} \mathbf{S}_B \mathbf{v} \, d\Omega + \int_{\Omega_i} \mathbf{S}_f \mathbf{v} \, d\Omega. \quad (3.4)$$

The solution of (1.7) is the union of the (3.4) over all finite elements

$$\sum_{i=1}^{n_t} \left(\int_{\Omega_i} \frac{\partial \mathbf{W}}{\partial t} \mathbf{v} \, d\Omega + \oint_{\partial \Omega_i} \mathbf{F} \mathbf{v} \, \mathbf{n} \, ds - \int_{\Omega_i} \mathbf{F} \nabla \mathbf{v} \, d\Omega \right) = \sum_{i=1}^{n_t} \left(\int_{\Omega_i} \mathbf{S}_B \mathbf{v} \, d\Omega + \int_{\Omega_i} \mathbf{S}_f \mathbf{v} \, d\Omega \right). \quad (3.5)$$

Note 3. In case of continuous solution at the elements' boundary, the term $\sum_{i=1}^{n_t} \oint_{\partial \Omega_i} \mathbf{F} \mathbf{v} \, \mathbf{n} \, ds$ is equal to zero thus discontinuous Galerkin method is equal to the finite element method and the method is congruent with finite volume method in the case when the function $\mathbf{v}(\mathbf{x}, t)$ is a constant.

3.2 Numerical Scheme of Discontinuous Galerkin Method in 1D

In this section 1D numerical scheme of DG method is derived and the following system of indexes is kept. Index i is iterator over the finite elements. Index k is iterator over the equations in the mathematical model (i.e., k varies from 1 to 2 in case of 1D and from 1 to 3 in case of 2D Shallow Water Equations). Index p iterates over the Gaussian points. Finally, indexes j and l iterate over the basis functions.

In 1D case the Equation 3.5 is simplified to

$$\sum_{i=1}^{n_t} \left(\int_{I_i} \frac{\partial \mathbf{W}}{\partial t} \mathbf{v} \, dx + [\mathbf{F} \cdot \mathbf{v}]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - \int_{I_i} \mathbf{F} \frac{\partial \mathbf{v}}{\partial x} \, dx \right) = \sum_{i=1}^{n_t} \left(\int_{I_i} \mathbf{S}_B v \, dx + \int_{I_i} \mathbf{S}_f v \, dx \right), \quad (3.6)$$

where I_i is the interval $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$.

The components $W_{i,k}$ of the solution $\mathbf{W}_i = [W_{i,1}, W_{i,2}]^T = [h_i, (hu)_i]^T$ are expressed as a linear combination of some basis functions $\varphi^l(x)$

$$W_{i,k}(x, t) = \sum_{l=1}^{n_b} w_{i,k}^l(t) \varphi_i^l(x), \quad k = 1, 2 \quad (3.7)$$

where n_b means the number of the basis functions and $w_{i,k}^l$ are time dependent coefficients of the linear combination. In the following the parameters t and x are omitted. Let us take the testing function $v(x) = v_{i,k}(x)$, $k = 1, 2$, where

$$v_{i,1} = \begin{bmatrix} \varphi_i^j(x) \\ 0 \end{bmatrix} \quad (3.8)$$

and

$$v_{i,2} = \begin{bmatrix} 0 \\ \varphi_i^j(x) \end{bmatrix} \quad (3.9)$$

then substitution of (3.7) in (3.6) yields

$$\sum_{i=1}^{n_t} \left(\int_{I_i} \sum_{l=1}^{n_b} \frac{dw_{i,k}^l}{dt} \varphi_i^l \varphi_i^j \, dx + [F_k \cdot \varphi_i^j]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - \int_{I_i} F_k \frac{\partial \mathbf{v}}{\partial x} \, dx \right) = \sum_{i=1}^{n_t} \left(\int_{I_i} S_{B,k} \varphi_i^j \, dx + \int_{I_i} S_{f,k} \varphi_i^j \, dx \right), \quad (3.10)$$

here index k means the k^{th} component of the flux \mathbf{F} or the source terms $\mathbf{S}_B, \mathbf{S}_f$. The partial derivative was replaced by normal derivative, because $w_{i,k}^l$ is only time dependent.

The basis functions $\varphi_i^l(x)$ are chosen so that they are non-zero only at the interval I_i , thus the linear combination of the solution is also non-zero only in the interval I_i as shown in Figure 3.1. Thanks to this feature, the summation $\sum_{i=1}^{n_t}$ in (3.10) vanishes.

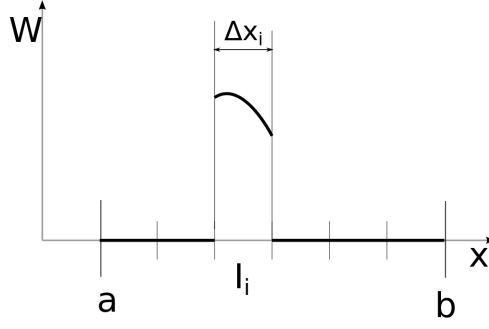


Figure 3.1: The basis function $\varphi_i^j(x)$ of the finite element $I_i \subset I = [a, b]$.

Equation (3.10) yields the system of n_b independent differential equations

$$\begin{aligned} \int_{I_i} \sum_{l=1}^{n_b} \frac{dw_{i,k}^l}{dt} \varphi_i^l \varphi_i^1 dx + [F_k \cdot \varphi_i^1]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - \int_{I_i} F_k \frac{\partial \varphi_i^1}{\partial x} dx &= \int_{I_i} S_{B,k} \varphi_i^1 dx + \int_{I_i} S_{f,k} \varphi_i^1 dx, \\ \int_{I_i} \sum_{l=1}^{n_b} \frac{dw_{i,k}^l}{dt} \varphi_i^l \varphi_i^2 dx + [F_k \cdot \varphi_i^2]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - \int_{I_i} F_k \frac{\partial \varphi_i^2}{\partial x} dx &= \int_{I_i} S_{B,k} \varphi_i^2 dx + \int_{I_i} S_{f,k} \varphi_i^2 dx, \\ &\vdots \end{aligned} \tag{3.11}$$

$$\int_{I_i} \sum_{l=1}^{n_b} \frac{dw_{i,k}^l}{dt} \varphi_i^l \varphi_i^{n_b} dx + [F_k \cdot \varphi_i^{n_b}]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - \int_{I_i} F_k \frac{\partial \varphi_i^{n_b}}{\partial x} dx = \int_{I_i} S_{B,k} \varphi_i^{n_b} dx + \int_{I_i} S_{f,k} \varphi_i^{n_b} dx.$$

The first summands of Equation 3.11 can be expressed as

$$\begin{aligned} \int_{I_i} \sum_{l=1}^{n_b} \frac{dw_{i,k}^l}{dt} \varphi_i^l \varphi_i^1 dx, \\ \int_{I_i} \sum_{l=1}^{n_b} \frac{dw_{i,k}^l}{dt} \varphi_i^l \varphi_i^2 dx, \\ \vdots \\ \int_{I_i} \sum_{l=1}^{n_b} \frac{dw_{i,k}^l}{dt} \varphi_i^l \varphi_i^{n_b} dx \end{aligned} = \mathbf{M}_i \frac{d}{dt} \mathbf{w}_{i,k} \tag{3.12}$$

where $\mathbf{w}_{i,k} = [w_{i,k}^1, w_{i,k}^2, \dots, w_{i,k}^{n_b}]^T$ and components $M_{i,l,j}$ of the mass matrix \mathbf{M}_i are computed using

Gaussian quadrature, described in Section 3.2.1,

$$M_{i,j} = \int_{I_i} \varphi_i^l \varphi_i^j dx = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \varphi_i^l \varphi_i^j dx = \dots$$

$$x = x_{i+\frac{1}{2}} \cdot \frac{\xi+1}{2} - x_{i-\frac{1}{2}} \cdot \frac{\xi-1}{2}, \xi \in [-1, 1]$$

$$dx = \frac{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}}{2} d\xi = \frac{|I_i|}{2} \cdot d\xi$$

$$\varphi_i^*(\xi) = \varphi_i(x(\xi))$$

$$\dots = \int_{-1}^1 \varphi_i^{*l} \varphi_i^{*j} \frac{|I_i|}{2} d\xi = \frac{|I_i|}{2} \sum_{p=1}^n \varpi_p \varphi_i^{*l}(\xi_p) \varphi_i^{*j}(\xi_p). \quad (3.13)$$

here ϖ_p and ξ_p means weights and points of the Gaussian quadrature and functions $\varphi_i^{*j/l}$ are the mapping of the functions $\varphi_i^{j/l}$ from the interval I_i to the interval $[-1, 1]$. The third summand in (3.11) is also computed in aid of Gaussian quadrature

$$\int_{I_i} F_k(w(x, t)) \frac{d\varphi_i^j}{dx} dx = \int_{-1}^1 F_k(w(x(\xi), t)) \frac{d\varphi_i^{*j}}{d\xi} \underbrace{\frac{d\xi}{dx}}_{\frac{2}{|I_i|}} \frac{|I_i|}{2} d\xi \approx \sum_{p=1}^n \varpi_p F_k(w(x(\xi_p), t)) \frac{d\varphi_i^{*j}}{d\xi}. \quad (3.14)$$

Similar approach is for the bed source terms

$$\int_{I_i} S_{B/f,k}(x) \varphi_i^j(x) dx \approx \sum_{p=1}^n \varpi_p S_{B/f,k}(\xi_p) \varphi_i^{*j}(\xi_p). \quad (3.15)$$

The flux F_k in the second summands is computed by a Riemann solver such that

$$\begin{aligned} F_k(x_{i+\frac{1}{2}}) &\approx \Phi_k(\mathbf{W}_i, \mathbf{W}_{i+1}), \\ F_k(x_{i-\frac{1}{2}}) &\approx \Phi_k(\mathbf{W}_{i-1}, \mathbf{W}_i). \end{aligned} \quad (3.16)$$

Previously described schemes (2.32) or (2.72) are appropriate for approximation (3.16).

Now (3.11) can be written as

$$\mathbf{M}_i \frac{d}{dt} \mathbf{w}_{i,k} + \begin{bmatrix} [\Phi_k \cdot \varphi_i^1]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - \int_{I_i} F_k \frac{\partial \varphi_i^1}{\partial x} dx \\ [\Phi_k \cdot \varphi_i^2]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - \int_{I_i} F_k \frac{\partial \varphi_i^2}{\partial x} dx \\ \vdots \\ [\Phi_k \cdot \varphi_i^{n_b}]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - \int_{I_i} F_k \frac{\partial \varphi_i^{n_b}}{\partial x} dx \end{bmatrix} = \begin{bmatrix} \int_{I_i} S_{B,k} \varphi_i^1 dx \\ \int_{I_i} S_{B,k} \varphi_i^2 dx \\ \vdots \\ \int_{I_i} S_{B,k} \varphi_i^{n_b} dx \end{bmatrix} + \underbrace{\begin{bmatrix} \int_{I_i} S_{f,k} \varphi_i^1 dx \\ \int_{I_i} S_{f,k} \varphi_i^2 dx \\ \vdots \\ \int_{I_i} S_{f,k} \varphi_i^{n_b} dx \end{bmatrix}}_{\mathbf{RHS}_{i,k}^f} \quad (3.17)$$

or after some rearrangement

$$\mathbf{M}_i \frac{d}{dt} \mathbf{w}_{i,k} = \mathbf{RHS}_{i,k}(\mathbf{w}_{i,k}) + \mathbf{RHS}_{i,k}^f(\mathbf{w}_{i,k}). \quad (3.18)$$

The time integration of the semi discrete scheme

$$\frac{d}{dt} \mathbf{w}_{i,k} = \mathbf{M}_i^{-1} \mathbf{RHS}_{i,k}(\mathbf{w}_{i,k}) \quad (3.19)$$

is analogical to the time integration of the finite volume method described in Chapter 4. The time integration of the bed friction source term needs special treatment which is described in Section 3.2.4.

3.2.1 Gaussian Quadrature for Curve Integrals

Numerical integration of the polynomials over an interval $I = [-1, 1]$ can be done in aid of Legendre-Gauss quadrature integral approximation

$$\int_{-1}^1 f(x)dx = \sum_{p=1}^{\infty} \varpi_p f(x_p) \approx \sum_{p=1}^{n_p} \varpi_p f(x_p) \quad (3.20)$$

where ϖ_p are weights, $x_p \in [-1, 1]$ and n_p is the number of integration points. The integral is approximated exactly if the number n_p satisfies

$$n_p \leq 2M - 1 \quad (3.21)$$

where M means the order of polynomial function $f(x)$. The integration rule (3.20) can be converted to arbitrary interval $[a, b]$

$$\int_a^b f(x)dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{b-a}{2}x_i + \frac{b+a}{2}\right) dx \approx \frac{b-a}{2} \sum_{p=1}^{n_p} \varpi_p f\left(\frac{b-a}{2}x_i + \frac{b+a}{2}\right) dx. \quad (3.22)$$

The table of some weights and integrating points can be seen in Table 3.1.

3.2.2 Limiting Process

The solution of the DGFEM is, in case of higher order of accuracy, oscillating around the shocks and has to be limited. First way of the solution limiting is adding artificial viscosity into the mathematical model. Second method is usage of the classical limiters adopted from the FVM. Both methods need some criterion of defining 'troubled' cells as described later on.

Scientific literature describes two types of limiting processes. The process when the limiter is applied to all finite volume cells is called the **global limiting process**. This is natural for the FVM when all the variables are described by the piecewise linear functions. Accordingly to [85], in case of the higher order of accuracy the global limiting strategy may decrease the formal accuracy of a numerical scheme not only around the shocks and local extremes (which is accepted) but also within the smooth differentiable regions of the solution (which is unwelcome). This was also confirmed by the simulations of the homogeneous Euler equations by Krivodonova et al. [51]. In case of **local limiting process** the solution is limited only in the problematic cells also called as 'troubled' cells.

3.2.2.1 Artificial Viscosity

The idea of adding artificial viscosity in to the mathematical model was adopted from the finite element method. The artificial viscosity ϵ is increased in the finite volume cell Ω_i if this cell meets the criteria of being troubled. The solution in the cells where the artificial viscosity is added is less oscillating.

3.2.2.1.1 Defining Troubled Cells

The 'troubled' cell is located by the residuum R_k defined as [86]

$$R_k(W_{i,k}) = \frac{\int_{\Omega_i} (W_{i,k} - \overline{W}_{i,k})^2 d\Omega}{\int_{\Omega_i} (W_{i,k})^2 d\Omega}, \quad \overline{W}_{i,k} = \int_{\Omega_i} W_{i,k} d\Omega \quad (3.23)$$

When the residuum $R(\rho)$ is larger than some small positive value then the artificial viscosity ϵ is increased in Ω_i .

3.2.2.1.2 Adding Artificial Viscosity into the Mathematical Model

To add artificial viscosity, the Equations 1.7 have to be transformed into the parabolic form. The Shallow Water Equations including artificial viscosity without the source terms are

$$\frac{\partial \mathbf{W}(\mathbf{x}, t)}{\partial t} + \nabla \mathbf{F}(\mathbf{W}(\mathbf{x}, t)) = \epsilon \frac{\partial^2 \mathbf{W}(\mathbf{x}, t)}{\partial \mathbf{x}^2}, \mathbf{x} \in \Omega, \quad (3.24)$$

here ϵ is a small constant governing the viscosity.

3.2.2.1.3 The Numerical Solution of the Artificial Viscosity

As in (3.2) the right hand side of the equation (3.24) is multiplied by the testing function φ . Integration over the finite element and integration-by-parts yields

$$\int_{\Omega_i} \epsilon \frac{\partial^2 \mathbf{W}(\mathbf{x}, t)}{\partial \mathbf{x}^2} \varphi \, d\Omega = \oint_{\partial \Omega_i} \epsilon \frac{\partial \mathbf{W}}{\partial \mathbf{x}} \varphi \, ds - \int_{\Omega_i} \epsilon \frac{\partial \mathbf{W}}{\partial \mathbf{x}} \frac{\partial \varphi}{\partial \mathbf{x}} \, d\Omega \quad (3.25)$$

Derivative $\frac{\partial \mathbf{W}}{\partial \mathbf{x}}$ in the curve integral from (3.25) is approximated using the average

$$\frac{\partial \mathbf{W}}{\partial \mathbf{x}} = \frac{1}{2} \left(\left(\frac{\partial \mathbf{W}}{\partial \mathbf{x}} \right)_R + \left(\frac{\partial \mathbf{W}}{\partial \mathbf{x}} \right)_L \right), \quad (3.26)$$

where indexes L,R means the derivatives in the adjacent finite elements. And the spatial derivative of k^{th} conservative variable in Ω_i (denoted as $W_{i,k}$) is computed as

$$\frac{\partial W_{i,k}(x, t)}{\partial \mathbf{x}} = \sum_{j=1}^{nb} w_{i,k}^j(t) \frac{\partial \varphi_i^j(\mathbf{x})}{\partial \mathbf{x}}. \quad (3.27)$$

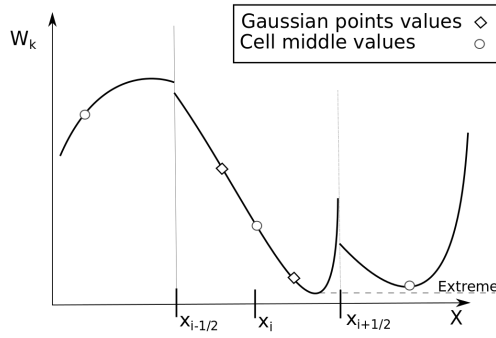
3.2.2.2 Slope Limiters

The classical FVM slope limiters can be adopted also by by DGFEM as described in the following section. One of the local limiting method is 'local limiting for the second-order moments' published by Yang and Wang [87]. The cell is called non-problematic if the local solution is locally-monotonic and bounded by means. Yang and Wang employed the TVD-minmod limiter as a monotonicity indicator. Kesserwani and Liang [52] used the local approximate solution and its derivative evaluations at the local Gaussian-points. Their approach is followed hereafter.

3.2.2.2.1 Defining Troubled Cells

One of the criteria of 'troubled' cell is met when the solution isn't 'bounded by means'. This means that the solution values must be bounded in the interval $[W_{i,k}^{min}, W_{i,k}^{max}]$ where $W_{i,k}^{min} = \min(W_{i-1,k}, W_{i,k}, W_{i+1,k})$ and $W_{i,k}^{max} = \max(W_{i-1,k}, W_{i,k}, W_{i+1,k})$ are maximal and minimal values in the centres of adjacent cells. Of course it is not numerically possible to check whole solution and we can do so only in finite number of points. In [52], the Gaussian points (x_p) and points at the edges of the cell ($x_{i \pm \frac{1}{2}}$) are chosen to be checked. It means that the cell is untroubled if

$$W_{i,k}^{min} < W_{i,k}(x_p) < W_{i,k}^{max} \quad (3.28)$$

Figure 3.2: Local solutions of $W_k(x, t)$.

and

$$W_{i,k}^{min} < W_{i,k}(x_{i\pm\frac{1}{2}}) < W_{i,k}^{max}. \quad (3.29)$$

But, for the strongly curved solution (see Figure 3.2), the solution extreme doesn't have to be in the Gaussian points or at the cell edge. To deal with this problem Kesserwani proposed criterion of the monotonicity [52]. This criterion defines the cell as untroubled if all derivatives of the local solution in the Gaussian points and at the cell edges have the same sign.

Another criterion of defining troubled cell is for example criterion of 'discontinuity-detection indicator'. As discontinuous Galerkin method (DGFEM) solutions across inter-cells boundaries always show strong superconvergence phenomena [88], Krivodonova et al. [51] introduced discontinuity-detector defined as

$$DS_{i+\frac{1}{2}} = \frac{|W_{i,k}(x_{i+\frac{1}{2}}) - W_{i+1,k}(x_{i+\frac{1}{2}})|}{|\Delta x/2|^{3/2} |W_{i,k}(x_i)|}, \quad (3.30)$$

and $DS_{i-\frac{1}{2}}$ follows by symmetry. Krivodonova suggests to limit the solution when $DS_{i\pm\frac{1}{2}}$ exceeds unity. But as Kesserwani claims, unity is not reliable and in [52] there is suggested to limit the solution when $DS_{i\pm\frac{1}{2}}$ exceeds 100. And as the numerical results show [52, 87], this troubled cell indicator is reliable only for solutions described by lower order of accuracy.

3.2.2.2.2 Slope Limiting Process

When the troubled cells are located, the limiting process has to be applied. In [89] Cockburn and Shu made use of the Legendre polynomials. The coefficients of the first order polynomials $w_{i,1}$ are computed by the minmod limiter, the coefficients $w_{i,0}$ are left unchanged and coefficients of the higher order polynomials are set to zero. Thanks to the Legendre polynomials the method is still mass conservative, because Legendre polynomials are orthogonal at the interval $\langle -1, 1 \rangle$. Minmod limiter is used for limiting of the solution also for WENO schemes in [90].

Biswas et al. in [91] suggested to use minmod limiter not only for the low-order coefficients but also for the higher ones. The limiter they suggested is

$$w_{i,k}^l = \frac{1}{2l+1} \minmod(w_{i,k}^l, w_{i,k}^{l-1} - w_{i-1,k}^{l-1}, w_{i+1,k}^{l-1} - w_{i,k}^{l-1}), \quad (3.31)$$

minmod function is defined by (2.147). Firstly, higher-order coefficients are limited and then limiting continue with lower-order coefficients. Once the lower-order coefficients are limited, the procedure is repeated and the higher-order coefficients are re-limited using the updated low-order coefficients.

3.2.2.3 Novel Limiting Method

In previous sections different criteria for defining troubled cells and limiting processes were described. Advantages and weaknesses of these processes are discussed in the following section. In this section, the novel limiting method is suggested. This method is unique in troubled cell definition and different criteria for limiting the water depth and the discharge.

Artificial viscosity was used to limit the solution for example in [92, 93]. In this work the problems of aerodynamics were solved without appearance of the vacuum, so the situation when the air density approaches zero value was not discussed. However the situation when the water depth in SWE approaches zero value is common and DGFEM brings the following problem. In picture 7.1 the initial conditions of the 1D dam break on dry bed are shown. The water depth $h_0 = 0.6 \text{ m}$ and length $x_0 = 0.6 \text{ m}$. As the basis functions were used first three Legendre polynomials. In Figure 3.3, first time iteration of the numerical simulation for computational area of length $l_0 = 1$ and ten finite volume cells is shown. DGFEM is not monotone by nature and detail view in Figure 3.3 shows that the solution contains the negative values of the water depth even if the values at the edges are positive and the solution does not corresponds to the shape of the analytical solution obtained by (7.1). Because the artificial viscosity cannot directly affect the slope of the solution, novel limiting method, proposed in this work, is based on slope limiters.

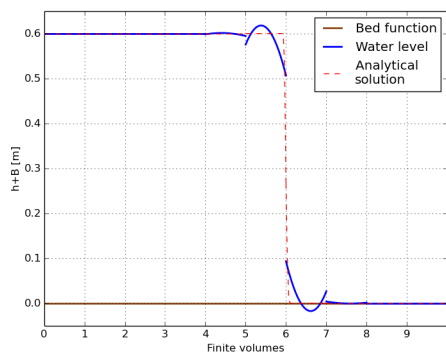


Figure 3.3: Non-limited water depth after the first time iteration.

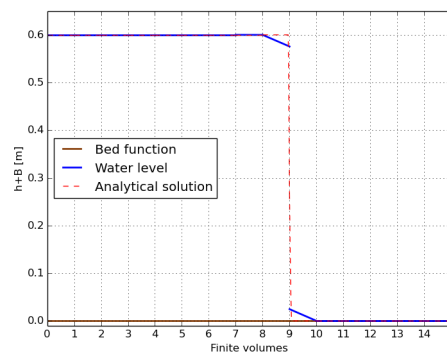


Figure 3.4: Limited water depth after the first time iteration.

Within this work the cell is considered to be troubled if some of the water depth value $h(x, t)$ in the middle of the cell is not bounded by the water depth values at the edges, i.e.

$$\max\left(h(x_{i-\frac{1}{2}}, t), h(x_{i+\frac{1}{2}}, t)\right) < h(x_i, t) \quad \text{or} \quad \min\left(h(x_{i-\frac{1}{2}}, t), h(x_{i+\frac{1}{2}}, t)\right) > h(x_i, t) \quad (3.32)$$

It is also necessary to guarantee non-negative water depth around the wet/dry interface. Thus the cell is also defined as troubled if water depth at an edge of the finite volume cell or in the Gaussian point are negative.

When the cell is troubled, the *minmod* limiter (2.147) is used for coefficients of the first order polynomials $w_{i,k}^2$. The coefficients $w_{i,k}^1$ are left unchanged and other coefficients are set to zero

$$w_{i,k}^l = 0, \quad l = 3, 4, \dots, n_b. \quad (3.33)$$

The positivity of given reconstruction is guaranteed by modification (2.119). The Legendre polynomials

are orthogonal to each other, including unity, thus the integral average of a conservative variable

$$\int_{\Omega_i} W_{i,k}(x,t)d\Omega_i = w_{i,k}^1 \int_{\Omega_i} 1d\Omega_i + w_{i,k}^2 \underbrace{\int_{\Omega_i} 1 \cdot x d\Omega_i}_{=0} + w_{i,k}^3 \underbrace{\int_{\Omega_i} 1 \cdot \frac{1}{2}(-1 + 3x^2)d\Omega_i}_{=0} + \underbrace{\dots}_{=0} \quad (3.34)$$

remains unchanged by (3.33) and limiting method is conservative. The discharge is limited in the same way as the water depth, naturally without the corrections guaranteeing positivity of the water depth. Limited solution can be see in Figure 3.4.

In some cases non-limited solution or solution limited only around wet/dry interface gives better results than solution limited in all troubled cells (tested for 2D dam break 7.1.1 and planar surface in parabolic bowl 7.1.3 benchmarks). The limiting impact of given scheme can be diminished if we allow the midpoint value to exceed the edge value by some constant ϵ_e . Then the definition of the trouble cell (3.32) gets the form

$$\max\left(h(x_{i-\frac{1}{2}}, t), h(x_{i+\frac{1}{2}}, t)\right) + \epsilon_e < h(x_i, t) \quad \text{or} \quad \min\left(h(x_{i-\frac{1}{2}}, t), h(x_{i+\frac{1}{2}}, t)\right) - \epsilon_e > h(x_i, t) \quad (3.35)$$

But the aim of this work is invention of some 'constantless' limiter and the improvement is not remarkable, this modification was not used during the computations. The comparison of the novel limiting method with method published in [52] can be found in Section 7.1.1.2.

In case of the second order DGFEM, the solution is approximated by linear functions and thus the middle value is bounded by the values at the edges and no cell is defined as 'troubled'. In this case the solution can be limited by the global limiting process in every single cell as within the FVM. The global limiting process decreases the orders of accuracy mainly in case of higher order approximations.

Numerical simulations also shown, that combination of slope limiters (for limiting water depth) and artificial viscosity (for discharge limiting) brought worse results and convergence of the scheme.

3.2.3 Bed Slope Source Term

The bed function B in i^{th} finite element is approximated by approximating function \tilde{B} as

$$\tilde{B}_i(x) = \sum_{l=1}^{n_{b,B}} w_{B,i}^l \varphi_i^l(x), \quad x \in (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) \quad (3.36)$$

here $n_{b,B}$ means the number of basis functions describing the bed topography. The bed function coefficients $w_{B,i}^l$ are constant in time and different from the coefficients describing the water depth. The bed slope source term from the Equation 3.17 is approximated as

$$\int_{\Omega_i} -gh(x,t) \frac{\partial \tilde{B}}{\partial x} \varphi_i^j d\Omega = \int_{\Omega_i} -gh(x,t) \underbrace{\sum_{l=1}^{n_{b,B}} w_{i_0}^l \frac{\partial \varphi_i^l}{\partial x}}_{\frac{\partial \tilde{B}}{\partial x}} \varphi_i^j d\Omega, \quad j = 1, 2, \dots, n_b. \quad (3.37)$$

Some optimization method can be used for the setting of the coefficients $w_{i_0}^l$, but the method must guarantee the continuity of the bed functions at the cell edge, i.e.

$$\tilde{B}_i(x_{1+\frac{1}{2}}) = \tilde{B}_{i+1}(x_{1-\frac{1}{2}}), \quad \forall i = 0, 1, 2, \dots, n_t. \quad (3.38)$$

The integral in (3.37) is also computed by the Gaussian quadrature described in Section 3.2.1. The initial conditions 'lake in rest' of the water depth $h_i = W_{i,1} = \sum_{l=1}^{n_b} w_{i,1}^l \varphi_i^l$, with the initial water level

h_0 , can be set as

$$\begin{aligned} w_{i,1}^1 &= h_0 - w_{B,i}^1, \\ w_{i,1}^l &= -w_{B,i}^l, \quad \text{for } l = 2, 3, \dots, n_{b,B}. \\ w_{i,1}^l &= 0, \quad \text{for } l = n_{b,B} + 1, \dots, n_b. \end{aligned} \quad (3.39)$$

3.2.4 Bed Friction Source Term

The friction term is represented by the bed friction source term

$$\mathbf{S}_f = \begin{bmatrix} 0 \\ -\tau_f^x \end{bmatrix} \quad (3.40)$$

where τ_f^x is the bed shear stress. In Section 2.4, Manning's and Chezy formulations were described. Within this work Manning's formulation is implemented. The bed shear stress is computed by (2.130) i.e.

$$\tau_f^x = C_f u |u|,$$

where C_f is Manning's bed friction coefficient computed by Manning's expression (2.131)

$$C_f = g \frac{n^2}{\sqrt[3]{h}}$$

here n is the Manning's bed roughness factor which varies over the computational domain.

The bed friction source term

$$\int_{\Omega_i} -\tau_f^x \varphi_i^j d\Omega = - \int_{\Omega_i} C_f u |u| \varphi_i^j d\Omega, \quad j = 1, 2, \dots, n_b. \quad (3.41)$$

is computed in aid of Gaussian quadrature.

As discussed in Section 2.4, the bed friction source term needs special treatment when the bed friction attains large values. Within this work, a limiter similar to (2.139) is introduced and the solution is limited as

$$\mathbf{w}_{i,k}^{n+1} = \begin{cases} \mathbf{w}_{i,k}^* & \text{if } \text{sign} \left(W_{i,k} \left(\mathbf{w}_{i,k}^n \right) \right) = \text{sign} \left(W_{i,k} \left(\mathbf{w}_{i,k}^* \right) \right), \\ 0 & \text{otherwise,} \end{cases} \quad \text{for } k = 2 \quad (3.42)$$

where $\mathbf{w}_{i,k}^* = \mathbf{w}_{i,k}^n + \Delta t \mathbf{M}_i^{-1} \mathbf{RHS}_{i,k}^f$. Equation 3.42 de facto means that the discharge $W_{i,2} = (hu)_i$ is set to zero in the case when the bed friction term changes the flow direction.

3.3 Numerical scheme of Discontinuous Galerkin Method in 2D

In the following section, discontinuous Galerkin method is applied to the 2D model of the Shallow Water Equations. The numerical method is described for unstructured triangular meshes.

Within the finite element Ω_i , the vector of conservative variables (1.2) is also described by the linear combination of the time dependent coefficients $w_{i,k}^l(t)$ and space dependent basis functions $\varphi_i^l(x, y)$

$$W_{i,k}(x, y, t) = \sum_{l=1}^{n_b} w_{i,k}^l(t) \varphi_i^l(x, y), \quad k = 1, 2, 3. \quad (3.43)$$

One of the possible systems of basis functions are the Taylor basis system

$$\begin{aligned}
\varphi_1(x, y) &= 1, \\
\varphi_2(x, y) &= x - C_x, \\
\varphi_3(x, y) &= y - C_y, \\
\varphi_4(x, y) &= (x - C_x)^2, \\
\varphi_5(x, y) &= (x - C_x)(y - C_y), \\
\varphi_6(x, y) &= (y - C_y)^2, \\
\varphi_7(x, y) &= (x - C_x)^3, \\
\varphi_8(x, y) &= (x - C_x)^2(y - C_y), \\
\varphi_9(x, y) &= (x - C_x)(y - C_y)^2, \\
\varphi_{10}(x, y) &= (y - C_y)^3, \\
&\vdots
\end{aligned} \tag{3.44}$$

where C_x , C_y respectively, is X-coordinate, Y-coordinate respectively, of the element's center. The functions can be ordered in Pascal triangle

$$\begin{array}{ccccccc}
& & & & & & 1 \\
& & & & & & x - C_x & & y - C_y \\
& & & & & & (x - C_x)^2 & & (x - C_x)(y - C_y) & & (y - C_y)^2 \\
& & & & & & (x - C_x)^3 & & (x - C_x)^2(y - C_y) & & (x - C_x)(y - C_y)^2 & & (y - C_y)^3
\end{array} \tag{3.45}$$

Figure 3.5 shows these functions. For the first order of accuracy, only first basis function $\varphi_1(x, y)$ is used in (3.43), thus $n_b = 1$. The usage of the first three basis functions ($n_b = 3$) yields the second order of accuracy. To increase the order of accuracy, next line of the 'basis functions triangle' (3.45) must be added.

If the basis functions, describing the solution in i^{th} finite element Ω_i , are chosen such that they are non-zero only in Ω_i (see Figure 3.6) the sum $\sum_{i=1}^{n_t}$ in Equation 3.5 can be omitted. If the approximating functions φ_k for $k = 1, 2, \dots, n_b$ are chosen for the testing function v , then (3.5) yields

$$\mathbf{M}_i \frac{d}{dt} \mathbf{w}_{i,k} + \begin{bmatrix} \oint_{\partial\Omega_i} \Phi_k \cdot \varphi_i^1 dl - \int_{\Omega_i} F_k \frac{\partial \varphi_i^1}{\partial x} d\Omega \\ \oint_{\partial\Omega_i} \Phi_k \cdot \varphi_i^2 dl - \int_{\Omega_i} F_k \frac{\partial \varphi_i^2}{\partial x} d\Omega \\ \vdots \\ \oint_{\partial\Omega_i} \Phi_k \cdot \varphi_i^{n_b} dl - \int_{\Omega_i} F_k \frac{\partial \varphi_i^{n_b}}{\partial x} d\Omega \end{bmatrix} = \begin{bmatrix} \int_{\Omega_i} S_{B,k} \varphi_i^1 d\Omega \\ \int_{\Omega_i} S_{B,k} \varphi_i^2 d\Omega \\ \vdots \\ \int_{\Omega_i} S_{B,k} \varphi_i^{n_b} d\Omega \end{bmatrix} + \underbrace{\begin{bmatrix} \int_{\Omega_i} S_{f,k} \varphi_i^1 d\Omega \\ \int_{\Omega_i} S_{f,k} \varphi_i^2 d\Omega \\ \vdots \\ \int_{\Omega_i} S_{f,k} \varphi_i^{n_b} d\Omega \end{bmatrix}}_{\mathbf{RHS}_{i,k}^f} \tag{3.46}$$

where $\mathbf{M}_i = [M_{l,j}]$ and

$$M_{l,j} = \int_{\Omega_i} \varphi_i^l \varphi_i^j d\Omega_i. \tag{3.47}$$

The volume integral is approximated by the Gaussian quadrature described in Section 3.3.1 and the curve integral is computed by the numerical method described in previous section. Similarly to (3.18),

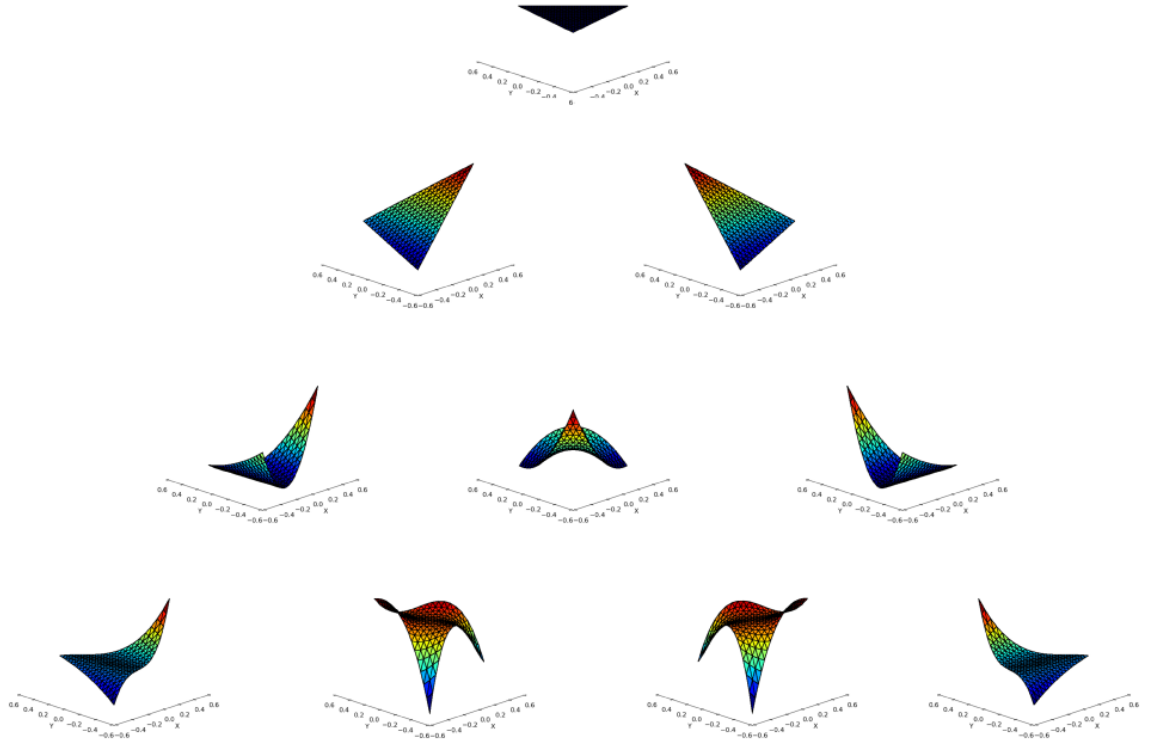


Figure 3.5: The system of Taylor basis functions $\varphi_k(x, y)$.

Equation 3.46 can be rewritten as

$$\mathbf{M}_i \frac{d}{dt} \mathbf{w}_{i,k} = \mathbf{RHS}_{i,k}(\mathbf{w}_{i,k}) + \mathbf{RHS}_{i,k}^f(\mathbf{w}_{i,k}). \quad (3.48)$$

The time integration of the semi discrete scheme

$$\frac{d}{dt} \mathbf{w}_{i,k} = \mathbf{M}_i^{-1} \mathbf{RHS}_{i,k}(\mathbf{w}_{i,k}) \quad (3.49)$$

is also analogical to the time integration of the finite volume method described in Chapter 4. The time integration of the bed friction source term needs special treatment which is described in Section 3.3.4.

3.3.1 Gaussian Quadrature for 2D Triangular Elements

There are two kinds of integrals in (3.5). The curve integral over the finite element edge was described in Section 3.2.1. The volume integral is computed as follows.

3.3.1.1 Barycentric Coordinates

The location of a point in a triangle can be described by the barycentric coordinates. The barycentric coordinates of a triangle are given by the triples of numbers (b_1, b_2, b_3) . The location of a point in

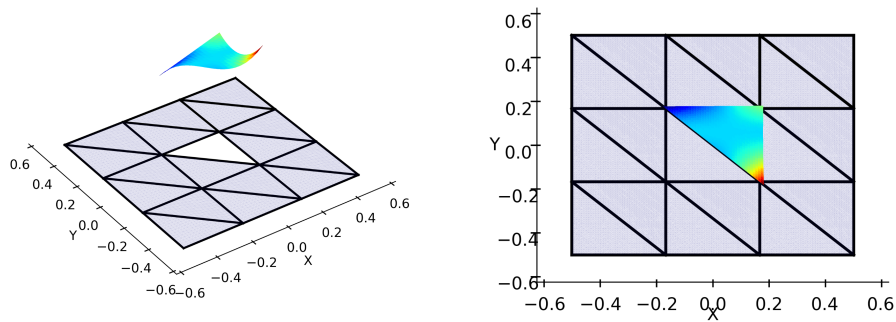


Figure 3.6: The basis function $\varphi_i^j(x, y)$ of the finite element $\Omega_i \subset \Omega = [-0.5 \times 0.5, -0.5 \times 0.5]$.

Cartesian coordinates are then expressed as

$$\begin{aligned} x_b &= b_1x_1 + b_2x_2 + b_3x_3 \\ y_b &= b_1y_1 + b_2y_2 + b_3y_3 \end{aligned} \tag{3.50}$$

where x_i and y_i are Cartesian coordinates of the triangle vertices. The vertices of the triangle have the coordinates $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$. The centre of mass of a triangle has the coordinates $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Some of other points in the triangle can be seen in Figure 3.7.

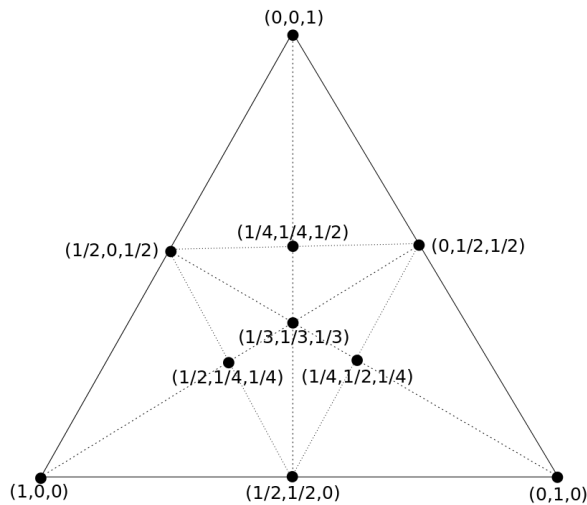


Figure 3.7: Barycentric coordinates.

3.3.1.2 Gaussian Quadrature for Volume Integral

Similarly to the curve integral, the volume integral can be approximated by the summation of the weights and functional values in the barycentric coordinates

$$\int_{\Omega_i} f(x, y) d\Omega_i \approx \sum_{p=1}^{n_p} \varpi_p f(x_{b_p}, y_{b_p}). \tag{3.51}$$

Some weights ϖ_p and corresponding barycentric coordinates can be seen in Table 3.2 where \mathcal{P} means the maximal degree of the polynomial which can be integrated exactly by the rule (3.51). x_{b_p}, y_{b_p} means the x and y coordinates created by p^{th} barycentric coordinates.

3.3.2 Limiting Process

Within this work 2D solution is limited in the similar way as 1D solution in Section 3.2.2.3. The cell is defined as troubled if the value in the middle of the finite element, defined by the barycentric coordinates $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, is not bounded by the values in the vertices i.e.

$$\begin{aligned} \max(W_{i,k}(1, 0, 0), W_{i,k}(0, 1, 0), W_{i,k}(0, 0, 1)) &< W_{i,k}(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}) \\ \text{or} \\ \min(W_{i,k}(1, 0, 0), W_{i,k}(0, 1, 0), W_{i,k}(0, 0, 1)) &> W_{i,k}(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}). \end{aligned} \quad (3.52)$$

In troubled cells the order of approximating polynomials is reduced to the second order and coefficients of higher degree polynomials are set to zero. This limiting process needs a special treatment to keep the integral averages of the conservative variables in the finite elements. In this work, three of possibilities are discussed.

First possibility is to add a constant $C_{i,l}$ to the basis functions (3.44), so that the integral average of a basis function, with higher polynomial degree than one, is zero. New basis functions $\varphi_l^*(x, y)$ have the form

$$\begin{aligned} \varphi_l^*(x, y) &= \varphi_l(x, y), \quad \text{for } l = 1, \\ \varphi_l^*(x, y) &= \varphi_l(x, y) - C_{i,l}, \quad \forall l > 1, \end{aligned} \quad (3.53)$$

where

$$C_{i,l} = \frac{\int_{\Omega_i} \varphi_l(x, y) d\Omega_i}{\int_{\Omega_i} 1 d\Omega_i}. \quad (3.54)$$

Now the integral of the basis function with $l > 1$ over the finite element is zero and limiting process is conservative, even if higher polynomials of the solution are set to zero.

Second possibility is orthogonalising process for the basis function system in each finite element. The functions $\varphi_i(x, y)$ must be orthogonal i.e.

$$\int_{\Omega_i} \varphi_i(x, y) \varphi_j(x, y) d\Omega_i \begin{cases} \neq 0, & i = j, \\ = 0, & i \neq j. \end{cases} \quad (3.55)$$

Orthogonalising process is not discussed in this work, but the orthogonalisation can be done for example by the Gram-Schmitt orthogonalising method.

The last possibility mentioned in this work is mapping of the basis functions in to the system of the orthogonal functions which are defined at the reference triangle. The advantage of this method is that there is no need to do the orthogonalising process in each triangle. In this work the orthogonal functions are defined in the $\xi - \eta$ space on reference triangle shown in Figure 3.8. This polynomials are defined as [86]

$$\varphi_{i,j}(\xi, \eta) = p_i^0 \left(\frac{2\xi}{1-\eta} \right) p_j^{2i+1}(\eta) \left(\frac{1-\eta}{2} \right)^i \quad (3.56)$$

where

$$p_n^\alpha(a) = \frac{1}{2^n n! (1-a)^\alpha} \frac{d^n}{da^n} ((1-a)^\alpha (x^2-1)^n) \quad (3.57)$$

is n^{th} Jacobi polynomial. In (3.58) there are first ten basis functions written in 'basis functions triangle' and in Figure 3.9 the basis are visualised.

$$\begin{array}{cccc}
 & & & 1 \\
 & & & \xi \\
 & & \frac{1}{2}(1+3\eta) & \\
 & \frac{1}{8}(12\xi^2 - (\eta-1)^2) & \frac{1}{2}\xi(3+5\eta) & \frac{5}{2}\eta^2 + \eta - \frac{1}{2} \\
 & \frac{1}{8}\xi(20\xi^2 - 3(\eta-1)^2) & -\frac{1}{16}(-12\xi^2 + (\eta-1)^2)(5+7\eta) & \frac{1}{4}\xi(1+18\eta+21\eta^2) & \frac{1}{8}(-3-15\eta+15\eta^2+35\eta^3) \\
 & & & & (3.58)
 \end{array}$$

The division of the orthogonal basis functions φ_l by its norm $||\Omega(\xi, \eta)|| = \int_{\Omega} \varphi_i d\Omega$ yields orthonormal system. The norms of the basis (3.58) are written in (3.59).

$$\begin{array}{cccc}
 & & & 1 \\
 & & & \frac{1}{6} & \frac{1}{2} \\
 & & & \frac{1}{15} & \frac{1}{9} & \frac{1}{3} \\
 & & & \frac{1}{4} & \frac{1}{12} & \frac{1}{20} & \frac{1}{28} \\
 & & & & & & (3.59)
 \end{array}$$

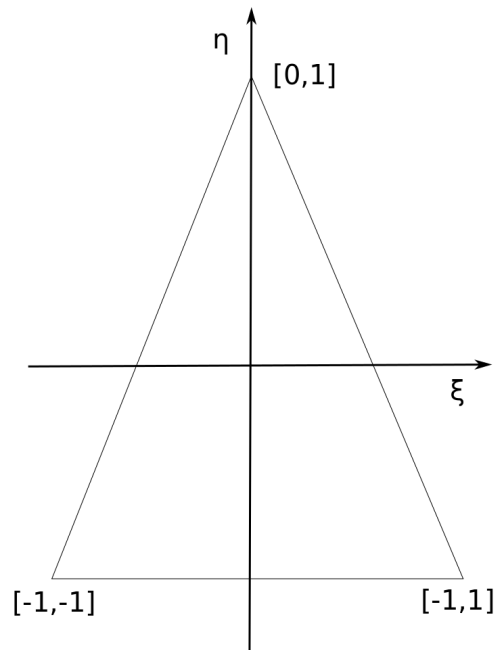


Figure 3.8: Reference triangle.

The transformation between the axes $\mathbf{x} = [x, y]$ and $\boldsymbol{\xi} = [\xi, \eta]$ can be expressed in the matrix form

$$\mathbf{x} - \mathbf{x}_1 = \mathbf{T}_i(\boldsymbol{\xi} - \boldsymbol{\xi}_1) \quad (3.60)$$

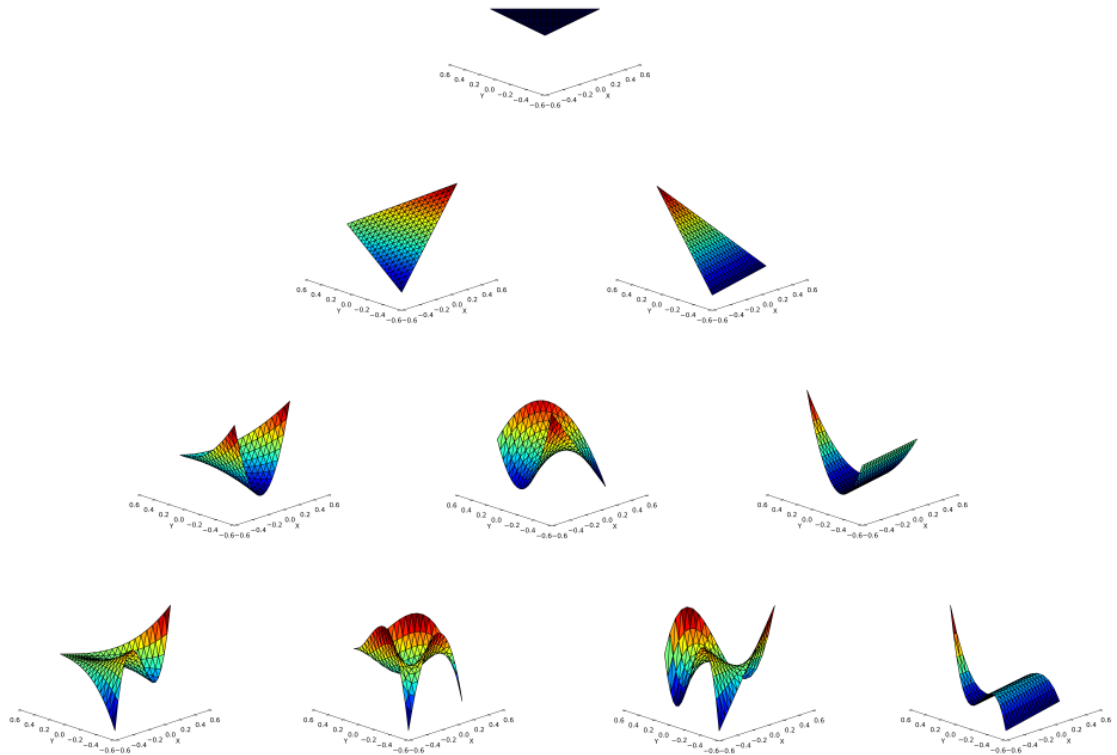


Figure 3.9: The system of orthogonal basis functions $\varphi_k(x, y)$.

or

$$\mathbf{x} = \mathbf{x}_1 + \mathbf{T}_i(\boldsymbol{\xi} - \boldsymbol{\xi}_1) \quad (3.61)$$

where $\mathbf{x}_1 = [x_1, y_1]$ and $\boldsymbol{\xi}_1 = [-1, -1]$ are the coordinates of the first vertices in the triangles. Jacobian matrix \mathbf{T} of rank 2×2 can be found by solution of the following system of linear equations

$$\begin{aligned} \mathbf{x}_2 - \mathbf{x}_1 &= \mathbf{T}_i(\boldsymbol{\xi}_2 - \boldsymbol{\xi}_1) \\ \mathbf{x}_3 - \mathbf{x}_1 &= \mathbf{T}_i(\boldsymbol{\xi}_3 - \boldsymbol{\xi}_1) \end{aligned} \quad (3.62)$$

here $\boldsymbol{\xi}_i, \mathbf{x}_i$, $i = 2, 3$ are also the vertices of the corresponding triangle Ω_i .

Derivatives of the Basis Functions and Volume Integral

Equation 3.5 contains the volume integral and also the derivatives of the basis functions. The computations of the volume integral and the derivatives for the orthogonal basis functions defined on the reference triangle is described in the following section.

The derivatives of the basis functions are

$$\begin{aligned} \frac{\partial \varphi}{\partial x} &= \frac{\partial \varphi}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial \varphi}{\partial \eta} \frac{\partial \eta}{\partial x} \\ \frac{\partial \varphi}{\partial y} &= \frac{\partial \varphi}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial \varphi}{\partial \eta} \frac{\partial \eta}{\partial y} \end{aligned} \quad (3.63)$$

Where the derivatives $\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta}$ can be derived from $\boldsymbol{\xi} - \boldsymbol{\xi}_1 = \mathbf{T}_i^{-1}(\mathbf{x} - \mathbf{x}_1)$ i.e.

$$\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \eta_1 \end{bmatrix} + \begin{bmatrix} T_{11}^{-1} & T_{12}^{-1} \\ T_{21}^{-1} & T_{22}^{-1} \end{bmatrix} \begin{bmatrix} x - x_1 \\ y - y_1 \end{bmatrix}. \quad (3.64)$$

Equation 3.64 yields

$$\begin{aligned} \frac{\partial \xi}{\partial x} &= T_{11}^{-1}, \\ \frac{\partial \eta}{\partial x} &= T_{21}^{-1}, \end{aligned} \quad (3.65)$$

$$\begin{aligned} \frac{\partial \xi}{\partial y} &= T_{12}^{-1}, \\ \frac{\partial \eta}{\partial y} &= T_{22}^{-1}, \end{aligned}$$

and the substitution of (3.65) into (3.63) results in

$$\frac{\partial \varphi}{\partial x} = \frac{\partial \varphi}{\partial \xi} T_{11}^{-1} + \frac{\partial \varphi}{\partial \eta} T_{21}^{-1}, \quad (3.66)$$

$$\frac{\partial \varphi}{\partial y} = \frac{\partial \varphi}{\partial \xi} T_{12}^{-1} + \frac{\partial \varphi}{\partial \eta} T_{22}^{-1}.$$

The volume integral over the reference triangle is transformed into

$$\int_{\Omega_i} f(x, y) d\Omega_i = \int_{\Omega_\Delta} f(x, y) J(\mathbf{T}_i) d\Omega_\Delta \approx \sum_{p=1}^{n_p} \varpi_p f(x_{b_p}, y_{b_p}) |\Omega_i|, \quad (3.67)$$

where $J(\mathbf{T}_i)$ is the Jacobian of the transformation $\mathbf{x} \rightarrow \boldsymbol{\eta}$. This Jacobian is equal to the area of the finite element $|\Omega_i|$.

3.3.3 Bed Slope Source Term

The bed slope source from (3.46) is easily computed by the Gaussian quadrature (3.51) as

$$\int_{\Omega_i} S_{B,k}(x, y) \varphi_i^l(x, y) d\Omega = \sum_{p=1}^{n_p} \varpi_p S_{B,k}(x_{b_p}, y_{b_p}) \varphi_i^l(x_{b_p}, y_{b_p}). \quad (3.68)$$

This work describes only bed function approximated by the linear functions within 2D solver. In the following, there is described the setting of the bed function coefficients for Taylor and orthogonal basis systems. The bed function is approximated by the linear combination (3.43) such that

$$B_i(x, y) = \sum_{l=1}^{n_B} w_{B,i}^l \varphi_l(x, y) \quad (3.69)$$

$n_B = 3$ because only linear functions are considered. Let the bed altitude (Z) to be defined in the vertexes of the finite element Ω_i (points 1,2,3 in Figure 3.10). Two arbitrary vectors (\vec{u}, \vec{v}) creating the element edges can be computed as

$$\begin{aligned} u_x &= X_2 - X_1 \\ u_y &= Y_2 - Y_1 \\ u_z &= Z_2 - Z_1 \end{aligned} \quad (3.70)$$

$$\begin{aligned} v_x &= X_3 - X_1 \\ v_y &= Y_3 - Y_1 \\ v_z &= Z_3 - Z_1 \end{aligned}$$

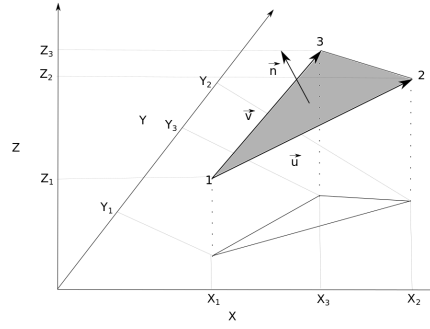


Figure 3.10: Approximated bed function in a finite element.

The components of the normal vector \vec{n} are

$$\begin{aligned} n_x &= u_y v_z - u_z v_y, \\ n_y &= u_z v_x - u_x v_z, \\ n_z &= u_x v_y - u_y v_x. \end{aligned} \quad (3.71)$$

If the bed function is approximated in aid of the Taylor basis (3.45) then the coefficients of the linear approximation (3.69) are

$$\begin{aligned} w_{B,i}^1 &= \frac{Z_1 + Z_2 + Z_3}{3}, \\ w_{B,i}^2 &= -\frac{n_x}{n_z}, \\ w_{B,i}^3 &= -\frac{n_y}{n_z}. \end{aligned} \quad (3.72)$$

To set 'lake in rest' initial condition of the water depth with initial water level h_0 , the coefficients of the water depth $W_{i,1}(x, y, t) = \sum_{l=1}^{n_b} w_{i,1}^l(t) \varphi_i^l(x, y)$ are

$$\begin{aligned} w_{i,1}^1 &= h_0 - w_{B,i}^1, \\ w_{i,1}^2 &= -w_{B,i}^2, \\ w_{i,1}^3 &= -w_{B,i}^3, \\ w_{i,1}^l &= 0, \quad \text{for } l = 4, 5, \dots, n_b. \end{aligned} \quad (3.73)$$

In case of orthogonal basis functions (3.58) the bed function coefficients $w_{B,i}^l$ have to be set in different way. Within this work, it is suggested to set these coefficients in the following way. The multiplying of the bed approximation by the basis function φ_j and integrating over the finite element yields

$$\int_{\Omega_i} B_i \varphi_l d\Omega_i = \sum_{l=1}^{n_B} w_{B,i}^l \int_{\Omega_i} \varphi_l \varphi_j d\Omega_i \quad (3.74)$$

$B_i(x, y)$ was replaced by B_i for the sake of simplicity. The orthogonality of the basis functions yields

$$\int_{\Omega_i} B_i \varphi_l d\Omega_i = \begin{cases} 0, & l \neq j \\ w_{B,i}^l \int_{\Omega_i} \varphi_l \varphi_j d\Omega_i, & l = j. \end{cases} \quad (3.75)$$

and (3.75) results in

$$w_{B,i}^l = \frac{\int_{\Omega_i} B_i \varphi_l d\Omega_i}{\int_{\Omega_i} \varphi_l \varphi_l d\Omega_i}. \quad (3.76)$$

The initial condition 'lake in rest' of the water depth can be set in the same way as in (3.73).

3.3.4 Bed Friction Source Term

The bed friction in 2D is similar to 1D case. The bed friction source term (1.6) contains two non-zero components

$$\mathbf{S}_f = \begin{bmatrix} 0 \\ -\tau_f^x \\ -\tau_f^y \end{bmatrix}.$$

I.e. the friction term is non-zero for $k = 2, 3$ and numerical approximation is computed in aid of Gaussian quadrature for volume integral

$$\begin{aligned} \int_{\Omega_i} -\tau_f^x \varphi_i^j d\Omega &= - \int_{\Omega_i} C_f u |u| \varphi_i^j d\Omega, \quad j = 1, 2, \dots, n_b; \quad k = 2. \\ \int_{\Omega_i} -\tau_f^y \varphi_i^j d\Omega &= - \int_{\Omega_i} C_f v |v| \varphi_i^j d\Omega, \quad j = 1, 2, \dots, n_b; \quad k = 3. \end{aligned} \quad (3.77)$$

As mentioned before, the bed shear stress coefficient C_f is the same for the X and Y direction. The Equation 3.77 cannot be directly added to the Equation 3.46. To implement this source term, the same limiter as in 3.42 has to be used.

$$\mathbf{w}_{i,k}^{n+1} = \begin{cases} \mathbf{w}_{i,k}^* & \text{if } \text{sign}\left(W_{i,k}\left(\mathbf{w}_{i,k}^n\right)\right) = \text{sign}\left(W_{i,k}\left(\mathbf{w}_{i,k}^*\right)\right), \\ 0 & \text{otherwise,} \end{cases} \quad \text{for } k = 2, 3 \quad (3.78)$$

where $\mathbf{w}_{i,k}^* = \mathbf{w}_{i,k}^n + \Delta t \mathbf{M}_i^{-1} \mathbf{RHS}_{i,k}^f$. The Equation 3.78 also means that the discharge $W_{i,2} = (hu)_i$, $W_{i,3} = (hv)_i$ respectively, is set to zero in the case when the bed friction term changes the flow direction.

Table 3.1: Weights and points of the Gaussian quadrature

n_p	Weights ϖ_p	Integration points x_p
1	2.0000000000000000	0.0000000000000000
2	1.0000000000000000	-0.5773502691896257
	1.0000000000000000	0.5773502691896257
3	0.8888888888888888	0.0000000000000000
	0.5555555555555556	-0.7745966692414834
	0.5555555555555556	0.7745966692414834
4	0.6521451548625461	-0.3399810435848563
	0.6521451548625461	0.3399810435848563
	0.3478548451374538	-0.8611363115940526
	0.3478548451374538	0.8611363115940526
5	0.5688888888888889	0.0000000000000000
	0.4786286704993665	-0.5384693101056831
	0.4786286704993665	0.5384693101056831
	0.2369268850561891	-0.9061798459386640
	0.2369268850561891	0.9061798459386640
6	0.3607615730481386	0.6612093864662645
	0.3607615730481386	-0.6612093864662645
	0.4679139345726910	-0.2386191860831969
	0.4679139345726910	0.2386191860831969
	0.1713244923791704	-0.9324695142031521
	0.1713244923791704	0.9324695142031521
7	0.4179591836734694	0.0000000000000000
	0.3818300505051189	0.4058451513773972
	0.3818300505051189	-0.4058451513773972
	0.2797053914892766	-0.7415311855993945
	0.2797053914892766	0.7415311855993945
	0.1294849661688697	-0.9491079123427585
	0.1294849661688697	0.9491079123427585
8	0.3626837833783620	-0.1834346424956498
	0.3626837833783620	0.1834346424956498
	0.3137066458778873	-0.5255324099163290
	0.3137066458778873	0.5255324099163290
	0.2223810344533745	-0.7966664774136267
	0.2223810344533745	0.7966664774136267
	0.1012285362903763	-0.9602898564975363
	0.1012285362903763	0.9602898564975363
9	0.3302393550012598	0.0000000000000000
	0.1806481606948574	-0.8360311073266358
	0.1806481606948574	0.8360311073266358
	0.0812743883615744	-0.9681602395076261
	0.0812743883615744	0.9681602395076261
	0.3123470770400029	-0.3242534234038089
	0.3123470770400029	0.3242534234038089
	0.2606106964029354	-0.6133714327005904
	0.2606106964029354	0.6133714327005904

Table 3.2: Weights and barycentric coordinates of the Gaussian quadrature

\mathcal{P}	Weights ϖ_p	Barycentric coordinates (b_1, b_2, b_3)
1	1.0000000000000000	(0.3333333333333333, 0.3333333333333333, 0.3333333333333333)
2	0.3333333333333334	(0.1666666666666666, 0.6666666666666667, 0.1666666666666667)
	0.3333333333333334	(0.6666666666666667, 0.1666666666666667, 0.1666666666666666)
	0.3333333333333334	(0.1666666666666667, 0.1666666666666666, 0.6666666666666667)
3	-0.5625000000000000	(0.3333333333333333, 0.3333333333333333, 0.3333333333333334)
	0.5208333333333334	(0.2000000000000000, 0.6000000000000000, 0.2000000000000000)
	0.5208333333333334	(0.6000000000000000, 0.2000000000000000, 0.2000000000000000)
	0.5208333333333334	(0.2000000000000000, 0.2000000000000000, 0.6000000000000000)
4	0.223381589678012	(0.445948490915965, 0.108103018168070, 0.445948490915965)
	0.109951743655322	(0.091576213509770, 0.816847572980459, 0.091576213509771)
	0.223381589678012	(0.108103018168070, 0.445948490915965, 0.445948490915965)
	0.109951743655322	(0.816847572980459, 0.091576213509771, 0.091576213509770)
	0.223381589678012	(0.445948490915965, 0.445948490915965, 0.108103018168070)
	0.109951743655322	(0.091576213509771, 0.091576213509770, 0.816847572980459)
5	0.2250000000000000	(0.3333333333333333, 0.3333333333333333, 0.3333333333333334)
	0.132394152788506	(0.470142064105115, 0.059715871789770, 0.470142064105115)
	0.125939180544827	(0.101286507323457, 0.797426985353087, 0.101286507323456)
	0.132394152788506	(0.059715871789770, 0.470142064105115, 0.470142064105115)
	0.125939180544827	(0.797426985353087, 0.101286507323456, 0.101286507323457)
	0.132394152788506	(0.470142064105115, 0.470142064105115, 0.059715871789770)
	0.125939180544827	(0.101286507323456, 0.101286507323457, 0.797426985353087)
6	0.116786275726379	(0.249286745170911, 0.501426509658179, 0.249286745170910)
	0.050844906370207	(0.063089014491502, 0.873821971016996, 0.063089014491502)
	0.116786275726379	(0.501426509658179, 0.249286745170910, 0.249286745170911)
	0.050844906370207	(0.873821971016996, 0.063089014491502, 0.063089014491502)
	0.116786275726379	(0.249286745170910, 0.249286745170911, 0.501426509658179)
	0.050844906370207	(0.063089014491502, 0.063089014491502, 0.873821971016996)
	0.082851075618374	(0.636502499121399, 0.053145049844817, 0.310352451033784)
	0.082851075618374	(0.310352451033784, 0.053145049844817, 0.636502499121399)
	0.082851075618374	(0.053145049844817, 0.310352451033784, 0.636502499121399)
	0.082851075618374	(0.053145049844817, 0.636502499121399, 0.310352451033784)
	0.082851075618374	(0.310352451033784, 0.636502499121399, 0.053145049844817)
	0.082851075618374	(0.636502499121399, 0.310352451033784, 0.053145049844817)
	7	-0.149570044467682
0.175615257433208		(0.260345966079040, 0.479308067841920, 0.260345966079040)
0.053347235608838		(0.065130102902216, 0.869739794195568, 0.065130102902216)
0.175615257433208		(0.479308067841920, 0.260345966079040, 0.260345966079040)
0.053347235608838		(0.869739794195568, 0.065130102902216, 0.065130102902216)
0.175615257433208		(0.260345966079040, 0.260345966079040, 0.479308067841920)
0.053347235608838		(0.065130102902216, 0.065130102902216, 0.869739794195568)
0.077113760890257		(0.638444188569810, 0.048690315425316, 0.312865496004874)
0.077113760890257		(0.312865496004874, 0.048690315425316, 0.638444188569810)
0.077113760890257		(0.048690315425316, 0.312865496004874, 0.638444188569810)
0.077113760890257		(0.048690315425316, 0.638444188569810, 0.312865496004874)
0.077113760890257		(0.312865496004874, 0.638444188569810, 0.048690315425316)
0.077113760890257		(0.638444188569810, 0.312865496004874, 0.048690315425316)

Chapter 4

Time Integration

In this chapter, the principle of the time integration of the system of partial differential equations (2.11) is described. The time integration is solved by the discrete methods, i.e. methods which approximate the solution in the finite set of the points. Equation 2.11 can be also written as

$$\frac{d\mathbf{W}_i^n}{dt} = \mathbf{RHS}_i(\mathbf{W}^n), \quad (4.1)$$

where n means the time level.

The Euler's method, second and fourth order Runge-Kutta methods were chosen and are described within this work. These methods, unlike the methods of Tylor's type, do not have the solution conditioned by differentiation of the right hand side. Only $\mathbf{RHS}_i(\mathbf{W}^n)$ has to be enumerated in order to obtain the solution in the next time step \mathbf{W}^{n+1} . The initial conditions are $\mathbf{W}(0) = \mathbf{W}^0$.

4.1 Euler's Method

This method is only 1st order accurate, but its advantage is small computational demands. It suffices to enumerate the value of $\mathbf{RHS}_i(\mathbf{W}^n)$ only once in each time step.

4.1.1 The Principle of Euler's Method

The principle of the Euler method can be explained at some general first order differential equation

$$y'(x) = f(x, y(x)). \quad (4.2)$$

The boundary conditions are

$$y(x_0) = y_0, \quad y, x \in R. \quad (4.3)$$

The solution of the equation is to be solved in the computational domain $I = \langle x_0, x_0 + a \rangle$ where a is the length of the interval I . This interval will be divided by dividing points into disjunct intervals of length h_i . Dividing points are

$$\begin{aligned} x_1 &= x_0 + h_1, \\ x_2 &= x_0 + h_1 + h_2 = x_1 + h_2 \\ x_3 &= x_2 + h_3 \\ &\vdots \end{aligned} \quad (4.4)$$

Once the derivative $y'_0 = f(x_0, y_0)$ is calculated, the approximate solution Y_n of this differential equation at the subintervals is solved as [94]

$$\begin{aligned} Y_1 &= y_0 + h_1 y'_0 = y_0 + h_1 f(x_0, y_0), \\ Y_2 &= Y_1 + h_2 f(x_1, y_1), \\ Y_3 &= \dots \end{aligned} \quad (4.5)$$

4.1.2 Application of the Euler's Method

Using relations (4.5), the solution of (4.1) in next time level is

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n + \Delta t \cdot \mathbf{RHS}_i(\mathbf{W}^n). \quad (4.6)$$

4.2 Second-Order Runge-Kutta Method

For higher precision of the time integration, second-order Runge-Kutta method can be used. This method is also known as Euler's modified method [94]. The error of the method is $\mathcal{O}(\Delta t^2)$. Necessary derivatives $f(x_n, y_n)$ are computed by improved differential method with the aid of another auxiliary points between x_i and x_{i+1} . In each time step the value of $\mathbf{RHS}_i(\mathbf{W})$ has to be enumerated twice.

4.2.1 Principle of the Second-Order Runge-Kutta Method

The equation (4.2) is integrated over one subinterval

$$\int_{x_k}^{x_{k+1}} y'(x) dx = \int_{x_k}^{x_{k+1}} f(x, y(x)) dx. \quad (4.7)$$

The integral on the left hand side is enumerated

$$y(x_{k+1}) - y(x_k) = \int_{x_k}^{x_{k+1}} f(x, y(x)) dx, \quad (4.8)$$

and on the right hand side is used the trapezoidal rule

$$y(x_{k+1}) - y(x_k) = \frac{h}{2} [f(x_k, y(x_k)) + f(x_{k+1}, y^*(x_{k+1}))] + \mathcal{O}(h^3), \quad (4.9)$$

where $\mathcal{O}(h^3)$ is the error of the trapezoidal rule and $y^*(x_{k+1})$ is approximated by aforementioned Euler's method

$$y^*(x_{k+1}) = y(x_k) + h \cdot f(x_k, y(x_k)) + \mathcal{O}(h^2). \quad (4.10)$$

The final form of the approximate solution Y_{k+1} is

$$\begin{aligned} Y_{k+1} &= Y_k + \frac{h}{2} [f(x_k, y_k(x_k)) + f(x_{k+1}, y_{k+1}^*(x_k))], \\ Y_{k+1}^* &= Y_k + h \cdot f(x_k, y(x_k)). \end{aligned} \quad (4.11)$$

4.2.2 Application of the Second-Order Runge-Kutta Method

The solution \mathbf{W}_i^{n+1} at the $n + 1^{th}$ time level is computed as

$$\begin{aligned} \mathbf{W}_i^{n+1} &= \mathbf{W}_i^n + \frac{\Delta t}{2} (\mathbf{RHS}_i(\mathbf{W}^n) + \mathbf{RHS}_i(\mathbf{W}^{n+1*})), \\ \mathbf{W}_i^{n+1*} &= \mathbf{W}_i^n + \Delta t \cdot \mathbf{RHS}_i(\mathbf{W}^n). \end{aligned} \quad (4.12)$$

4.3 Fourth-Order Runge-Kutta Method

Higher order schemes of time integration are needed when the numerical flux is approximated by higher order schemes. One of the higher order schemes is fourth-order Runge-Kutta method (RK4). The RK4 method has the fourth-order of accuracy. It means that the error per step is of the order $\mathcal{O}(\Delta t^5)$ while the total accumulated error has order $\mathcal{O}(\Delta t^4)$ [95]. The higher accuracy of RK4 is redeemed by high computational demands. Computational demands increase twice in comparison with the second-order Runge-Kutta method and four-times in comparison with Euler's method. Approximate solution of the (4.2) is

$$\begin{aligned}
 Y_{n+1} &= Y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\
 k_1 &= f(x_n, y_n), \\
 k_2 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1), \\
 k_3 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2), \\
 k_4 &= f(x_n + h, y_n + h \cdot k_3).
 \end{aligned} \tag{4.13}$$

4.3.1 Application of the Fourth-Order Runge-Kutta Method

The solution \mathbf{W}_i^{n+1} of (4.1) at the $n + 1^{th}$ time level is accordingly to the (4.13)

$$\begin{aligned}
 \mathbf{W}_i^{n+1} &= \mathbf{W}_i^n + \frac{\Delta t}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4), \\
 \mathbf{k}_1 &= \text{RHS}_i(\mathbf{W}^n), \\
 \mathbf{k}_2 &= \text{RHS}_i(\mathbf{W}^{n+1*}) \\
 \mathbf{W}_i^{n+1*} &= \mathbf{W}_i^n + \frac{\Delta t}{2}\mathbf{k}_1, \\
 \mathbf{k}_3 &= \text{RHS}_i(\mathbf{W}^{n+1**}) \\
 \mathbf{W}_i^{n+1**} &= \mathbf{W}_i^n + \frac{\Delta t}{2}\mathbf{k}_2, \\
 \mathbf{k}_4 &= \text{RHS}_i(\mathbf{W}^{n+1***}) \\
 \mathbf{W}_i^{n+1***} &= \mathbf{W}_i^n + \Delta t\mathbf{k}_3.
 \end{aligned} \tag{4.14}$$

4.4 Necessary Condition of Convergence

Special condition for the time-step size is needed for explicit temporal quadratures in contrast to implicit ones. More detailed analysis of the stability of both implicit and explicit quadratures can be found in [96].

In this work the Courant-Friedrichs-Lewy condition (CFL condition) for aforementioned explicit temporal quadratures is described. The condition is named after Richard Courant, Kurt Friedrichs, and Hans Lewy who described it in [97].

Definition 6. (Convergent numerical method) Numerical method is called convergent when the numerical solution converges to the exact solution as the grid is refined (as $\Delta x \rightarrow 0$, $\Delta t \rightarrow 0$) [98].

CFL condition says that the method can be convergent only if its numerical domain of dependence contains the true domain of dependence of the partial differential equation, at least in the limit as Δt

and Δx go to zero [98]. Principle of CFL condition is based on evaluation of maximal possible time-step Δt_i for each finite volume/element and choosing the smallest one. In case of the mesh created by the quadrangles the local time-step can be computed as

$$\Delta t_i = \frac{\text{CFL}}{\frac{\lambda_{A_i}}{\Delta x} + \frac{\lambda_{B_i}}{\Delta y}}, \quad (4.15)$$

where

$$\lambda_{A_i} = |u_i| + \sqrt{gh_i}, \quad \lambda_{B_i} = |v_i| + \sqrt{gh_i}, \quad (4.16)$$

CFL is chosen constant and Δx , Δy are lengths of quadrangle in X and Y directions.

In case of the mesh created by triangles should be Δt_i computed as

$$\Delta t_i = \frac{\text{CFL}}{\frac{\lambda_{A_i} + \lambda_{B_i}}{\Delta_i}}, \quad (4.17)$$

here Δ_i denotes the root of the volume of finite volume Ω_i . CFL constant can be larger for the methods of higher time integrating accuracy.

In case of the DGFEM, the CFL condition has to be stricter. It is recommended to compute the maximal possible time step as

$$\Delta t_i = \frac{\text{CFL}}{(2\mathcal{P} + 1) \frac{\lambda_{A_i} + \lambda_{B_i}}{\Delta_i}}, \quad (4.18)$$

where \mathcal{P} means the highest polynomial degree of the basis functions.

Chapter 5

Unstructured Meshes

Albeit unstructured meshes are clumsy to use, the benefit of its flexibility in meshing the irregular complex domains, makes unstructured meshes very useful. In the following only triangular meshes are described and the approach from [99] is followed.

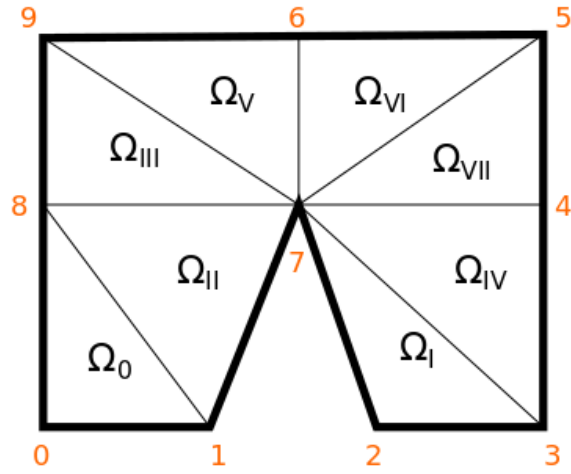
5.1 Input data

Input data are matrices \mathbf{P} and \mathbf{TP} which are defined by some meshing program. In the matrix \mathbf{P} , the coordinates of particular points are stored. The numbers of points creating particular finite volume are stored in the matrix \mathbf{TP} . The size of \mathbf{TP} , \mathbf{P} respectively, is $[n_t \times 3]$, $[n_p \times 2]$ respectively, where n_t is the number of the finite volumes and n_p is the number of the points. Points in the finite volumes are numbered contra-clockwise. For the mesh shown in Figure 5.1 the matrix \mathbf{P} looks like

$$\mathbf{P} = \begin{bmatrix} 0.0 & 0.0 \\ 1.0 & 0.0 \\ 2.0 & 0.0 \\ 3.0 & 0.0 \\ 3.0 & 1.0 \\ 3.0 & 2.0 \\ 1.5 & 2.0 \\ 1.5 & 1.0 \\ 0.0 & 1.0 \\ 0.0 & 2.0 \end{bmatrix} \quad (5.1)$$

and one of the possible matrices \mathbf{TP} is

$$\mathbf{TP} = \begin{bmatrix} 0 & 1 & 8 \\ 7 & 2 & 3 \\ 8 & 1 & 7 \\ 8 & 7 & 9 \\ 7 & 3 & 4 \\ 6 & 9 & 7 \\ 7 & 5 & 6 \\ 5 & 7 & 4 \end{bmatrix}. \quad (5.2)$$

Figure 5.1: Division of the computational domain Ω into triangles

5.2 EP Matrix

In matrix \mathbf{EP} the numbers of points of particular edge are stored. The size of matrix \mathbf{EP} is $[n_e \times 2]$ where n_e is the total number of the edges. This matrix is assembled using auxiliary matrices $\mathbf{TP}^{1,2}$, $\mathbf{TP}^{2,3}$ and $\mathbf{TP}^{3,1}$. These matrices are made of columns (1st-2nd, 2nd-3rd, 3rd-1st) of the matrix \mathbf{TP} :

$$\mathbf{TP}^{1,2} = \begin{bmatrix} 0 & 1 \\ 7 & 2 \\ 8 & 1 \\ 8 & 7 \\ 7 & 3 \\ 6 & 9 \\ 7 & 5 \\ 5 & 7 \end{bmatrix}, \mathbf{TP}^{2,3} = \begin{bmatrix} 1 & 8 \\ 2 & 3 \\ 1 & 7 \\ 7 & 9 \\ 3 & 4 \\ 9 & 7 \\ 5 & 6 \\ 7 & 4 \end{bmatrix}, \mathbf{TP}^{3,1} = \begin{bmatrix} 8 & 0 \\ 3 & 7 \\ 7 & 8 \\ 9 & 8 \\ 4 & 7 \\ 7 & 6 \\ 6 & 7 \\ 4 & 5 \end{bmatrix}. \quad (5.3)$$

Each row of this matrices represents an edge. But some duplicities like [7-8, 8-7] or [8-1, 1-8] occur and these have to be removed. After that the final matrix **EP** should look like

$$\mathbf{EP} = \begin{array}{l} \left[\begin{array}{l} 0 \ 1 \\ 1 \ 8 \\ 8 \ 0 \\ 7 \ 2 \\ 2 \ 3 \\ 3 \ 7 \\ 1 \ 7 \\ 7 \ 8 \\ 7 \ 9 \\ 9 \ 8 \\ 6 \ 7 \\ 7 \ 5 \\ 5 \ 6 \\ 6 \ 9 \\ 7 \ 4 \\ 4 \ 5 \\ 3 \ 4 \end{array} \right] \begin{array}{l} \textit{edge number 0} \\ \textit{edge number 1} \\ \textit{edge number 2} \\ \textit{edge number 3} \\ \textit{edge number 4} \\ \textit{edge number 5} \\ \textit{edge number 6} \\ \textit{edge number 7} \\ \textit{edge number 8} \\ \textit{edge number 9} \\ \textit{edge number 10} \\ \textit{edge number 11} \\ \textit{edge number 12} \\ \textit{edge number 13} \\ \textit{edge number 14} \\ \textit{edge number 15} \\ \textit{edge number 16} \end{array} \end{array} . \quad (5.4)$$

Total number of the edges n_e is equal to the size of the matrix **EP**.

5.3 **TE, ET and ETyp** Matrices

In matrix **TE** the numbers of edges creating given finite volume are stored, i.e.

$$\mathbf{TE} = \begin{array}{l} \left[\begin{array}{l} 0 \ 1 \ 8 \\ 7 \ 2 \ 3 \\ 8 \ 1 \ 7 \\ 8 \ 7 \ 9 \\ 7 \ 3 \ 4 \\ 6 \ 9 \ 7 \\ 7 \ 5 \ 6 \\ 5 \ 7 \ 4 \end{array} \right] \end{array} \quad (5.5)$$

In the matrix **ET** the numbers of triangles which are adjacent to an edge are stored. The size of this matrix is $[n_e \times 2]$. Simultaneously the type of the edge can be find and saved in the matrix **Etyp** $[n_e \times 1]$. Inner edges are numbered by 0 and the edges generating the boundary are numbered

by 1. The final matrices **ET** and **ETyp** should be

$$\mathbf{ET} = \begin{bmatrix} 0 & -1 \\ 1 & 3 \\ 0 & -1 \\ 1 & -1 \\ 1 & -1 \\ 2 & 8 \\ 2 & -1 \\ 3 & 4 \\ 4 & 6 \\ 3 & -1 \\ 5 & 6 \\ 5 & 7 \\ 4 & -1 \\ 5 & -1 \\ 7 & 8 \\ 6 & -1 \\ 7 & -1 \end{bmatrix}, \mathbf{ETyp} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}. \quad (5.6)$$

Number -1 means that the edge creates only one finite volume, thus it is a wall edge.

5.4 **Enx, Eny, Edl** and **Tzn** Matrices

Matrix **Edl** is the matrix where the lengths of the edges are stored. The x (y respectively) component of the normal vector is stored in matrix **Enx** (**Eny** respectively). In the matrix **Tzn** the information whether the normal vector is pointing inside (-1) or outside (1) of the finite volume is stored.

$$\mathbf{Tzn} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & -1 & 1 \end{bmatrix} \quad (5.7)$$

The whole computational mesh with numbers of points, edges finite volumes and normal vectors is shown in Figure 5.2.

5.5 Implementation of the Unstructured-Mesh Algorithm

A simple algorithm, computing the right hand side from (2.11), is shown in this section. Provided that the numerical fluxes **Phi** across the edges and the bed slope source terms **S_b** are already computed, the right hand side can be, in C language, implemented as

```
for(i=0;i<nt;i++){
```

```

for(k=0;k<3;k++){
    RHS[i+k*nt] =-1/To[i]*(Tzn[i+0*nt] * Phi[TE[i+0*nt]+ne*k] * Edl[TE[i+0*nt]] +
        Tzn[i+1*nt] * Phi[TE[i+1*nt]+ne*k] * Edl[TE[i+1*nt]] +
        Tzn[i+2*nt] * Phi[TE[i+2*nt]+ne*k] * Edl[TE[i+2*nt]] )+
        Sb[i+k*nt];
}
}

```

in the code ne means total number of the edges n_e and nt means number of the finite volumes n_t .

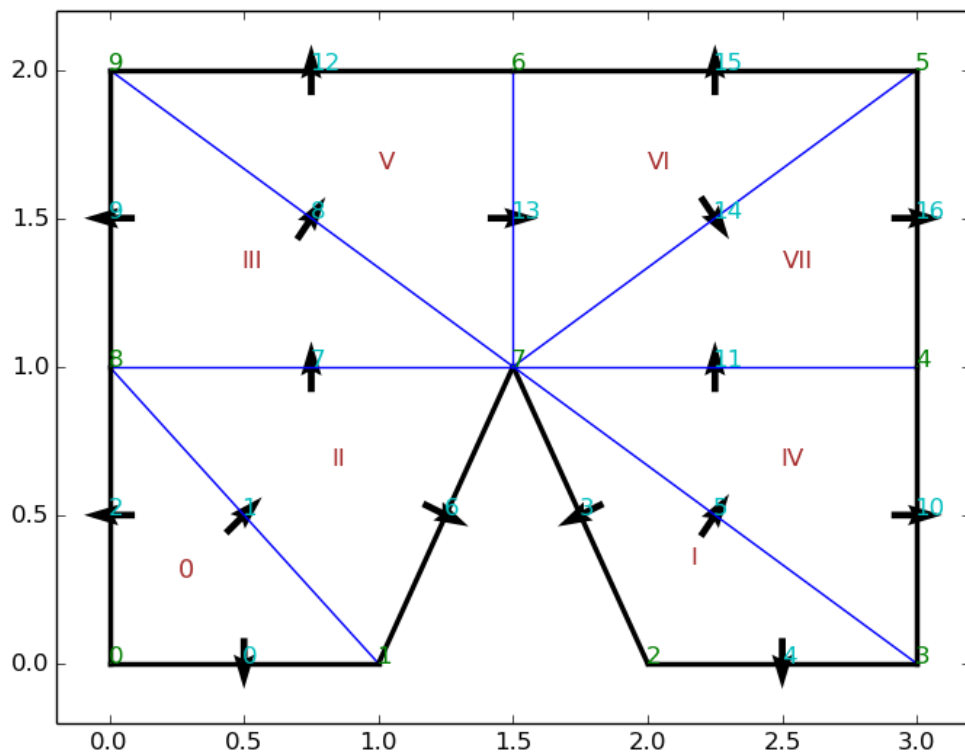


Figure 5.2: Computational mesh with normal vectors, numbers of edges, volumes and points. Inner edges are blue, wall edges are black.

Chapter 6

Parallelisation of the Code

The following chapter describes an introduction into the parallel computing, i.e. the converting sequential code into multi-threaded. The thread is a small sequence of program which can be executed. For example computing of the numerical flux across several edges or computing bed source term in several finite elements/volumes. There are described only basics of the parallel computing in this work and we refer the reader to [100–103] to know more about the parallel computing-especially the creation of the pools which is not described here.

The solver was written in C language and for the parallelisation **openMP** library was used. OpenMP (Open Multi-Processing) is an application programming interface (API) that supports multi-platform shared memory multiprocessing programming. The most time consuming operations in solvers are the *for* loops. This *for* loops can be easily divided in to the threads within openMP library. One thread contains specific amount of iterations. If in the code is imported omp library by the command

```
#include <omp.h>
```

then the loop can be easily parallelised by command

```
#pragma omp parallel for
```

written before the initialisation of the loop. This command automatically sets the number of threads which perform the *for* loop. The number of the threads can be also set manually by the command

```
#pragma omp parallel for num_threads(n_thread)
for(int i=0;i<n_interation;i++){
...
}
```

where `n_thread` means the number of the threads and `n_interation` number of iterations. The amount of performed work (iterations) is assigned to every thread automatically. We can assign how much of the work the thread performs by specifying the parallel regions in the code. This parallel region (...) is bounded by curly brackets and initialised by the command

```
#pragma omp parallel{
...
}
```

and command `num_threads(n_thread)` sets the number of the threads











```
#pragma omp parallel num_threads(n_thread){
    ...
}
```

The parallel regions can be run by different threads, but they share the variables. To make a copy of variables we can use command `private`











```
#pragma omp parallel num_threads(n_thread) private(vaiable1,variable2,variable3){
    ...
}
```

The last command, we need to use, is `omp_get_thread_num()`. This command returns the ID of currently running thread.

It is reasonable to divide the iterations among the threads uniformly, to do so we can proceed as follows. Let us consider a test case with 25 iterations and 10 threads. First way can be to divide the number of iterations by number of the threads, round it down and distribute as follows

Thread										
Iterations	0, 1	2, 3	4, 5	6, 7	8, 9	10, 11	12, 13	14, 15	16, 17	18, 19, 20 21, 22, 23 24
Number of iterations	2	2	2	2	2	2	2	2	2	7

It is obvious that the last thread number 9 is over loaded. The rounding up yields the following division











Thread										
Iterations	0, 1 2	3, 4 5	6, 7 8	9, 10 11	12, 13 14	15, 16 17	18,19 20	21, 22 23	24	-
Number of iterations	3	3	3	3	3	3	3	3	1	0

Now the thread number 9 is idle. Thus within this work we use the following division of the iterations. Each thread computes the iteration with the same ID plus n-times number of the threads if it does not exceeds the number of the iterations. The parallelisation can be done as

```
int ID;
#pragma omp parallel num_threads(n_thread) private(ID){

    ID = omp_get_thread_num();

    for(int i=ID;i < n_iteration; i=i+n_thread){
        ...
    }
}
```

Thread										
Iterations	0, 10 20	1, 11 21	2, 12 22	3, 13 23	4, 14 24	5, 15	6, 16	7, 17	8, 18	9, 19
Number of iterations	3	3	3	3	3	2	2	2	2	2

Now the iterations are relatively equally distributed among the threads but the number of threads shouldn't be larger than the number of the cores. The creation of excessive threads slows down the computations.

When computing with more threads, we need to avoid a writing to the same variable by more threads. To achieve this, we can use the lock of the variable or use the 'critical block' syntax. The critical block (...) is bounded by curly brackets and initialised by the command

```
#pragma omp critical{
    ...
}
```

The code inside the critical block can be performed only by one thread and others wait until the code is performed by the thread which entered this block first. This can be helpful for computing of the maximal possible time step dt

```
if(dt_new < dt){
    #pragma omp critical{
        if(dt_new < dt){
            dt = dt_new;
        }
    }
}
```

To lock a variable we need to use `omp_lock_t` variable. The lock can be written either in the class of the finite element or we can have the field of the locks. To explain the principle of the locks, the second possibility was chosen. First of all we need to initialize the array of locks

```
omp_lock_t lock[nt];
```

`nt` means the number of variables (right hand sides). Updating of the variables can be achieved by

```
omp_lock_t lock[nt]; // array of locks
double RHS[nt]; // some array of variables, for example right hand sides of the elements
```

```
#pragma omp parallel{
    ID = omp_get_thread_num();
    for(int i=ID; i < nt; i=i+n_thread){
        omp_set_lock(&(lock[i]));
        RHS[i] = do_something(); // only one thread here at a time; others blocked
        omp_unset_lock(&(lock[i]));
    }
}
```


This syntax is useful for computing of the numerical fluxes across the edges. Because one edge is shared by two finite elements, it may happen that one right hand side is updated by two different threads simultaneously. The computation of the numerical fluxes can be computed as follows

```

omp_lock_t lock[ne]; // array of locks
double RHS[nt]; // right hand sides of the elements
double num_flux;
int l,r,ID;
#pragma omp parallel private(ID,l,r,num_flux){
    ID = omp_get_thread_num();
    for(int e=ID;e < ne; e=e+n_thread){ // loop throught the edges
        l=ET[e];
        r=ET[e+ne];
        num_flux=compute_numerical_fluxes();
        omp_set_lock(&(lock[j]));
        omp_set_lock(&(lock[l]));
        RHS[l] = RHS[l]+ 1*num_flux*Edl[e]; //only one thread here at a time
        if(r!=-1){ // if the edge is not at the boundary of the computational area
            RHS[r] = RHS[r]+ -1*num_flux*Edl[e];
        }
        omp_unset_lock(&(lock[j]));
        omp_unset_lock(&(lock[l]));
    }
}

```

here ne means the number of the edges, $Edl[e]$ is length of the edge and matrix \mathbf{ET} is described in Section 5.3. Above written scheme is valid only for FVM. Within the DGFEM, *for* loop through the integrating points, weights and basis functions must be added. Adding of the numerical flux value differs only by the sign, it is because of the normal vector of the edge which is pointing from the left finite element to the right one. This method of computing numerical flux can be used within DGFEM only if the same number of integrating points for the curve integral in each finite element is used (it means the same degree of basis functions). If we would like to avoid the usage of the locks, we can iterate over the finite elements/volumes and computed the numerical fluxes. However the same numerical flux over an edge must be computed twice or the right hand side must be computed with excessive *for* loop and matrix \mathbf{Tzn} as described in Section 5.5.

The time iteration cannot be done before the right hand side is computed. To stop all threads before the time integration and force them to wait till all of them reach this point we can use the command

```
#pragma omp barrier
```

With the previously described commands we can sketch the diagram of the parallel processes. For the sake of simplicity, only Euler time integration is shown in Figure 6.1.

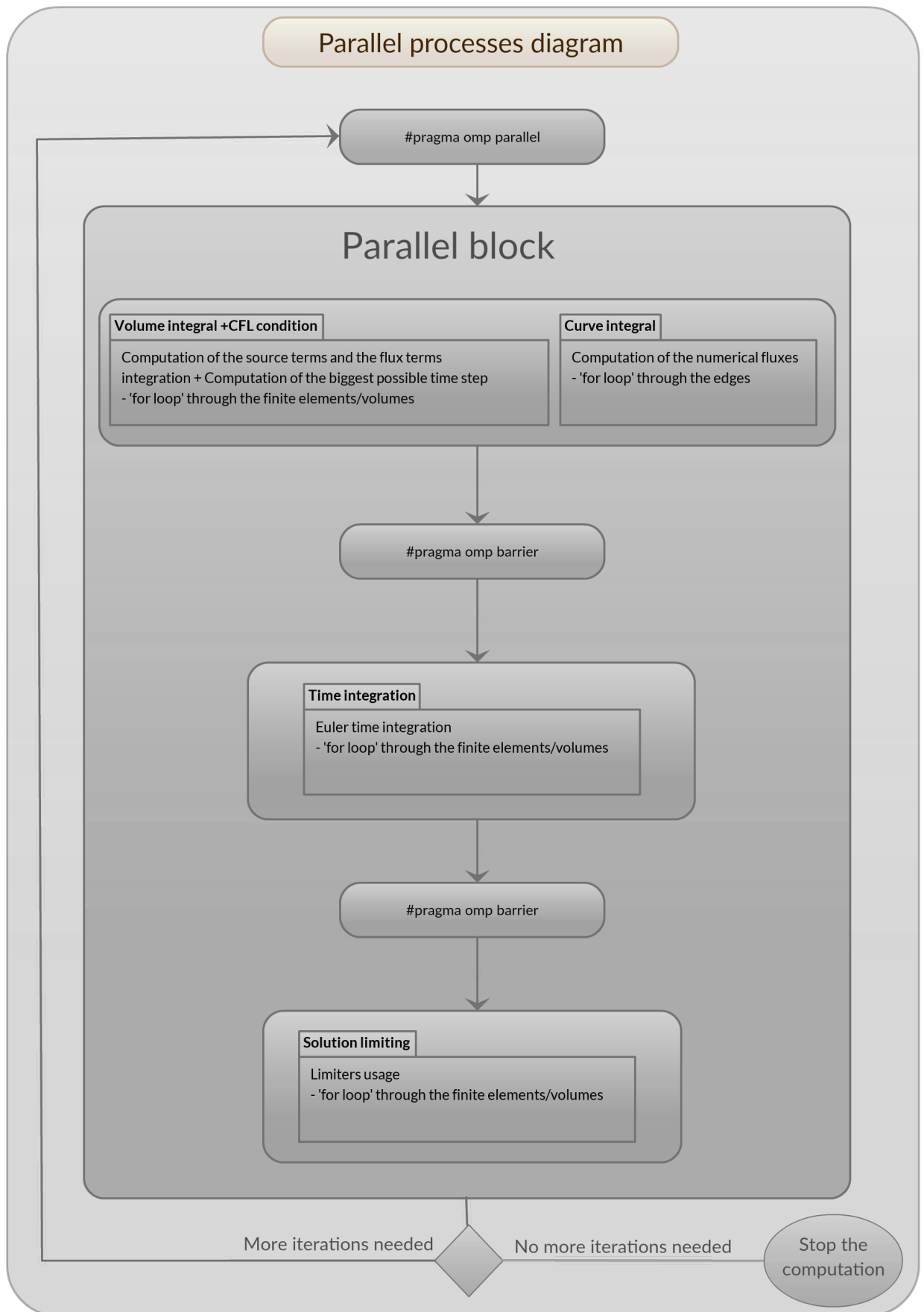


Figure 6.1: The scheme of the parallel processes in Euler time integration.

Chapter 7

Results

In the following chapter the numerical results and validations of the developed software and numerical methods are shown. The chapter is divided into two parts. In the first part, the solver is validated against analytical solutions. The comparison with experimental data is discussed in the second part. Solver used for the numerical simulation is written in C++ language and visualised with help of Python language and Mayavi library.

7.1 Comparisons with Analytical Solutions

In this section the comparison of the numerical results with the analytical solutions of the SWE is described.

7.1.1 2D Dam-break

The dam-break problem has become de facto standard test in validation of Shallow Water Equations solvers. The details of the analytical solution of rectangular dam break were summarized by Dressler in [2]. This section was focused on the comparison of the numerical fluxes and different wet/dry interface treatment during the flooding process.

Initial conditions are shown in Figure 7.1. The water depth $h_0 = 0.6$ m and length $x_0 = 0.6$ m. In the time $t = 0$ s the barrier is instantaneously removed and mass of the fluid collapses under the gravity. Analytical solution of this problem was firstly given by Ritter in 1892 [104] who used the general formulae given by Saint-Venant [1].

7.1.1.1 Surge front velocity

Analytical solution of the position of the surge front-face (wet/dry interface) is $x = x_0 + t \left(2c_0 - At^{-\frac{1}{3}} \right)$ where A is a constant and $c_0 = \sqrt{gh_0}$ is the wave velocity [4]. It was proven by the experiments of A. Schoklitsch [105] that the real celerity is lower because of the bed friction.

Many approaches of finding the front face numerical velocity can be used. In the following the different approaches will be tested and compared with analytical solution to validate the numerical solver. First, the numerical value from the vector of the conservative values \mathbf{W} at the finite volume containing the front-face can be directly computed as $u = \frac{(hu)}{h}$. Second, considering that x_f^n means

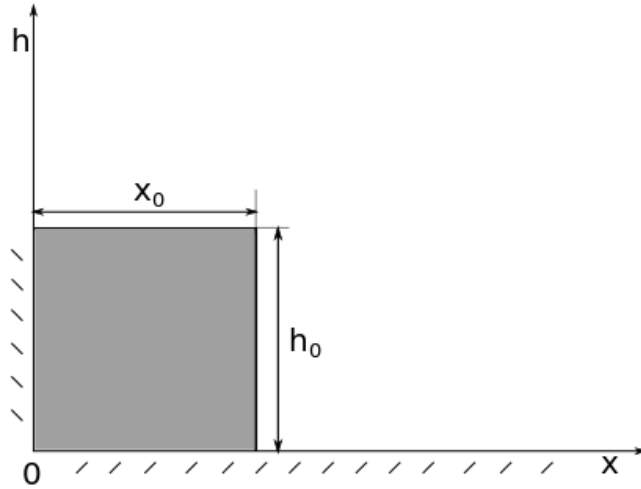


Figure 7.1: Initial conditions of the dam-break.

the position of the wet/dry interface at n^{th} time level, the front-face shift $\delta x = x_f^n - x_f^{n-1}$ can be tracked and velocity computed by division $u_{dt} = \frac{\delta x}{t^{n+1} - t^n}$. Maximum velocity $u_m = \max\left(\frac{(uh)_i}{h_i}\right)$ over the computational domain Ω and velocity u_f computed as whole front-face shift $(x_f - x_0)$ divided by the time t are also tracked.

Following results were computed by first-order Roe scheme for the $\mu_\lambda = 0.5$. Time integration was done by Euler's scheme with CFL=0.5. The position of the front-face x_f is considered to be the position of the edge between wet and dry finite volume cell. The cell is considered to be wet if water depth is higher than $1e-7$ m. The plot of these velocities compared by the analytical solution with $A=0.18$ is shown in Figure 7.2.

All approaches give similar results. Numerical velocity in the finite volumes corresponds to the velocity of the fluid flow and the analytical velocity. The extreme peak of u_{dt} at the beginning of the simulation is caused by the fact that water flows to the adjacent finite volume during the first iteration independently on Δt and Δx . Surprisingly the maximum velocity isn't at the front-face of the surge but is located behind the front-face. This distance is increasing in time.

7.1.1.2 Analytical solution of the Water Level and Velocity

Following benchmark compares Roe and HLL solvers with and without linear approximation for the finite volume method and also compares novel DGFEM limiting method with the reference scheme. The water level of the fluid mass can be expressed analytically. For the frictionless surge in a horizontal channel, the analytical solution was proposed by Barre de Saint-Venant in 1871 [1]. This analytical solution is [106]

$$\frac{x}{t\sqrt{gh_0}} = 2 - 3\sqrt{\frac{h}{h_0}} \quad \text{for} \quad -1 \leq \frac{x}{t\sqrt{gh_0}} \leq 2. \quad (7.1)$$

The distance of the front face from the damp site is $2\sqrt{gh_0} \cdot t$ and the distance of the back wave from the damp site is $\sqrt{gh_0} \cdot t$. The velocity of the flow is given by

$$u(x, t) = \frac{2}{3} \left(\frac{x - x_0}{t} + \sqrt{gh_0} \right). \quad (7.2)$$

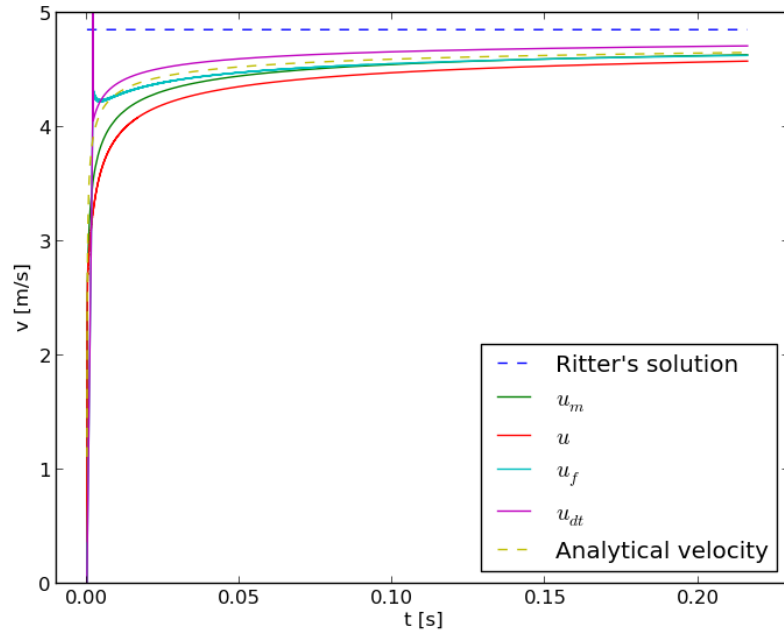


Figure 7.2: Front-face velocities computed by the solver.

The numerical results were computed for the dam break problem with length of computational domain 2.6 m. The initial depth 0.6 m was set for the domain where $x < 1$ m. Fourt order Runge-Kutta time integration with CFL=0.5 was used and μ_λ was set to 0.5. The simulation was stopped at the time $t = \frac{1.5}{2\sqrt{gh_0}} \approx 0.309$ s.

For the numerical results, plotted in Figure 7.3, the coarse mesh was used to highlight the difference between basic schemes and schemes with implemented linear reconstruction described in Section 2.5. It can be seen that the linear reconstruction decreases the numerical viscosity.

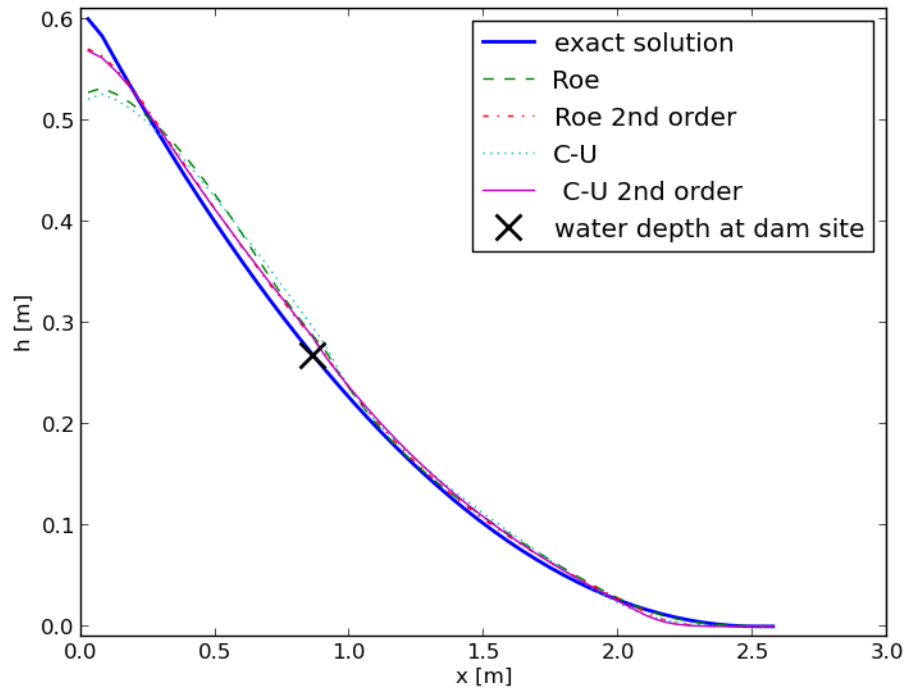
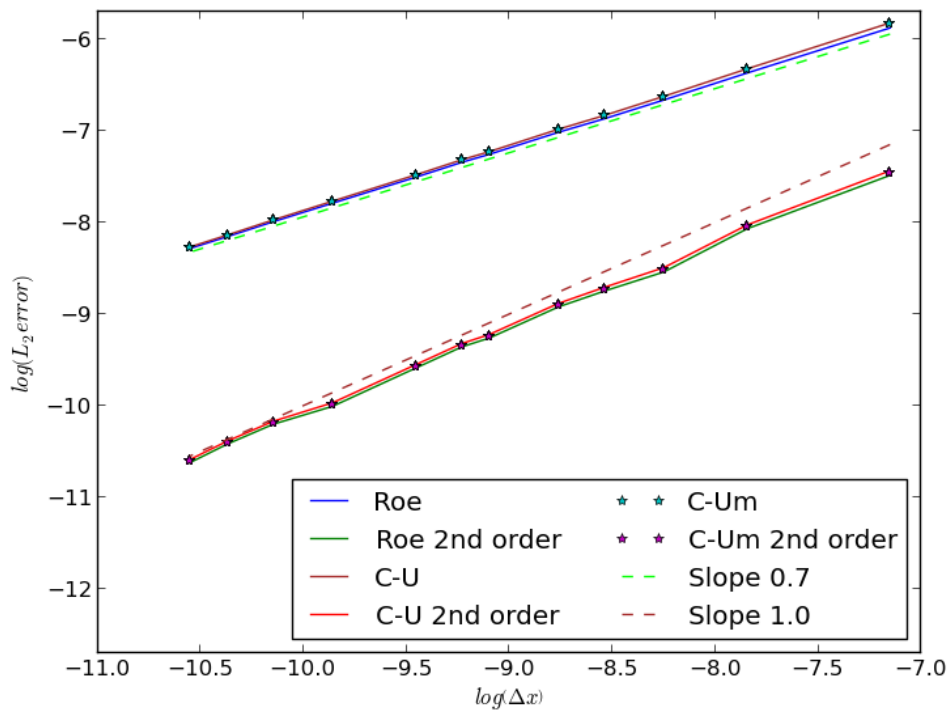
It should be noticed that the water depth at the dam site is constant in time and is equal to $\frac{4}{9}h_0$ [4]. In Figure 7.3, this point is shown by the black cross. Agreement between theoretical and numerical solution is not very good at the point where the dam was set. This is caused by the problem mentioned in Section 2.2.3. Velocity at the dam site is constant and equal to $\frac{2}{3}\sqrt{gh_0}$ [4] thus the Froude number is $F_r = 1$. This problem can be fixed by the entropy correction (2.83). The results with this correction can be seen in Figure 2.6.

Table 7.1 shows the L_2 errors of Roe scheme, HLL scheme and HLL scheme with our proposed modification (2.107) (denoted as HLL-m). L_2 error is defined as

$$L_2 \text{ error} = \sqrt{\frac{\sum_{i=1}^{n_t} (h_a(x_i) - h_n(x_i))^2 |\Omega_i|}{\sum_{i=0}^{n_t} |\Omega_i|}}, \quad (7.3)$$

where n_t is the number of finite volumes, $h_a(x_i)$ means analytical and $h_n(x_i)$ numerical solution of the water depth at the position x_i .

Dependency $\log(L_2 \text{ error})/\log(\Delta x)$ is shown in Figure 7.4. The order of accuracy of the schemes can be deduced from the slopes of the lines. For comparison is in the graph drawn line with slope 1.0 and line with slope 0.7. Even if both HLL and Roe schemes are formally of first order of accuracy the slope of L_2 error shows that real order of accuracy is lower. This problem is caused because the

Figure 7.3: Fifty finite volumes. $\Delta x = 0.052\text{m}$.Figure 7.4: L_2 error of first and second order schemes.

initial conditions are described by the discontinuous function and the accuracy of the schemes decays near the discontinuities. Proof of this claim can be found in [107]. The reader is referred to [108] to see more details on discontinuous solutions. The order of accuracy is increased to one when the linear reconstruction is used. It can be also noticed that the modification of the eigenvalues (2.107) doesn't decrease the accuracy of HLL scheme.

Table 7.1: L_2 Error of the Water Level

Δx	Roe	HLL	HLL-m	Roe 2nd	HLL 2nd	HLL-m 2nd
2.6e-03	2.7956e-03	2.9551e-03	2.9491e-03	5.5957e-04	5.8809e-04	5.7214e-04
1.3e-03	1.7118e-03	1.7905e-03	1.7893e-03	3.1270e-04	3.2592e-04	3.1866e-04
8.6e-04	1.2738e-03	1.3255e-03	1.3252e-03	1.9491e-04	2.0438e-04	1.9920e-04
6.5e-04	1.0386e-03	1.0767e-03	1.0767e-03	1.5841e-04	1.6504e-04	1.6139e-04
5.2e-04	8.9628e-04	9.2638e-04	9.2640e-04	1.3326e-04	1.3843e-04	1.3558e-04
3.7e-04	7.0399e-04	7.2494e-04	7.2502e-04	9.4830e-05	9.8564e-05	9.6491e-05
3.2e-04	6.4529e-04	6.6345e-04	6.6352e-04	8.6085e-05	8.9280e-05	8.7505e-05
2.6e-04	5.4869e-04	5.6290e-04	5.6299e-04	6.8372e-05	7.0951e-05	6.9513e-05
1.7e-04	4.1286e-04	4.2195e-04	4.2202e-04	4.4968e-05	4.6740e-05	4.5752e-05
1.3e-04	3.3731e-04	3.4390e-04	3.4395e-04	3.6987e-05	3.8217e-05	3.7531e-05
1.0e-04	2.8660e-04	2.9169e-04	2.9174e-04	2.9787e-05	3.0768e-05	3.0222e-05
8.6e-05	2.5175e-04	2.5588e-04	2.5592e-04	2.4297e-05	2.5138e-05	2.4670e-05

After comparison of the numerical schemes, the comparison of the limiting methods used for DGFEM was done. The new novel limiting method described in Section 3.2.2.3 was compared with the global limiting method when the minmod limiter was used as a shock detector. In all schemes the wet/dry interface described in Section 3.2.2.3 was used. This comparison can be seen in Table 7.2. The numerical results confirm the quality of suggested limiting process.

Table 7.2: Comparison of Limiting Processes of 2D Dam Break, L_2 Error of the Water Level

Number of finite elements	Δx	Novel limiting process	Global limiting process
10	2.6000e-01	2.3535e-02	3.1655e-02
20	1.3000e-01	1.6970e-02	2.2354e-02
30	8.6667e-02	1.4752e-02	1.8720e-02
50	5.2000e-02	6.0418e-03	8.9157e-03
100	2.6000e-02	4.3580e-03	5.7021e-03
300	8.6667e-03	1.3922e-03	2.0639e-03
500	5.2000e-03	7.9793e-04	1.2696e-03
1000	2.6000e-03	4.8845e-04	7.4913e-04

7.1.1.3 Wet/Dry Interface

In Section 2.2.4, there were described different methods of the velocity treatment at the wet/dry interface:

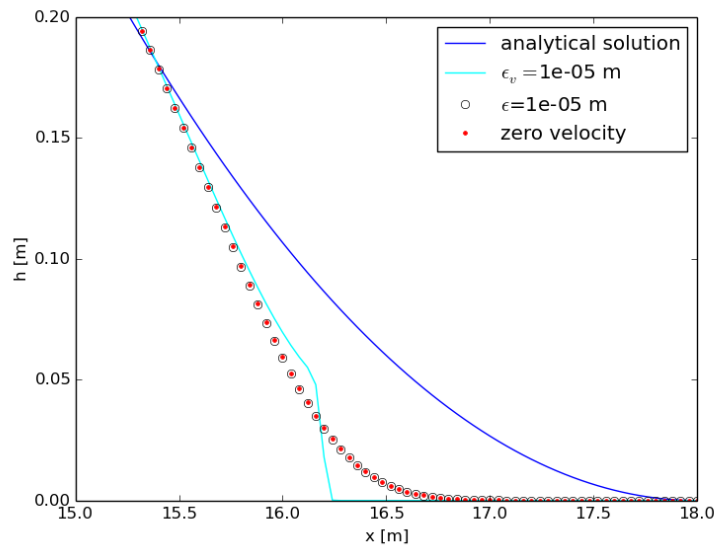


Figure 7.5: Tip of the surge wave.

- Method of neglecting computations when the water depth is smaller than some constant ϵ
- Method of setting zero velocity at the wet/dry interface
- Computation of the velocity by $u = \frac{\sqrt{2}h(hu)}{\sqrt{h^4 + \max(h^4, \epsilon_v)}}$

In previous cases the first method was used. The comparison of all these methods follows.

Figure 7.5 shows numerical results of all these methods. First two methods are giving approximately the same results. But Figure 7.5 shows that the third method with the correction (2.86) causes non-physical solution at the tip of the surge. This method gives better results with smaller ϵ_v . But detailed view in Figure 7.6 shows that some non-physical behaviour remains. Thus it is recommended to use the first method and neglect the computations if the water depth as smaller than some constant ϵ . The magnitude of this constant is arbitrary and depends on the study case.

7.1.1.4 Comparison of the Time-Integration Methods

In this section the time integration methods from Chapter 4 are evaluated. HLL scheme with linear reconstruction was used in this test. The simulation of the dam-break was stopped in the time $t = \frac{1}{2\sqrt{gh_0}}$ s. At the end of the simulation was computed L_2 error of the results in the area $x \geq x_0$. The constants used in simulation were CFL=0.5, $\phi = 0.35$ and $\epsilon_v = 10^{-15}$.

In Table 7.3, it can be seen that the accuracy of Second-Order and Fourth-Order Runge-Kutta methods does not differ a lot and usage of Second-Order Runge-Kutta is sufficient. The numerical result also show, that this method is stable and relatively low computational demands make second order Runge-Kutta method more favourable.

7.1.2 Flow over the Bump

This section is focused on comparison between FVM and DGFEM. The steady state flow over the bump is computed in this testing case. The inlet is set on the left side of the computational domain

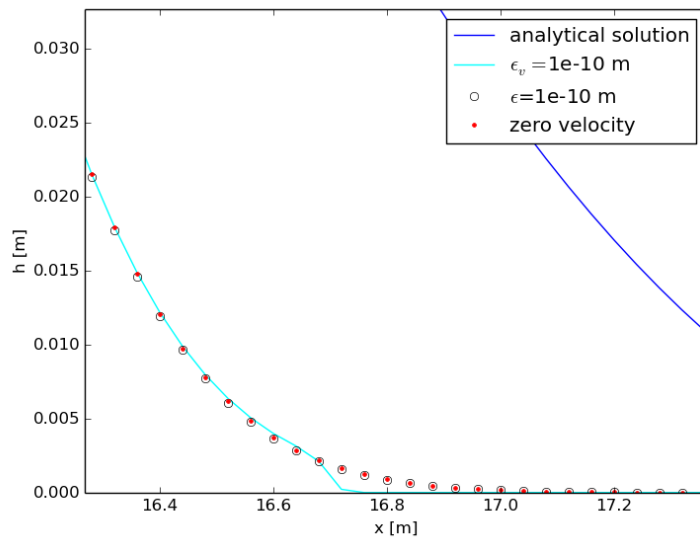


Figure 7.6: Detail of the tip of the surge wave.

Table 7.3: L_2 Error of the Time-Integration Methods

Finite Volumes	Δx	Euler	2nd order Runge-Kutta	4th order Runge-Kutta
500	5.200000e-03	1.169065e-03	1.167722e-03	1.164827e-03
750	3.466667e-03	9.754459e-04	9.537336e-04	9.525770e-04
1000	2.600000e-03	7.763235e-04	7.664815e-04	7.651388e-04
2000	1.300000e-03	6.102051e-04	5.026562e-04	5.021041e-04
3000	8.666667e-04	1.057995e-03	3.770444e-04	3.762079e-04
4000	6.500000e-04	1.158979e-03	3.249432e-04	3.230943e-04
7500	3.466667e-04	1.261652e-03	5.456766e-04	5.279745e-04
10000	2.600000e-04	1.466307e-03	7.381620e-04	7.217265e-04

and the outlet is set on the right side of the computational domain. Analytical solutions of steady state flows over the bump can be found for example in [5] or [109]. Besides the water level, the volume discharge rate, defined as $q = h \cdot u$, is also shown. Semi-analytic solutions were obtained by SWASHES software which is free-ware software downloadable from the repository <http://www.univ-orleans.fr/mapmo/soft/SWASHES>

The length of the computational domain is $L=25$ m with a topography given by

$$B(x) = \begin{cases} 0.2 - 0.05(x - 10)^2 & \text{if } 8 \text{ m} < x < 12 \text{ m,} \\ 0 & \text{else.} \end{cases} \quad (7.4)$$

In the following the subcritical flow, transcritical flow and transcritical flow with the shock are discussed. Initial conditions for all these three types of flow are

$$\begin{aligned} h(x) + B(x) &= 2, & x \in \Omega, \\ u(x) &= 0, & x \in \Omega. \end{aligned} \quad (7.5)$$

The numerical solutions of finite volume method with linear reconstruction and discontinuous

Galerkin method are compared. HLL scheme is used for the computation of the numerical flux. In DGFEM, the Taylor basis functions were used for the approximation of the solution and artificial viscosity was used to avoid spurious oscillations. The cell was defined as troubled if residuum R_k (3.23) was larger than 0.008 and artificial viscosity in (3.24) was set to $\epsilon = 0.8$. In case of FVM the solution was limited by minmod limiter. The simulation is stopped when the steady state flow residuum R_h is smaller than $1e-10$. This residuum is defined as

$$R_h = \sum_{i=1}^{n_t} \frac{|h_i^n - h_i^{n-1}|}{\Delta t} \quad (7.6)$$

where h_i^n is the numerical value of the water depth in i^{th} finite volume cell in n^{th} time level and Δt is the time step.

In the case of the subcritical flow, the water depth is given by the resolution of

$$h^3(x) + \left(B(x) - \frac{q_{inlet}^2}{2gh_{outlet}^2} - h_{outlet} \right) h^2(x) + \frac{q_{inlet}}{2g} = 0, \quad \forall x \in [0, L] \quad (7.7)$$

where q_{inlet} is fixed volume discharge rate at the inlet and h_{outlet} is fixed water level at the outlet. The boundary conditions are

$$\begin{aligned} q_{inlet} &= 4.42 \text{ m}^2/\text{s}, \\ h_{outlet} &= 2 \text{ m}. \end{aligned} \quad (7.8)$$

In Figures 7.7 and 7.8, the comparison of the numerical results of the subcritical steady state flow

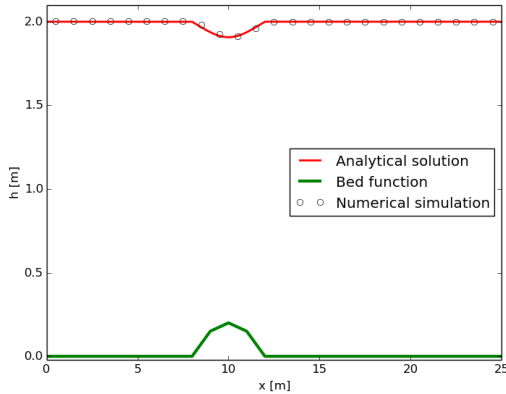


Figure 7.7: Water depth of the subcritical steady state flow computed by the FVM.

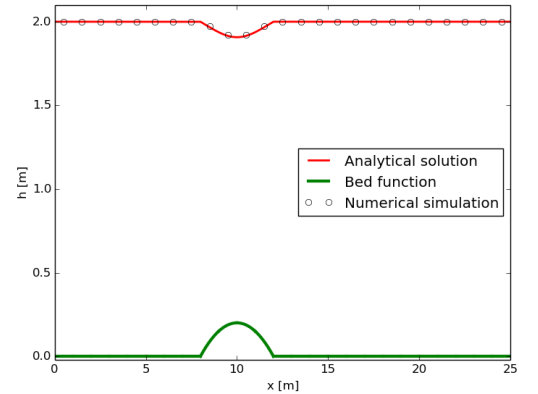


Figure 7.8: Water depth of the subcritical steady state flow computed by the DGFEM.

are shown. The results were computed for coarse mesh created by 25 of finite volume cells. In case of FVM the bed function can be described only by the linear functions (see Figure 7.7). DGFEM allows to describe the bed function by polynomials, in our simulation the bed function was described by third order polynomials (see Figure 7.8). The analytical and numerical solutions were compared by the L_2 error. This error is defined as

$$L_2 \text{ error} = \sqrt{\sum_{i=1}^{n_p} \frac{(h^n(x_i) - h^a(x_i))^2}{n_p}} \quad (7.9)$$

where h^n is the numerical and h^a analytical solution in point x_i , n_p means the number of these points. The points x_i and analytical solutions $h^a(x_i)$ were generated automatically by SWASHES software.

Table 7.4: L_2 error of the subcritical flow

Number of finite volume cells	FVM error	DGFEM-2 error	DGFEM-3 error	DGFEM-4 error	DGFEM-5 error	DGFEM-6 error
25	5.4987e-03	2.5164e-03	5.6094e-05	4.1755e-06	1.4415e-06	1.3580e-06
50	1.8398e-03	6.3393e-04	7.0966e-06	1.3844e-06	1.3577e-06	1.3578e-06
100	5.8673e-04	1.5895e-04	1.6163e-06	1.3578e-06	1.3578e-06	1.3578e-06

In Table 7.4 the comparison for 250 000 of these points is shown. The abbreviation DGFEM-2 means that the numerical solution was computed by the second order of accuracy, DGFEM-3 means the third order of accuracy etc. The discharge rates of the subcritical flow are shown in Figures 7.9 and 7.10. Let us highlight that in the case when there is no appearance of the shock in the computational area the high order of accuracy can substitute high resolution. The sixth order of accuracy DGFEM is giving the same error for 25, 50 and 100 of finite elements.

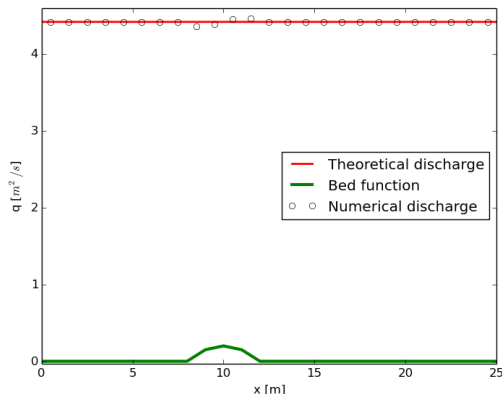


Figure 7.9: The discharge rate of the subcritical steady state flow computed by the FVM.

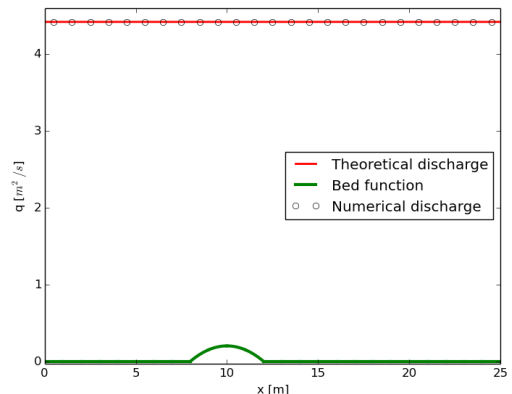


Figure 7.10: The discharge rate of the subcritical steady state flow computed by the DGFEM.

The analytical solution of the water depth in case of transcritical flow without shock is given by the resolution of [5]

$$h^3(x) + \left(B(x) - \frac{q_{inlet}^2}{2gh_M^2} - h_M - B_M \right) h^2(x) + \frac{q_{inlet}^2}{2g} = 0, \quad \forall x \in [0, L] \quad (7.10)$$

where $B_M = \max_{x \in [0, L]} B(x)$ and h_M is the corresponding water depth. Boundary conditions are

$$\begin{aligned} q_{inlet} &= 1.53 \text{ m}^2/\text{s}, \\ h_{outlet} &= 0.66 \text{ m}. \end{aligned} \quad (7.11)$$

The comparison of the analytical and numerical solution of the water depth is shown in Figures 7.11 and 7.12. The rate of discharge is shown in Figures 7.13 and 7.14.

The L_2 error, defined by (7.9), is shown in Table 7.5. Again 250 000 of points were used to compute this error.

The analytical solution of the water depth in case of transcritical flow with shock is given by the

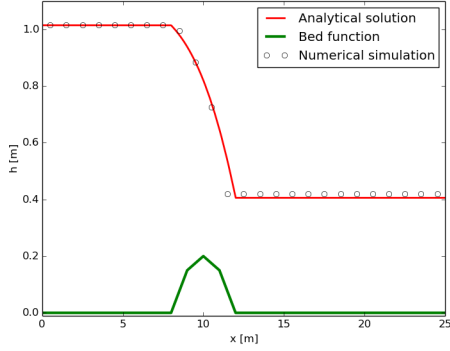


Figure 7.11: Water depth of the transcritical steady state flow without shock computed by the FVM.

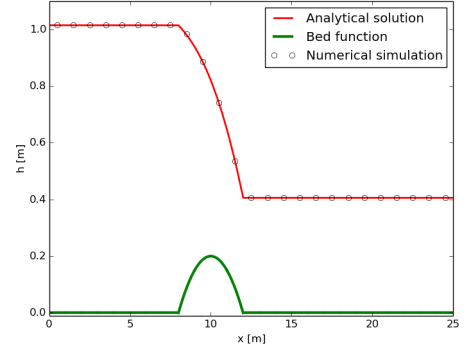


Figure 7.12: Water depth of the transcritical steady state flow without shock computed by the DGFEM.

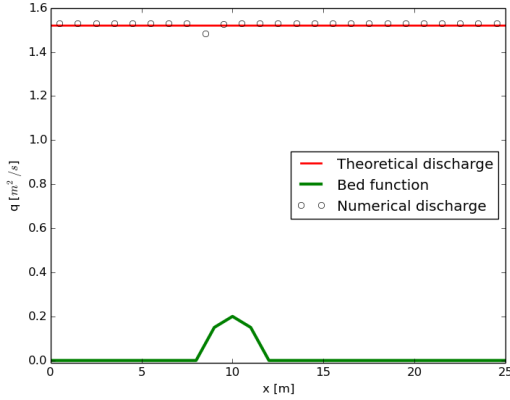


Figure 7.13: The discharge rate of the transcritical flow without shock computed by the FVM.

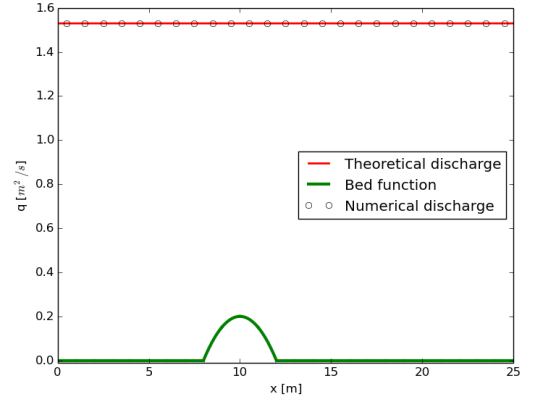


Figure 7.14: The discharge rate of the transcritical flow without shock computed by the DGFEM.

resolution of [5]

$$\begin{aligned}
 h^3(x_{shock}) + \left(B(x_{shock}) - \frac{q_{inlet}^2}{2gh_M^2} - h_M - B_M \right) h^2(x_{shock}) + \frac{q_{inlet}^2}{2g} &= 0, \\
 h^3(x_{shock}) + \left(B(x_{shock}) - \frac{q_{inlet}^2}{2gh_{outlet}^2} - h_{outlet} \right) h^2(x_{shock}) + \frac{q_{inlet}^2}{2g} &= 0, \\
 q_{inlet}^2 \left(\frac{1}{h_1} - \frac{1}{h_2} \right) + \frac{g}{2} (h_1^2 - h_2^2) &= 0
 \end{aligned} \tag{7.12}$$

where h_1 and h_2 are the water depths upstream and downstream respectively. The position of the shock x_{shock} is located thanks to the third relation in system (7.12) which is known as Rankine-Hugoniot's relation. The reader is referred to [110] to see the details. Boundary conditions can be set as

$$\begin{aligned}
 q_{inlet} &= 0.18 \text{ m}^2/\text{s}, \\
 h_{outlet} &= 0.33 \text{ m}.
 \end{aligned} \tag{7.13}$$

In Figures 7.15 and 7.16 the water level of transcritical flow with shock is shown. The discharge rate can be seen in Figures 7.17 and 7.18. And L_2 error is shown in Table 7.6.

As the numerical results show, the accuracy grows with the order of the basis functions, although it is not feasible to use higher than fourth order of accuracy. Higher order of basis function doesn't

Table 7.5: L_2 error of the transcritical flow without shock

Number of finite volume cells	FVM error	DGFEM-2 error	DGFEM-3 error	DGFEM-4 error	DGFEM-5 error	DGFEM-6 error
25	2.8854e-02	1.3794e-03	2.6663e-04	2.6554e-04	2.7573e-04	2.6760e-04
50	1.0923e-02	3.0813e-04	4.0530e-05	4.0391e-05	4.0960e-05	4.0588e-05
100	3.9630e-03	7.0970e-05	6.0961e-06	6.0824e-06	6.0993e-06	6.0925e-06

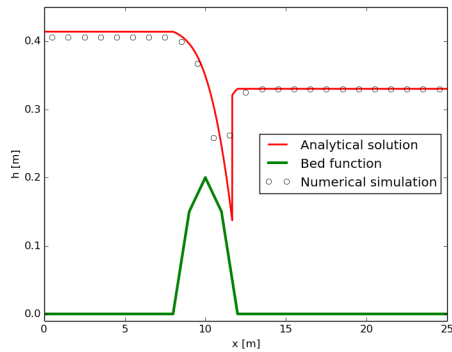


Figure 7.15: Water depth of the transcritical steady state flow with shock computed by the FVM.

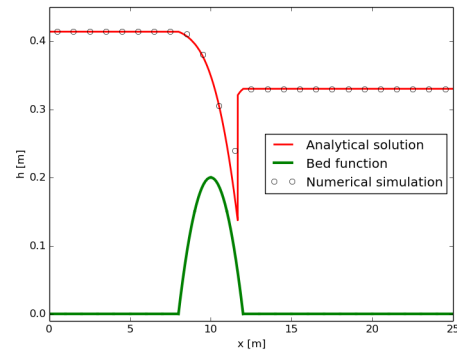


Figure 7.16: Water depth of the transcritical steady state flow with shock computed by the DGFEM.

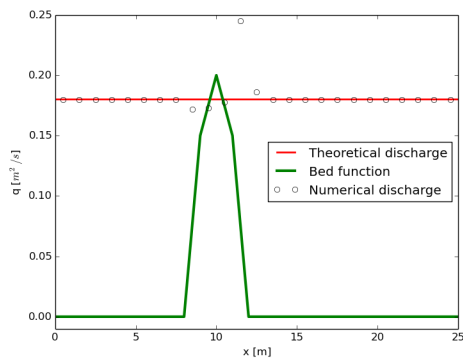


Figure 7.17: The discharge rate of the transcritical flow with shock computed by the FVM.

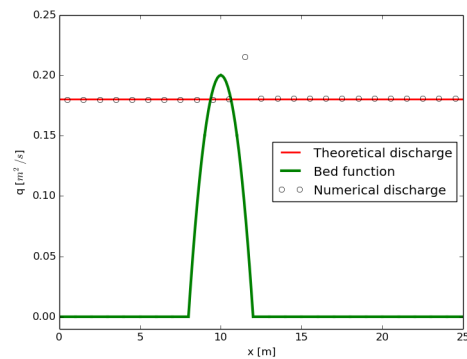


Figure 7.18: The discharge rate of the transcritical flow with shock computed by the DGFEM.

bring any significant improvement and method becomes computationally very demanding. DGFEM gives better results than FVM mainly because of the bed function which in DGFEM case described by the polynomial and thus resolved more accurately. But DGFEM surpasses FVM also in case when both, the bed function and water depth, are described by the piece-wise linear functions. The L_2 error of this simulation is shown in Table 7.7.

The conclusion of this comparison is following. It is feasible to use discontinuous Galerkin method in case when we are expecting no shocks or big discontinuities in the computational domain. Especially when the bed function can be described by the polynomials. In this case, even very coarse mesh with

Table 7.6: L_2 error of the transcritical flow with shock

Number of finite volume cells	FVM error	DGFEM-2 error	DGFEM-3 error	DGFEM-4 error	DGFEM-5 error	DGFEM-6 error
25	1.8518e-02	1.2170e-02	8.5854e-03	8.6700e-03	7.1875e-03	6.5647e-03
50	1.1544e-02	6.6635e-03	5.8915e-03	5.4981e-03	6.5960e-03	6.0943e-03
100	7.5337e-03	5.2003e-03	5.1978e-03	4.8596e-03	5.2210e-03	9.8696e-03

Table 7.7: L_2 error of the simulations with piece-wise linear approximation of the bed function

Number of finite volume cells	Subcritical flow		Transcritical flow without the shock		Transcritical flow with the shock	
	FVM	DGFEM-2	FVM	DGFEM-2	FVM	DGFEM-2
25	5.4987e-03	4.7431e-03	2.8854e-02	6.5852e-03	1.8518e-02	1.2694e-02
50	1.8398e-03	1.1916e-03	1.0923e-02	2.0435e-03	1.1544e-02	7.0095e-03
100	5.8673e-04	2.9840e-04	3.9630e-03	6.6363e-04	7.5337e-03	5.2452e-03

high degree of basis functions can give very precise solution (see Table 7.4). In this case of subcritical flows the computational demands were even smaller than the computational demands of FVM. More than 2000 of finite volumes had to be used to gain error smaller than $1.3580e-06$ by finite volume method. The same error can achieve discontinuous Galerkin method only by 25 finite elements and sixth order of accuracy. But in case when some shock is appearing in the computational domain, DGFEM does not give so convincing results in comparison with FVM and computational demands are much higher.

The accuracy also decreases for the subcritical flow when the bed function is not described by higher degree the polynomials, but only by linear functions. Computational demands and L_2 error of the schemes is shown in Table 7.8.

Table 7.8: L_2 error of the subcritical flow for 25 finite cells and linear bed function approximation

Method	L_2 error	Time [s]
FVM	5.4987e-03	0.18
DGFEM-2	4.7431e-03	2.93
DGFEM-3	4.7594e-03	5.87
DGFEM-4	4.7595e-03	24.67
DGFEM-5	4.7595e-03	75.77
DGFEM-6	4.7595e-03	192.83

Note 4. The computational demands of DGFEM method are smaller than FVM when the bed function is described by higher polynomials. To gain error smaller than $1.35e-06$ by FVM, 2000 of finite volumes had to be used and computation took longer than one hour to gain the steady state. DGFEM method reached this result for 25 of finite elements in 192 seconds.

7.1.3 Planar Surface in Parabolic Bowl without Friction

One of the classical benchmarks for testing bed slope and wetting and drying processes is the planar water surface in a parabola without friction. The exact solution of the two-dimensional problem was derived in [111]. Here, the one-dimensional benchmark presented in [5] is tested. The topography is a parabolic shape given by

$$B(x) = h_0 \left(\frac{1}{a^2} \left(x - \frac{L}{2} \right)^2 - 1 \right). \quad (7.14)$$

where the length of the computational domain $L = 4$ m, $h_0 = 0.5$ m and the constant $a=1$. Exact solution is given by

$$h(x, t) = \begin{cases} -h_0 \left(\left(\frac{1}{a} \left(x - \frac{L}{2} \right) + \frac{C}{\sqrt{2gh_0}} \cos \left(\frac{\sqrt{2gh_0}}{a} t \right) \right)^2 - 1 \right) & \text{for } x_1(t) \leq x \leq x_2(t), \\ 0 & \text{otherwise,} \end{cases} \quad (7.15)$$

with location of the wet/dry interfaces at time t being calculated as

$$\begin{aligned} x_1(t) &= -\frac{1}{2} \cos \left(\frac{\sqrt{2gh_0}}{a} t \right) - a + \frac{L}{2}, \\ x_2(t) &= -\frac{1}{2} \cos \left(\frac{\sqrt{2gh_0}}{a} t \right) + a + \frac{L}{2} \end{aligned} \quad (7.16)$$

and

$$C = \sqrt{\frac{2gh_0}{2a}}. \quad (7.17)$$

The initial conditions of the water depth are given by (7.15) for $t = 0$ s.

This benchmark is very sensitive to the viscosity of the numerical scheme and also gives very good test of the scheme behaviour at wet/dry interface. The water mass oscillates in the bowl and the time of one oscillation is $T = a\pi\sqrt{2gh_0} \approx 1.003$ s. The comparison of novel limiting method described in Section 3.2.2.3 and reference method described by Kesserwani in [52] is shown in Table 7.9. The original Kesserwani's paper does not cover the wet/dry interface computations, thus his scheme was modified and for wet/dry computations the method described in Section 2.2.4 was used. The L_2 error in Table 7.9 was computed in time $t=5T \approx 5.015$ s. The numerical accuracy decays by the time because

Table 7.9: L_2 error of the novel and reference scheme for different number of finite elements

Number of finite elements	Δx	Novel solver	Reference solver
20	0.2000	9.55665e-03	1.56004e-02
30	0.1333	7.65887e-03	1.28066e-02
50	0.0800	4.31255e-03	2.03828e-02
100	0.0400	3.41727e-03	9.89428e-03
200	0.0200	2.01553e-03	3.51478e-03
300	0.0133	2.20309e-03	3.70284e-03
500	0.0080	1.66309e-03	2.75976e-03
750	0.0053	1.59126e-03	2.66527e-03
1000	0.0040	1.57110e-03	2.05729e-03

of the numerical viscosity of the scheme. The numerical dissipation can be seen in Figure 7.19 and the L_2 error for different number of oscillations is shown in Table 7.10. There can be noticed that

Table 7.10: L_2 error of the novel scheme for different number of oscillations

Number of periods	DGFEM-3	DGFEM-2	FVM
1	1.96097e-03	5.96472e-03	1.89603e-02
10	4.58955e-03	3.01939e-02	1.04647e-01
100	2.70580e-02	1.23954e-01	1.76875e-01
1000	1.09285e-01	1.76527e-01	1.77150e-01

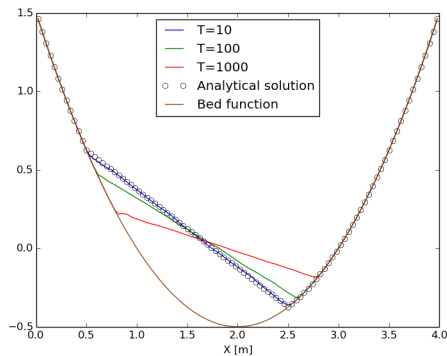


Figure 7.19: The third order DGFEM solution in different time periods.

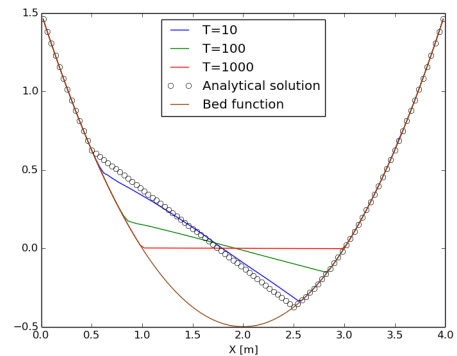


Figure 7.20: The second order DGFEM solution in different time periods.

the second order discontinuous Galerkin solver and second order finite volume method are much more viscous than the third order solver in Figures 7.19, 7.19 and in Table 7.10. This is caused not only by the lower order of accuracy, but also by the velocity treatment (2.86) at the wet/dry interface.

The details of the wet/dry interface, highlighted in Figure 7.21, are shown in in Figures 7.22 and 7.23. It can be seen that meanwhile in second order the water depth takes very small values, the curvature of the third order scheme causes that the water depth is much larger than zero. The results were computed for the $\epsilon_v = 1e - 14$. The modification (2.86) does not causes only the numerical viscosity but for bad choice of the ϵ_v this modification can cause absolutely non-physical behaviour of the scheme as shown in Figure 7.24 for $\epsilon_v = 1e-10$. Thus it is recommended not to use the modification (2.86) but some threshold ϵ as described in Section 2.2.4. The numerical results also show that the scheme brings better results with threshold $\epsilon=1e-10$ than with the ' ϵ_v correction'.

Except of DGFEM methods, this testing case can be used also for validation of novel wet/dry interface within FVM. The comparison with the original scheme described in [41] is shown in Figures 7.25 and 7.26. These numerical results clearly show that modification of the bed source term (2.123) and numerical flux computation (2.127) at the wet/dry interface improve the numerical results of given scheme.

7.2 Comparison with Experiments

In this section the numerical results will be validated against the experimental data. In the first testing case novel mass conservative modification (2.127) is tested. The second experiment validates the final volume software without the bed slope source term. In this benchmark the wet/dry interface flow and 2D numerical fluxes are tested. For the numerical flux HLL scheme with modification (2.107) was

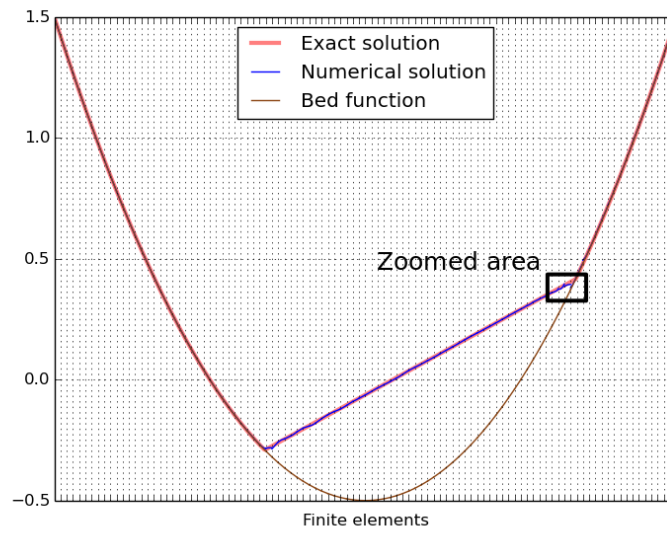
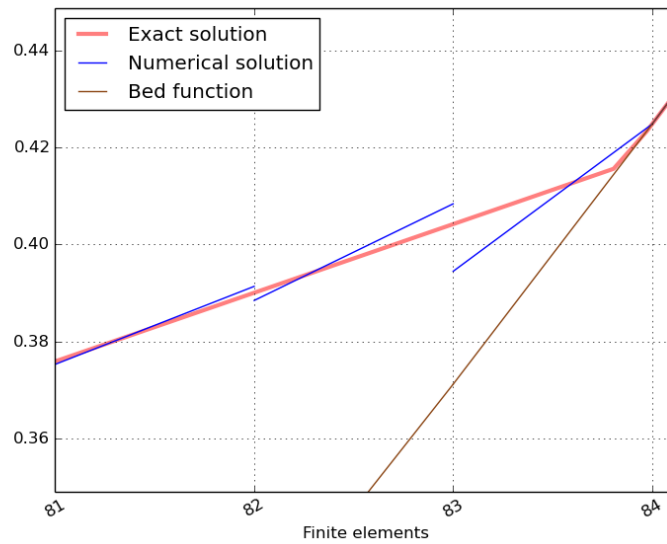
Figure 7.21: Wet/dry interface of the solution in time $t=0.75T$.

Figure 7.22: Detail of the second order DGFEM solution in the zoomed area.

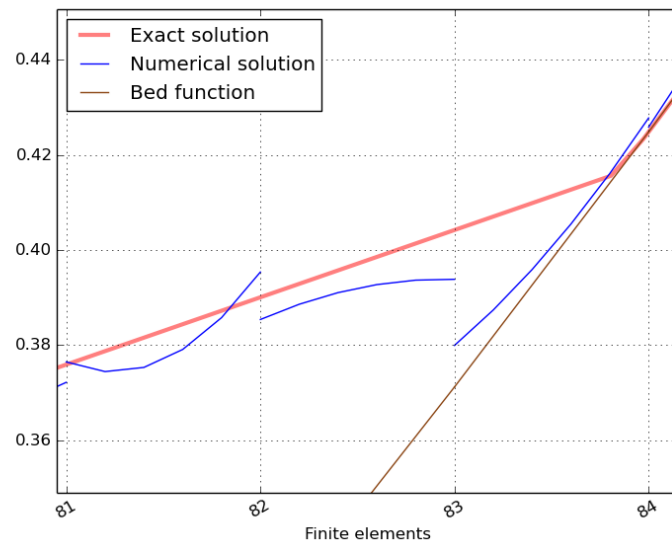


Figure 7.23: Detail of the third order DGFEM solution in the zoomed area.

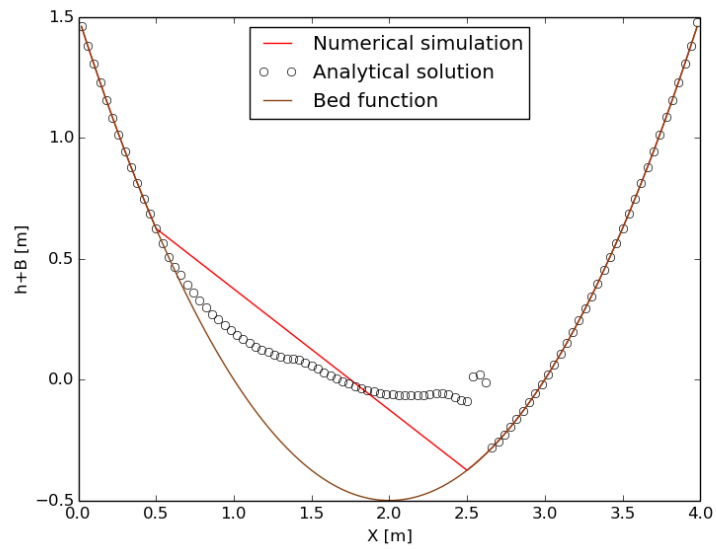


Figure 7.24: Non-physical solution in time $t=10T$ caused by the velocity modification.

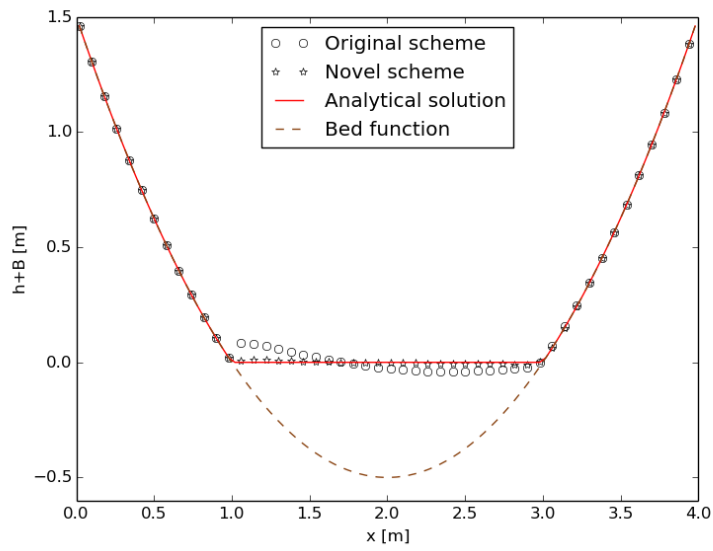


Figure 7.25: Solution at time $t = 5.5T$ simulated for 100 finite volumes.

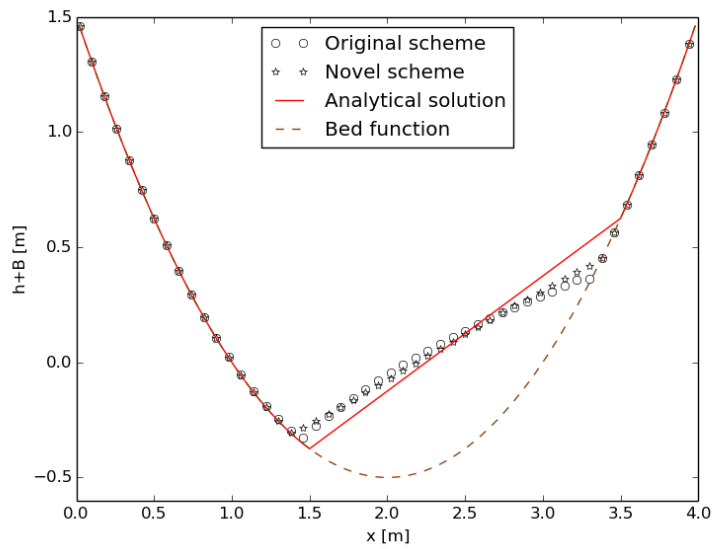


Figure 7.26: Solution at time $t = 5T$ simulated for 100 finite volumes.

used. The third experiment contains non-flat bed topography and the approximation of the bed slope source term is tested for the discontinuous Galerkin method. In this experiment the wet/dry interface and novel limiting process, described in Section 3.3.2, were tested.

7.2.1 Dam Break Over a Triangular Bottom Sill

The first experiment is compared with 1D finite volume software with novel mass conservative modification keeping the 'C-property' condition (2.127).

In experiment published in [64], the wetting and drying processes were captured. The bed function and the initial conditions are shown in Figure 7.27. The channel has a rectangular cross section. The length of the channel is 5.6 m and the width is 0.5 m. The initial water depth h_0 of the still water in the reservoir is 0.111 m. The water gate of the reservoir is set at the position $l_0 = 2.39$ m. The symmetrical bump, located at position 4.45 m, is 0.065 m high and has a slope ± 0.14 . Behind the bump there is a pool of still water with the initial water height 0.02 m. The Manning bed roughness factor is measured experimentally in [64] to be $0.011 \text{ s m}^{-1/3}$.

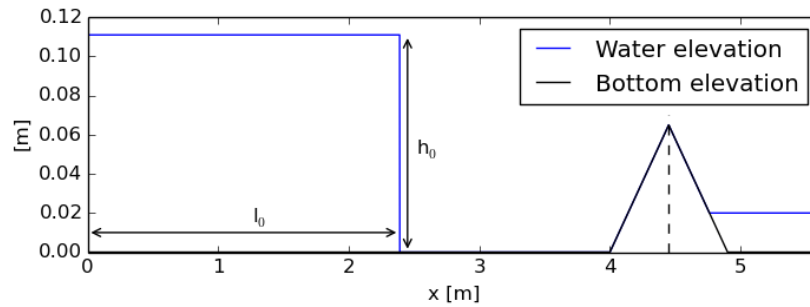


Figure 7.27: Initial conditions of the experiment.

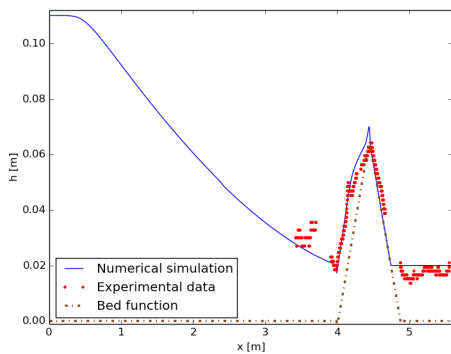


Figure 7.28: Comparison of the simulation and experimental data at time $t = 1.8$ s.

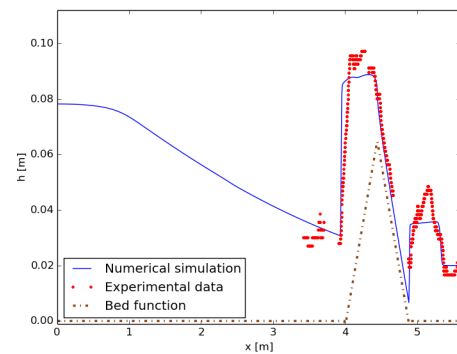


Figure 7.29: Comparison of the simulation and experimental data at time $t = 3.0$ s.

In Figures 7.28, 7.29, 7.30 and 7.31 the simulation results and the experimental data are plotted together. The computational area was discretized by 1000 finite volumes. It is seen that the proposed scheme gives good predictions also for the experimental data with friction term. Moreover the scheme is mass conservative.

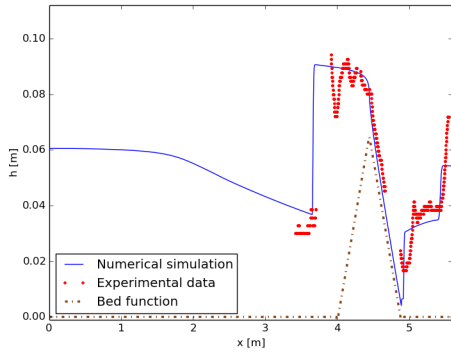


Figure 7.30: Comparison of the simulation and experimental data at time $t = 3.7$ s.

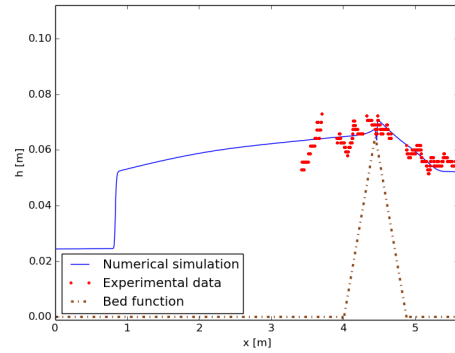


Figure 7.31: Comparison of the simulation and experimental data at time $t = 8.4$ s.

7.2.2 H-dam Experiment

In this section the comparison of the finite volume method with the experiment of L. Fraccarollo and E. F. Toro [79] is shown. H-dam problem is 3D flow in the the experimental flume. This flume is two metres wide and one meter long. In the middle of the front wall is 40 cm wide opening which is closed by the water gate. The flume geometry can be seen in Figure 7.32. During the experiment the water depth at four locations was captured. These points are drawn in Figure 7.33 and their positions in meters are A($x=1.0$, $y=1.0$), B($x=0.18$, $y=1.0$), C($x=0.48$, $y=0.4$) and D($x=1.802$, $y=1.45$). The initial water depth in the reservoir was 0.6 m and there was dry bottom everywhere except the flume.

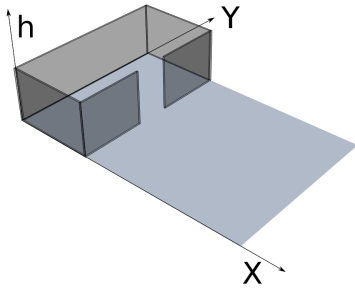


Figure 7.32: Visualization of the H-dam.

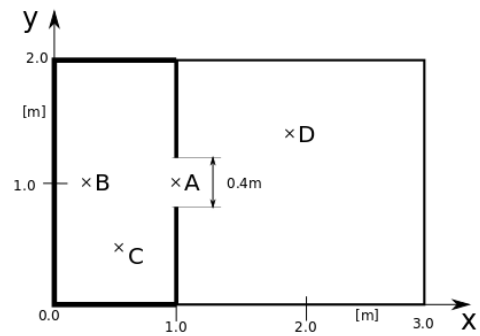
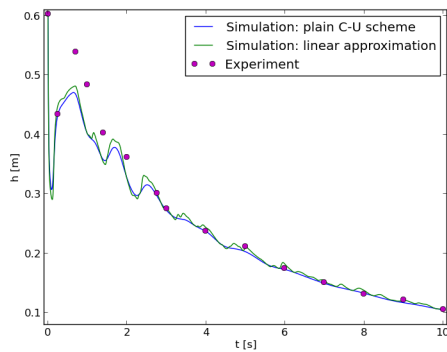
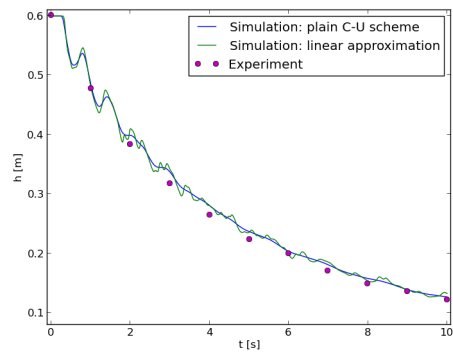
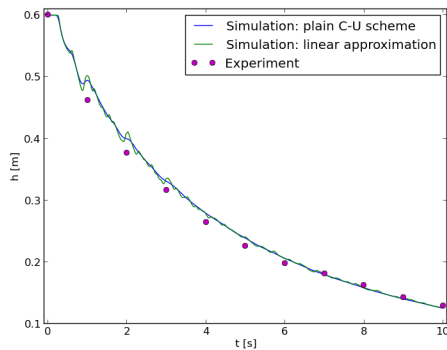
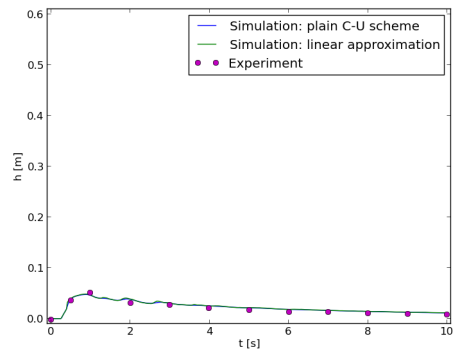


Figure 7.33: Geometry of the H-dam.

The flood-plain boundaries are simulated as all opened and boundary condition is set to outlet with extrapolating all values (depth and celerity of the flux). The domain is discretised into rectangles with $\Delta x = \Delta y = 0.01$ m. The bed friction with Manning's friction coefficient $C_f = 0.0001$ was involved in the simulation, CFL constant was set to 0.5 and $\epsilon_v = 1e-15$.

In Figures 7.34, 7.35, 7.36 and 7.37 can be seen the comparison between simulated hydro-graphs and experimental data. The numerical simulation was computed by HLL scheme with modification (2.107). As expected, plain numerical scheme is more viscous than the scheme with the linear reconstruction (2.5).

The steep drop of the water level at the point A which is at the beginning of the simulation is

Figure 7.34: Sensor A($x=1.0$, $y=1.0$)Figure 7.35: Sensor B($x=0.18$, $y=1.0$)Figure 7.36: Sensor C($x=0.48$, $y=0.4$)Figure 7.37: Sensor D($x=1.802$, $y=1.45$)

caused by the fact that during the very early time period of the experiment and simulation, the water flow has the character of 2D dam break. As mentioned in section 7.1.1 the water depth at the dam site in 2D case is constant and equal to $\frac{4}{9}h_0$. In case of $h_0 = 0.6$ m, this constant depth is $0.2\overline{66}$ m. But before the water level drops to this value the flow becomes 3D and perpendicular flows cause the surge of the water level. There are some discrepancies between the numerical results and the experiment after the surge of the water level. G. Gottardi and M. Venutelli in their work [112] believe that these discrepancies occurring at the beginning of the simulation in Figure 7.34 are not due to numerical approximation but due to the mathematical model of Shallow Water Equations which isn't suitable to solve the type of the fluid flow which occurs at the dam gate position immediately after it's removing. In other points B and C can be seen very good agreement between the simulation and the experiment. The agreement is even better at the D point which is further from the water gate and simulated water depth is more shallow here.

In Figures 7.38 and 7.39 the comparison of the computations with different finite volume cell width is shown. From this plot, it can be seen that the solution is converging and with smaller cell width the numerical viscosity of the scheme decreases.

Table 7.11 shows L_1 error of the water depth. This error was computed as

$$L_1 \text{error} = \frac{1}{n_t} \sum_{i=1}^{n_m} \frac{|h_{ex}^i - h_{num}^i|}{\max(h_{ex}^i, 1e-10)} \quad (7.18)$$

where h_{ex}^i is the experimental, h_{num}^i is the numerical value of the water level in corresponding time

and n_m is the number of the measured times. In Figures 7.38, 7.39 and table 7.11 the comparison of basic scheme without the linear reconstruction is shown.

The values in table 7.11 have monotonic character and in general the smaller cell width yields better results. Higher mesh resolution is needed at the place of the dam site, where the water level attains high gradients. To simulate the water level more precisely, the cell width should be smaller than 0.01 m. On the other hand the computations give good agreement with the experimental data at the point D even for lower mesh resolution and cell width of 0.05 m suffices.

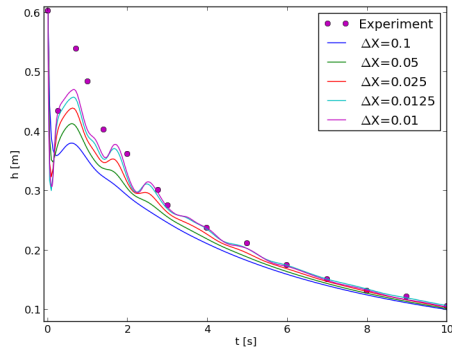


Figure 7.38: Comparison of the computations with different finite volume cell width at the point A.

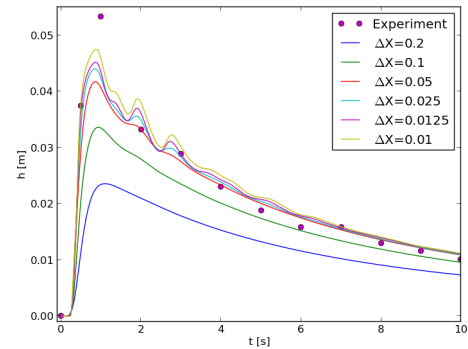


Figure 7.39: Comparison of the computations with different finite volume cell width at the point D.

Table 7.11: L_1 Error of the Water Level

$\Delta x = \Delta y$	A	B	C	D
0.2	0.163457	0.132316	0.125966	0.340554
0.1	0.131234	0.130057	0.112161	0.142759
0.05	0.102865	0.098472	0.07931	0.073187
0.04	0.09319	0.0908	0.071626	0.070216
0.025	0.073526	0.078179	0.058245	0.068996
0.02	0.064471	0.073454	0.055909	0.066053
0.013	0.05122	0.070692	0.054282	0.070499
0.01	0.044427	0.039532	0.03719	0.082341

7.2.3 Obstacle Experiment

The last experiment validates the invented 2D discontinuous Galerkin solver. Kleefsman et al. [66] conducted another dam break experiment. In contrast to the previous experiment, the bed slope source term has to be considered in this case. For the experiment large tank with the size $3.22 \times 1 \times 1$ m was used. At the position shown in Figure 7.40 and Figure 7.41, rectangular obstacle was placed. The right part of the tank was closed by the water gate. The initial condition of the left part was a dry bed and behind the gate, there was 0.55m of water. The gate was released by a weight which almost instantaneously pulls the door up. The experiment was designed for the validation of the simulations

conducted by volume of fluid method and fully 3D model of flows, but the capabilities of discontinuous Galerkin method and Shallow Water Equations to solve this problem, are discussed in this section.

During the experiment the measurements of the water heights at four places were performed. The positions of the sensors are shown in Figures 7.40 and 7.41. The Manning's bed roughness factor was set to 0.01 in whole computational domain, CFL was set to 0.5 and simulation was performed by first, second and third order DGFEM scheme.

In Figures 7.42 and 7.43, there are shown the simulations with different meshes compared with measurements in sensor H2. The first coarse mesh made of 640 finite elements is shown in Figure 7.42 and in Figure 7.43 the results of relatively fine mesh created by 256000 finite elements is plotted. The comparison of the DGFEM with different accuracy on both meshes yields that the higher orders DGFEM does not bring remarkable performance. This is because of the shock appearance. As discussed in Section 7.1.1.2, the accuracy of the schemes is decreased in case of discontinuous solutions.

There are the plots of the measured water depth and water depth simulated by DGFEM-3 in Figures 7.44, 7.45, 7.46 and 7.47. Discrepancies between the numerical simulation and sensor H1 in Figure 7.44 are, accordingly to the author's opinion, caused by the measuring method used in experiment and used mathematical model. From the video record and Figure 18 in the original article [66], it can be seen that in time 0.56s there is some splashing water over the position of the sensor H1. The sensor probably wasn't able to catch this water and the mathematical model of the Shallow Water Equations is not able to solve a splashing liquid. Nevertheless, the numerical results have very good agreement with the experiment.

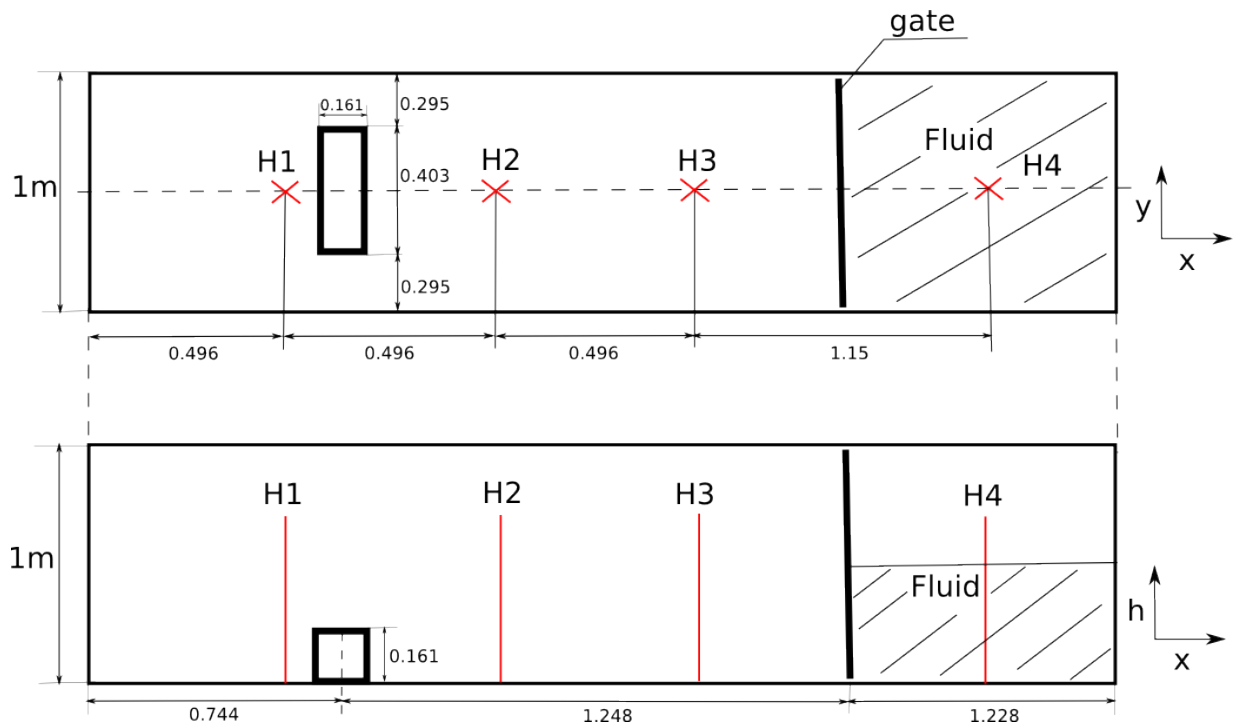


Figure 7.40: The experimental setup.

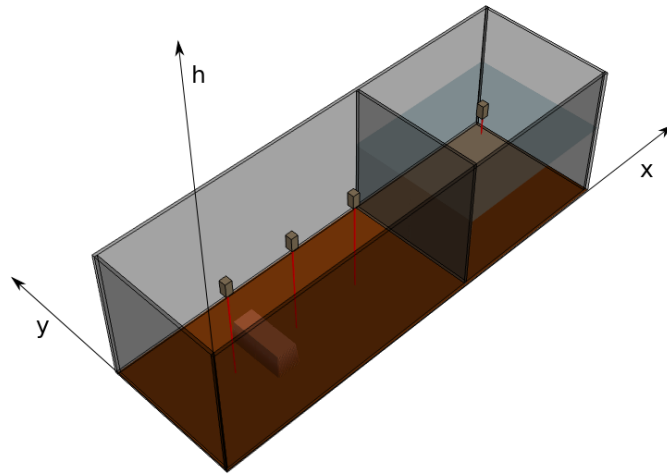


Figure 7.41: Visualisation of the initial conditions.

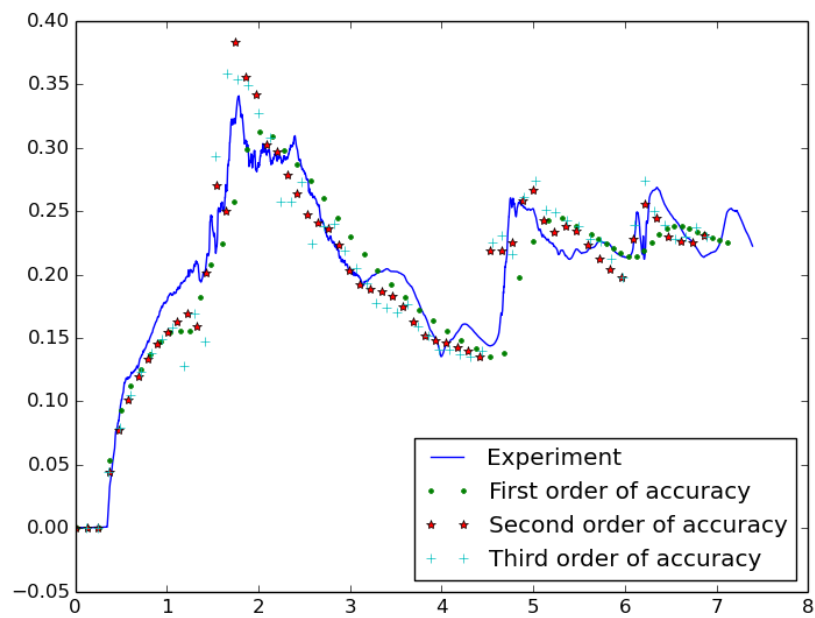


Figure 7.42: Comparison of the experimental data measured by sensor H2 and the numerical simulation simulated on the mesh created by 640 finite elements.

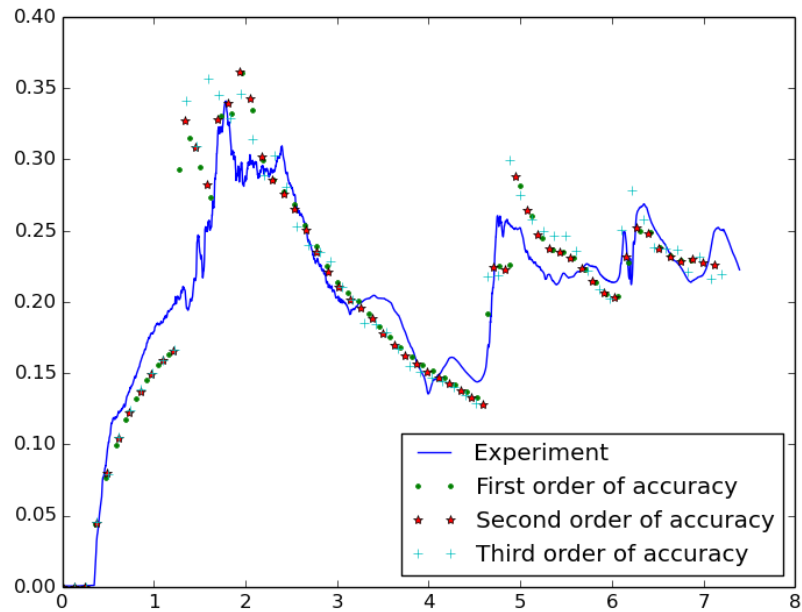


Figure 7.43: Comparison of the experimental data measured by sensor H2 and the numerical simulation simulated on the mesh created by 256000 finite elements.

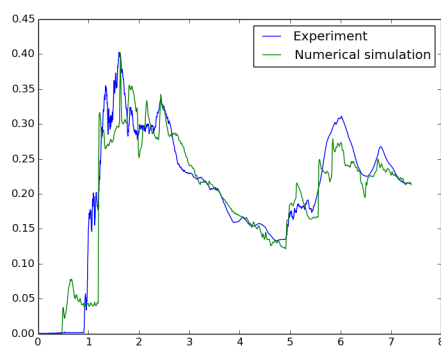


Figure 7.44: Comparison of the numerical simulation and the experimental data measured by sensor H1.

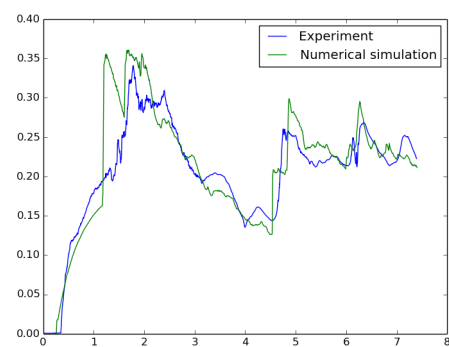


Figure 7.45: Comparison of the numerical simulation and the experimental data measured by sensor H2.

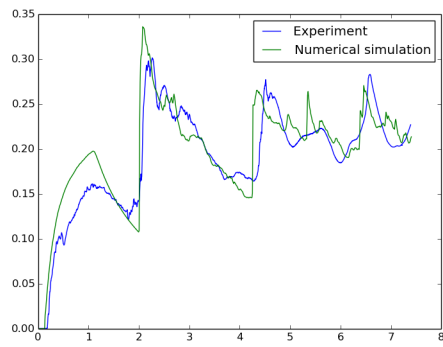


Figure 7.46: Comparison of the numerical simulation and the experimental data measured by sensor H3.

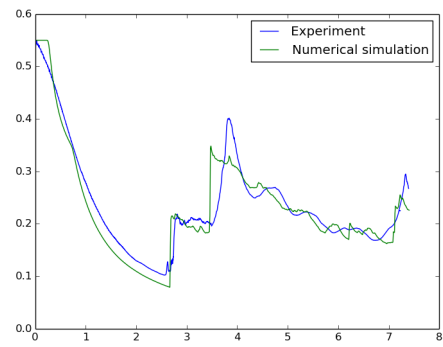


Figure 7.47: Comparison of the numerical simulation and the experimental data measured by sensor H4.

Conclusion

In this work the system of Shallow Water Equations also known as Saint-Venant's equations is described. In the mathematical model the bed slope source term and the bed friction source term were involved. This mathematical model is suitable for simulations of water flows when the vertical fluxes are negligible and horizontal scales of water mass are much bigger than vertical scales.

Basic mathematical features and eigenvalues of Saint-Venant's equations were described in Section 1.1. These eigenvalues are important for understanding the principle of numerical schemes and for the boundary conditions settings. Mathematical model was discretised by finite volume method (Chapter 2) and discontinuous Galerkin method (Chapter 3).

In Chapter 2.2, there are two possible schemes for the numerical flux approximation described. First scheme is Roe scheme which is based on replacement of the original non-linear system of differential equations by new linearised system with constant coefficients. Second scheme is HLL scheme which is based on integration over Riemann's fans. Both schemes are based on up-wind principle which causes problems within the simulations of the flows where Froude number is equal to one. This situation causes that one of the eigenvalues is zero and scheme suffers from the lack of the numerical viscosity. Meanwhile the error of HLL scheme is relatively small, the behaviour of the Roe scheme is non-physical and needs the re-computation of the eigenvalues described in Section 2.2.3. Both schemes give relatively the same results as shown in Figure 7.4, but for the numerical flux approximation, HLL scheme was chosen because this scheme does not need any additional constant ϵ to correct the eigenvalues.

Numerical approximation of the bed source term for the FVM is described in Chapter 2.3. This chapter was focused on the bed source approximations which provide 'C-property' condition. The proofs of 'C-property' condition for Roe and HLL schemes in flooded domain are also described. The cell-centred bed approximation, described in this work, keeps this condition also at the wet/dry interface for the Roe scheme. To use this cell-centred bed approximation also within the HLL scheme, novel computation of the eigenvalues (2.107) was introduced in this work. The approach with the bed function approximated at the edges of finite volumes brings problems with negative water level at the wet/dry interface as described in Section 2.3.2.1. Solution of this problem proposed Alexander Kurganov in [41]. But this modification of water depth reconstruction is in conflict with 'C-property' condition. So in frame of present work, a novel method of computation of the bed slope source term (2.123) was invented. Computations do not crash because of the negative value and also 'C-property' condition is fully fulfilled. However this novel treatment is not mass conservative. The mass conservative scheme was obtained by modification (2.127) which was also invented in frame of this work.

Numerical simulations at the wet/dry interface needs special treatment. Not only because of the satisfaction of the 'C-property' condition but also because of the computations of the flow velocity. These velocities can, due to the computer inaccuracy, take non-physically large values when water depth

is approaching zero value. In Section 2.2.4 three methods of handling this problem were described. All methods were investigated in Section 7.1.1.3 and numerical simulations show that the optimal choice is to set a small constant ϵ and for practical problems water depth smaller than this constant consider to be zero. This method also decreases the computational demands. Second suitable approach is setting zero velocity at the wet/dry interface. This approach also prevents the occurrence of negative water depth when computational domain is drying and the cell-centred bed approximation is used. Unfortunately in case of edge based bed function approximation, this method cannot guarantee non-negative water depth. Modification with ϵ_v distorts the front face of the surge wave. Distortion is bigger with larger ϵ_v . The non-physical behaviour of the flow at the wet/dry interface was proved in Section 7.1.1.3 and Section 7.1.3.

The bed friction source term is described in Section 2.4. The approach published in [31] was used. This bed source term wasn't tested independently, but the bed friction was implemented in all simulations in Section 7.2.

The order of accuracy of the FVM can be improved by the linear reconstruction of variables as described in Section 2.5. To keep total variation diminishing property of proposed schemes minmod limiter was introduced.

Another way of achieving higher order of spatial accuracy is usage of the discontinuous Galerkin method. This method is described in Chapter 3. The high order of approximation has to be limited around the shocks to avoid a spurious oscillations. This can be done by artificial viscosity or by slope limiters. Both methods were described and tested within this work. In case of appearance of the wet/dry interface the slope limiters have to be used, as the artificial viscosity cannot directly affect the solution of the water depth and avoid appearance of the negative water depth. The combination of slope limiters, used for the limiting of the water depth, and artificial viscosity, used for dumping of the discharge, was also tested, but gave bad results. So the combination of artificial viscosity and slope limiters is not recommended.

As discussed in Section 3.2.2, it is not clear when the solution should be limited. Many authors use some constants to define 'troubled' cells. But there is uncertainty in the magnitude of this constants as discussed in Section 3.2.2.2. Thus within this work novel 'constantless' criterion (3.2.2.3) for defining 'troubled' cell was introduced. This criterion is focused on the shape of the solution in the finite element, unlike the other schemes which are focused on the monotonicity or discontinuity of the solution. The method was presented in author's publication [118]. The novel limiting method of the DGFEM was extended to 2D space (3.3.2). The usage of the orthogonal functions in the bed function approximation needs special computation of the bed function coefficients. Within this work, the novel evaluation of these coefficients (3.76) was introduced. This computations ensures the continuity of the bed topography function.

The bed friction source term is handled in the similar way as within the finite volume method. The bed friction for 1D solution is described in Section 3.2.4 and bed friction for the 2D solver is described in Section 3.3.4.

DGFEM can improve formal order of accuracy. But in a fact the improvement of accuracy can be seen only in computations without any shocks or discontinuities in the solution. In Table 7.4 it can be noticed that DGFEM can substitute large number of finite elements by the high order basis functions. The method gives the same accurate results for 25 of finite elements as well as for 100 of the finite elements when high degree of basis polynomials is used. However, the accuracy of DGFEM method decreases when the solution contains discontinuities or shocks. This can be seen in Tables 7.5 and

7.6 or in comparison with the experiment described in Section 7.2.2. The advantage of the DGFEM method is the fact that the bed function can be described by arbitrary polynomials, meanwhile within the FVM the bed function can be described only by linear functions. That is why the second order accurate DGFEM gives much better results than the second order accurate FVM in Tables 7.4, 7.5 and 7.6. But DGFEM brings some improvement of accuracy also when the bed function is described by the linear function. This comparison of the second order DGFEM and the second order FVM for bed function described by linear functions is shown in Table 7.7. If the bed function is not described by polynomials, than DGFEM cannot achieve a high performance and computational demands are to high as shown in Table 7.8.

Methods of the numerical time integration (Euler's method, the second and the fourth order Runge-Kutta methods) were described in Chapter 4. Euler method isn't suitable for the numerical flux approximation schemes of higher order accuracy. The comparison of the numerical and analytical solutions yields that the second order Runge-Kutta suffices for numerical simulations and fourth order Runge-Kutta doesn't bring remarkable improvement for the Shallow Water Equation as shown in Table 7.3. Moreover, fourth order Runge-Kutta is computationally more expensive.

The work brings new findings in simulation of the shallow water flow. The main achievement was made in the simulation at the wet/dry interface. Two new approximations of the bed source term, providing C-property condition at the wet/dry interface, for the HLL scheme were introduced. In Section 2.3.2 the approximation for the bed function evaluated at the edges of the finite volumes was introduced and in Section 2.3.1.2 a scheme for the cell-centred approximation of the bed function was described. For the bed approximation at the edges of the finite volumes, novel numerical scheme (2.123) for computation of the wet/dry interface was invented. This scheme keeps the 'C-property' condition not only at the fully flooded computational domain but also in the case when the finite volume cell contains wet/dry interface. Moreover, the scheme was extended by novel predictor (2.127), which ensures the mass conservativity of the scheme.

In Section 2.2.4, there were described different approaches of computations of the flow velocity. After numerical simulations and tests the approach with ϵ was chosen. This method does not distort the front face of the surge and computationally is undemanding. In case of using approach with ϵ_v , the using of values smaller than $1e-10$ was suggested. The comparison of all three methods is discussed in Section 7.1.1.3.

Future work can be focused on improving, future validation and practical usage of the solver which was written as a part of this work. One of practical usage can be defining flood areas or computations of the impact of the anti-flood barriers or during river catchment restorations. Mathematical model can be extended by porous flow source term. Porous flows are useful for flow simulations in urban areas where the scale of the mesh or missing data do not allow to describe some conglomeration of the buildings by the mesh.

Main Findings of the Work

Present work describes a numerical solution of the shallow water equations. The mathematical model was solved by the finite volume method and by the discontinuous Galerkin method. For both methods, the unstructured meshes were used. At the edges of the mesh cells the numerical flux is computed.

Roe and HLL approximate Riemann solvers for numerical flux computation were tested in Section 7.1.1.2. Both are giving approximately the same results which confirm the findings in [114]. HLL

scheme is in author's opinion preferable because Roe scheme needs Harten's correction (2.2.3) when Froude number takes on zero value. HLL scheme gives good results also without this correction, however the correction brings better results as shown in Figure 2.6.

Special treatment is needed when the water depth approaches zero value. This can happen mainly during flooding or drying processes. Three of these treatments are described in Section 2.2.4. All methods are tested in Section 7.1.1.3. It is concluded that the best solution gives the treatment when the water depth smaller than a chosen threshold is considered to be zero. This was also discussed in author's publication [116].

As stated in the introduction, the work is focused on the schemes which satisfy C-property condition. The scheme published in [41] is ensuring non-negativity of the water depth, but does not keep the C-property condition at the wet/dry interface. Within this work, a novel scheme for computation of the wet/dry interface was suggested. The scheme is fully mass conservative thanks to the modification (2.127) and the C-property condition at the wet/dry interface is kept thanks to the modification (2.123). These modifications also improve the behaviour of the numerical scheme as shown in Figures 7.26 and 7.25. More test cases and detailed description can be found in author's publication [119].

In Section 3.2.2.3 a novel limiting process for DGFEM was developed. This process is thoroughly described in author's publication [117]. The advantages of this limiting scheme were shown in Table 7.2 where the novel limiting scheme surpasses the global limiting scheme. More test cases and validations can be seen in [117].

Let us sum up the applicability of FVM and DGFEM. Theoretically DGFEM gives arbitrarily high order of accuracy. However, the order of accuracy decreases rapidly when some shock is present in the computational domain. This was discussed in Section 7.1.2 (see Tables 7.5 and 7.6). Figures 7.42 and 7.43 also confirm the fact that there is no significant improvement of the accuracy with higher order of DGFEM when the simulated problem contains a shock. Moreover DGFEM is computationally very demanding method in comparison with FVM (see Table 7.8). From this point of view, FVM is more favourable method. On the other hand, as discussed in Section 7.1.2, DGFEM method gives very accurate solution and has low computational demands for coarse meshes if the bed function is described by higher polynomials (see Table 7.4 and Note 4). Thus the suggestion for practical applications of DGFEM are following: fine mesh should be replaced by the coarse one to decrease computational demands and the bed function should be described by the piece-wise polynomial approximation to sustain the quality of the numerical results.

Bibliography

- [1] Barré de Saint Venant. Theorie du mouvement non permanent des eaux. *Théorie du mouvement non permanent des eaux, avec application aux crues des rivières et à l'introduction des marées dans leur lit*, 73:147–154, 1871.
- [2] R.F. Dressler. Hydraulic Resistance Effect Upon the Dam-Break Functions. *Journal of Research of the National Bureau of Standards*, 49, 1952.
- [3] R.F. Dressler. Comparison of Theories and Experiments for the Hydraulic Dam-Break Wave. *International Association of Scientific Hydrology*, 38:319–328, 1954.
- [4] J.J. Stoker. *Water waves*. Interscience Publishers, Wiley, New York, 1957.
- [5] O. Delestre, C. Lukas, P.A. Ksinant, F. Darboux, C. Laguerre, T.N.T. Vo, F. James, and S. Cordier. SWASHES: a compilation of shallow water analytic solutions for hydraulic and environmental studies. *International Journal for Numerical Methods in Fluids*, 72:269–300, June 2013.
- [6] E.F. Toro. *Shock-Capturing Methods for Shallow Water Flows*. John Wiley & Sons, LTD, 2001.
- [7] P.D. Bates, S.N. Lane, and Ferguson R.I. *Computational Fluid Dynamics: Applications in Environmental Hydraulics*. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England, 2005.
- [8] K.F. Cheung and Wu Y. Explicit solution to the exact Riemann problem and application in non-linear shallow-water equations. *International Journal for Numerical Methods in Fluids*, 57:1649–1668, 2008.
- [9] K. Hadašová. Aplikace Riemannových řešičů na dynamiku tekutin. Bachelor thesis, Katedra matematiky, Fakulta aplikovaných věd, Západočeská Univerzita v Plzni, Czech republic, 2008.
- [10] P.L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.
- [11] A. Harten, P. Lax, and B. Leer. On Upstream Differencing and Godunov–Type Schemes for Hyperbolic Conservation Laws. *Society for Industrial and Applied Mathematics*, 25:35–61, March 1983.
- [12] E.F. Toro. *Riemann Solvers and Numerical Methods of Fluid Dynamic*. Springer Dordrecht Heidelberg, London, New York, 2009.

- [13] A.I. Delis, M. Kazolea, and Kampanis N.A. A robust high-resolution finite volume scheme for the simulation of long waves over complex domains. *International Journal for Numerical Methods in Fluids*, 56:419–452, 2008.
- [14] K. Anastasiou and C. T. Chan. Solution of the 2D Shallow Water Equations Using the Finite Volume Method on Unstructured Triangular Meshes. *International Journal for Numerical Methods in Fluids*, 24:1225–1245, 1997.
- [15] A Bermúdez, A. Dervieux, J. A. Desiderib, and M. E. Vazqueza. Upwind Schemes for the Two-dimensional Shallow Water Equations with Variable Depth Using Unstructured Meshes. *Computer Methods in Applied Mechanics and Engineering*, 155:49–72, 1998.
- [16] S. F. Bradford and B. F. Sanders. Finite-Volume Model for Shallow-Water Flooding of Arbitrary Topography. *J. Hydraul. Eng.*, 128:289–298, 2002.
- [17] Z. Wang and Y. Geng. Two-dimensional shallow water equations with porosity and their numerical scheme on unstructured grids. *Water Science and Engineering*, 6:91–105, 2013.
- [18] A. Kurganov, S. Noelle, and G. Petrova. Semi-discrete Central-Upwind Schemes For Hyperbolic Conservation Laws And Hamilton-Jacobi Equations. *Society for Industrial and Applied Mathematics*, 23:707–740, 2001.
- [19] A. Kurganov and G. Petrova. Central-upwind schemes on triangular grids for hyperbolic systems of conservation laws. *Numerical Methods for Partial Differential Equations*, 21:536–552, 2005.
- [20] E. Audusse, F. Bouchut, M.O. Bristeau, R. Klein, and B. Perthame. A Fast and Stable Well-Balanced Scheme with Hydrostatic Reconstruction for Shallow Water Flows. *SIAM Journal on Scientific Computing*, 25(6):2050–2065, January 2004.
- [21] J. Hou, Q. Liang, F. Simons, and R. Hinkelmann. A 2D well-balanced shallow flow model for unstructured grids with novel slope source term treatment. *Advances in Water Resources*, 52:107–131, February 2013.
- [22] D. L. George. Numerical Approximation of the Nonlinear Shallow Water Equations with Topography and Dry Beds: A Godunov-Type Scheme. A thesis submitted in partial fulfillment of the requirements for the degree of master of science, University of Washington, USA, 2004.
- [23] A. Kurganov and D. Levy. Central-upwind Schemes for the Saint-Venant System. *ESAIM: M2AN*, 36:397–425, 2002.
- [24] T.C. Rebollo, A.D. Delgado, and E. Fernandez. An entropy-correction free solver for non-homogeneous shallow water equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37:755–772, 2003.
- [25] J. Hou, F. Simons, and R. Hinkelmann. A new TVD method for advection simulation on 2D unstructured grids. *International Journal for Numerical Methods in Fluids*, 71(10):1260–1281, April 2013.
- [26] G. Kesserwani. Topography discretization techniques for godunov-type shallow water numerical models: a comparative study. *Journal of Hydraulic Research*, 51:351–367, 2013.

- [27] V.T Chow. *Open-channel Hydraulics*. McGraw-Hill Book Company, ISBN 07-010776-9, Oxford, UK, 1959.
- [28] S.N. Lane. Hydraulic modelling in hydrology and geomorphology: A review of high resolution approaches. *Hydrological Processes*, 12:1131–1150, 1998.
- [29] D.B. Ingham and L. Ma. *Fundamental equations for CFD in River Flow Simulations*. John Wiley & Sons, Ltd. ISBN: 0-470-843-59-4(HB), 2006.
- [30] W. Rodi. A new algebraic relation for calculating the Reynolds stresses. *Zeitschrift faddotur Angewandte Mathematik und Mechanik*, 56:219–222, 1976.
- [31] Q. Liang and F. Marche. Numerical resolution of well-balanced shallow water equations with complex source terms. *Advances in Water Resources*, 32(6):873–884, June 2009.
- [32] V. Guinot and S. Soares-Frazão. Flux and source term discretization in two-dimensional shallow water models with porosity on unstructured grids. *International Journal for Numerical Methods in Fluids*, 50(3):309–345, January 2006.
- [33] Andreas Bollermann, Guoxian Chen, Alexander Kurganov, and Sebastian Noelle. A well-balanced reconstruction of Wet/Dry fronts for the shallow water equations. *Journal of Scientific Computing*, 56:267–290, 2013.
- [34] M. Fišer, O. Bublík, L. Lobovský, and J. Vimmr. Problems and Solutions Connected with the Wet/Dry Interface in the Mathematical Model of the Shallow Water Equations. In *30th Conference with International Participation Computational Mechanics 2014*, pages 27–28, Špičák, 2014.
- [35] Emmanuel Audusse and Marie-Odile Bristeau. A well-balanced positivity preserving “second-order” scheme for shallow water flows on unstructured meshes. *Journal of Computational Physics*, 206(1):311–333, June 2005.
- [36] S. Bunya, E.J. Kubatko, J.J. Westerink, and C. Dawson. A wetting and drying treatment for the Runge–Kutta discontinuous Galerkin solution to the shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 198(17-20):1548–1562, April 2009.
- [37] Vincenzo Casulli. A high-resolution wetting and drying algorithm for free-surface hydrodynamics. *International Journal for Numerical Methods in Fluids*, 60(4):391–408, June 2009.
- [38] J. Hou, Q. Liang, F. Simons, and R. Hinkelmann. A stable 2D unstructured shallow flow model for simulations of wetting and drying over rough terrains. *Computers & Fluids*, 82:132–147, August 2013.
- [39] J. Hou, F. Simons, Q. Liang, and R. Hinkelmann. An improved hydrostatic reconstruction method for shallow water model. *Journal of Hydraulic Research*, 52(June):1–8, May 2014.
- [40] Stephen C Medeiros and Scott C Hagen. Review of wetting and drying algorithms for numerical tidal flow models. *International Journal for Numerical Methods in Fluids*, 71(4):473–487, February 2013.

- [41] A. Kurganov and G. Petrova. A second-order well-balanced positivity preserving central-upwind scheme for the Saint-Venant system. *Communications in Mathematical Sciences*, 5(1):133–160, 2007.
- [42] P. Sivakumar, D.G. Hyams, L.K. Taylor, and W.R. Briley. A primitive-variable Riemann method for solution of the shallow water equations with wetting and drying. *Journal of Computational Physics*, 228(19):7452–7472, October 2009.
- [43] O. Delestre, S. Cordier, F. Darboux, and J. François. A limitation of the hydrostatic reconstruction technique for Shallow Water equations. *Comptes Rendus Mathématique*, 350(13-14):677–681, July 2012.
- [44] P. Colella and P. R. Woodward. The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations. *Journal of Computational Physics*, 54:174–201, September 1984.
- [45] Ami Harten, Bjorn Engquist, Stanley Osher, and Sukumar R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, iii. *Journal of Computational Physics*, 131(1):3 – 47, 1997.
- [46] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted eno schemes. *Journal of Computational Physics*, 126(1):202 – 228, 1996.
- [47] Xu-Dong Liu, Stanley Osher, and Tony Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200 – 212, 1994.
- [48] Dinshaw S. Balsara and Chi-Wang Shu. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *Journal of Computational Physics*, 160(2):405 – 452, 2000.
- [49] W. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. *Technical Report LA-UR-73-479, Los Alamos, NM, Los Alamos Scientific Laboratory*, 1973.
- [50] Q. Araud, P. Finaud-Guyot, V. Guinot, Mosé R., and J. Vazquez. An eigenvector-based linear reconstruction approach for time stepping in discontinuous galerkin scheme used to solve shallow water equations. *Int. J. Numer. Meth. Fluids*, 70:1590–1604, 2012.
- [51] L. Krivodonova, J. Xin, J.F. Remacle, N. Chevaugeon, and J.E. Flaherty. Shock detection and limiting with discontinuous galerkin methods for hyperbolic conservation laws. *Applied Numerical Mathematics*, 48:323–338, 2004.
- [52] Kesserwani G. and Liang Q. A conservative high-order discontinuous galerkin method for the shallow water equations with arbitrary topography. *International Journal for Numerical Methods in Engineering*, 86:47–69, 2011.
- [53] C. Eskilsson and S.J. Sherwin. A triangular spectral/hp discontinuous galerkin method for modelling 2d shallow water equations. *Int. J. Numer. Meth. Fluids*, 45:605–623, 2004.
- [54] D. Kuzmin. Slope limiting for discontinuous Galerkin approximations with a possibly non-orthogonal Taylor basis. *Int. J. Numer. Meth. Fluids*, (12):1–12, April 2011.
- [55] Liang Q. Kesserwani, G. A discontinuous galerkin algorithm for the two-dimensional shallow water equations. *Comput. Methods Appl. Mech. Engrg.*, 199:3356–3368, 2010.

- [56] A. Ern, S. Piperno, and K. Djadel. A well-balanced runge–kutta discontinuous galerkin method for the shallow-water equations with flooding and drying. *Int. J. Numer. Meth. Fluids*, 58:1–25, 2008.
- [57] Y. Xing. Exactly well-balanced discontinuous galerkin methods for the shallow water equations with moving water equilibrium. *Journal of Computational Physics*, 257:536–553, 2014.
- [58] Y. Xing and Ch.-W. Shu. High order well-balanced finite volume weno schemes and discontinuous galerkin methods for a class of hyperbolic systems with source terms. *Journal of Computational Physics*, 214:567–598, 2006.
- [59] Y. Xing, X. Zhang, and Shu Ch.-W. Positivity-preserving high order well-balanced discontinuous galerkin methods for the shallow water equations. *Advances in Water Resources*, 33:1476–1493, 2010.
- [60] O. Bublík, J. Vimmr, A. Pecka, and A. Jonášová. Computational mechanics 2015. 31st conference with international participation. In *A parallel implementation of the implicit discontinuous Galerkin finite element method for fluid flow problems*, number 31, Hotel Horizont, Špičák, Czech Republic, Nov 2015.
- [61] J.C. Martin and W.J. Moyce. Part IV. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 244:312–324, 1952.
- [62] P.K. Stansby, A. Chegini, and T.C.D. Barnes. The initial stages of dam-break flow. *Journal of Fluid Mechanics*, 347:407–424, 1998.
- [63] L.S. Nanía, M. Gómez, and J. Dolz. Experimental study of the dividing flow in steep street crossings. *J. Hydr. Res.*, 42:406–412, 2004.
- [64] S. Soares-Frazão. Experiments of dam-break wave over a triangular bottom sill. *Journal of Hydraulic Research*, 45:19–26, 2007.
- [65] S. Soares-Frazão and Y. Zech. Experimental study of dam-break flow against an isolated obstacle. *Journal of Hydraulic Research*, 45(sup1):27–36, December 2007.
- [66] K.M.T. Kleefsman, G. Fekken, A.E.P. Veldman, B. Iwanowski, and B. Buchner. A volume-of-fluid based simulation method for wave impact problems. *Journal of Computational Physics*, 206:363–393, 2005.
- [67] V. Bukreev and V. Zykov. Bore impact on a vertical plate. *Journal of Applied Mechanics and Technical Physics*, 49:926–933, 2008.
- [68] L. Lobovský, E. Botia-Vera, F. Castellana, J. Mas-Soler, and A. Souto-Iglesias. Experimental investigation of dynamic pressure loads during dam break. *Journal of Fluids and Structures*, 48:407–434, 2014.
- [69] S. Soares-Frazão and Y. Zech. Dam-break flow through an idealised city, *Journal of Hydraulic Research*. 45:648–658, 2008.

- [70] F. Alcrudo, P. Brufau P. Garcia, J. Murillo, D. Garcia, and J. Mulet. *The Model City Flooding Experiment*. Study case, Area de Mecánica de Fluidos, CPS-Universidad de Zaragoza, Spain, 2008.
- [71] Ch. Ziveri. *2D Finite Volume Numerical Modeling of Free Surface Flows with Topography*. Doctoral thesis, Università degli Studi di Parma, Faculty of Engineering, 2008.
- [72] E. Baluchová. Greenova věta – její formulace a aplikace. Bachelor thesis, Přírodovědecká fakulta, Univerzita palackého v Olomouci, Czech republic, 2009.
- [73] P.L. Roe. Proceedings. *Seventh Int. Conf. Num. Meth. Fluid Dyn.*, Springer-Verlag, New York/Berlin, 1981.
- [74] Bublík et al. Metody numerického řešení základních modelů proudění tekutin a jejich optimalizace, Modelové skalární PDR: Zobecnění upwind schématu. Frvs 958/2009/g1, Katedra mechaniky, Faculty of Applied Sciences, University of West Bohemia, Pilsen, Czech republic, 2009.
- [75] P. L. Roe. Characteristic based schemes for the euler equations. *Ann. Rev. Fluid Mech.*, pages 337–365, 1986.
- [76] E. Godlewski and P. A. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer-Verlag, 1996.
- [77] A. Kurganov and G. Petrova. A Central-Upwind Scheme for Nonlinear Water Waves Generated by Submarine Landslides. *Research supported by NSF Grants DMS-0310585 and DMS-0610430*.
- [78] A. Bermúdez and M. E. Vazques. Upwind Methods for Hyperbolic Conservations Laws with Source Terms. *Computers Fluids*, 23:1049–1071, 1994.
- [79] Luigi Fraccarollo and Eleuterio F. Toro. Experimental and numerical assessment of the shallow water model for two-dimensional dam-break type problems. *Journal of Hydraulic Research*, 33(6):843–864, November 1995.
- [80] N. Dodd. Numerical model of wave run-up, overtopping, and regeneration. *Coastal Engineering*, 47:1–26, 2002.
- [81] J.G. Zhou, D.M. Causon, C.G. Mingham, and D.M. Ingram. The Surface Gradient Method for the Treatment of Source Terms in Shallow-Water Equations. *Journal of Computational Physics*, 168:1–25, 2001.
- [82] www.cfd online. http://www.cfd-online.com/wiki/tvd_scheme.
- [83] A. Kurganov and E. Tadmor. New High-Resolution Central Schemes for Nonlinear Conservation Laws and Convection-Diffusion equations. *J. Comput. Phys.*, 160:241–282, 2000.
- [84] H. Nessyahu and E. Tadmor. Non-oscillatory Central Differencing for Hyperbolic Conservation Laws. *Journal of Computational Physics*, 87:408–463, 1990.
- [85] J. Qiu and C.W. Shu. A comparison of troubled-cell indicators for runge–kutta discontinuous galerkin methods using weighted essentially nonoscillatory limiters. *SIAM Journal on Scientific Computing*, 27:995–1013, 2005.

- [86] O. Bublík. Aplikace nespojité Galerkinovy metody konečných prvků na řešení úloh mechaniky tekutin. Thesis, Department of Mechanics, Faculty of Applied Sciences, University of West Bohemia, Pilsen, Czech republic, 2014.
- [87] M. Yang and Z.J. Wang. A parameter-free generalized moment limiter for high-order methods on unstructured grids. *Advances in Applied Mathematics and Mechanics*, 1:451–480, 2009.
- [88] S. Adjerid, K. Devine, J. Flaherty, and Krivodonova L. A posteriori error estimation for discontinuous galerkin solutions of hyperbolic problems. *Comput. Methods Appl. Mech. Engrg*, 191:1097–1112, 2002.
- [89] B. Cockburn and C.W. Shu. Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws ii: General framework. *Math. Comp.*, 52:411–435, 1989.
- [90] Jeremy Dewar, Alexander Kurganov, and Maren Leopold. Pressure-based adaption indicator for compressible euler equations. *Numerical Methods for Partial Differential Equations*, 31(6):1844–1874, 2015.
- [91] R. Biswas, Karen D.D., and J.E. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Applied Numerical Mathematics*, 14:255–283, 1994.
- [92] J. Česenek, M. Feistauer, and A. Kos. Dgfem for the analysis of airfoil vibrations induced by compressible flow. *ZAMM - Journal of Applied Mathematics and Mechanics*, 93:387–402, 2013.
- [93] O. Bublík and A. Vimmr, J.and Jonášová. A local time step discontinuous galerkin finite element method for the solution of euler equations on unstructured meshes. *27th conference with international participation Computational Mechanics 2011*, pages 1–2, 2011.
- [94] K. Rektorys et al. *Přehled užité matematiky*. Nakladatelství technické literatury, Praha, 1988.
- [95] www.en.wikipedia.org. http://en.wikipedia.org/wiki/runge%e2%80%93kutta_methods.
- [96] D.D. Knight. *Elements of Numerical Methods for Compressible Flows*. Cambridge University Press, ISBN: 0-52155474-8, UK, 2006.
- [97] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzgleichungen der mathematischen Physik. *Mathematische Annalen*, 100:32–74, 1928.
- [98] R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, ISBN:9780521009249, UK, 2002.
- [99] O. Bublík. Písemna práce ke státní doktorské zkoušce v oboru Aplikovaná mechanika. Thesis, Katedra mechaniky, Faculty of Applied Sciences, University of West Bohemia, Pilsen, Czech republic, 2010.
- [100] T.G. Mattson, B.A. Sanders, and B.L. Massingill. *Patterns for Parallel Programming*. Addison-Wesley Professional, ISBN: 10: 0-321-22811-1, Oxford, UK, 2004.
- [101] M. Quinn. *Parallel Programming in C with MPI and OpenMP*. McGraw-Hill Science, ISBN 10: 0072822562, Richmond, US, 2003.
- [102] R. Chandra, L. Dagum, D. Kohr, D. Maydas, J. McDonald, and R. Menon. *Parallel Programming in OpenMP*. Morgan Kaufmann, ISBN 9781558606715, 2000.

-
- [103] B. Chapman, G. Jost, and R. Pas. *Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation)*. MIT Press, ISBN-10:0262533022, Massachusetts, US, 2008.
- [104] A. Ritter. Die Fortpflanzung der Wasserwellen. *Z. Ver. deut. Ing*, 36:–, 1892.
- [105] A. Schoklitsch. Über Dambruchwellen. *Sitzber. Akad. Wiss*, 1917.
- [106] H. Chason. Analytical Solution of Dam Break Wave with Flow Resistance: Application to Tsunami Surges. *Coast. Eng. J.*, 48:3341–3353, 2005.
- [107] R.J. LeVeque. *Numerical Methods for Conservation Laws, Lectures in mathematics*. ETH Zürich, ISBN 3-7643-2723-5, Switzerland, 1992.
- [108] N.N. Kuznetsov. Accuracy of some approximate methods for computing the weak solutions of a first-order quasi-linear equation. *USSR Comput. Math. and Phys.*, 16:105–119, 1976.
- [109] F.M. Henderson. *Open channel flow*. Nordby, g. edn., Civil engineering, MacMillan, New York, 1966.
- [110] S. Noelle, Y. Xing, and C. W. Shu. High-order well-balanced finite volume weno schemes for shallow water equation with moving water. *Journal of Computational Physics*, 226:29–58, 2007.
- [111] W.A. Thacker. Some exact solutions to the nonlinear shallow-water wave equations. *Journal of Fluid Mechanics*, 107:499–508, June 1981.
- [112] G. Gottardi and M. Venutelli. Central scheme for two-dimensional dam-break flow simulation. *Advances in Water Resources*, 27:259–268, 2004.

Author's Publications

- [113] M. Fišer. Numerické modelování proudění vody způsobené náhlým protržením hráze. Bachelor thesis, Department of Mechanics, Faculty of Applied Sciences, University of West Bohemia, Pilsen, CZ, 2009.
- [114] M. Fišer. Numerické řešení modelu mělké vody popsaného pomocí Saint-Venantových rovnic s uvažováním vlivu dna. Master thesis, Department of Mechanics, Faculty of Applied Sciences, University of West Bohemia, Pilsen, CZ, 2011.
- [115] M. Fišer, O. Bublík, L. Lobovský, and J. Vimmr. Modification of the central-upwind scheme providing c-property implemented on shallow water equations. *29th conference with international participation Computational Mechanics 2013 - Extended Abstracts*, pages 27–28, 2013.
- [116] M. Fišer, O. Bublík, L. Lobovský, and J. Vimmr. Problems and solutions connected with the wet/dry interface in the mathematical model of the shallow water equations. *30th Conference With International Participation In Computational mechanics - Extended Abstracts*, pages 31–32, 2014.
- [117] M. Fišer, O. Bublík, C. Mingham, and J. Vimmr. Novel limiting criterion for discontinuous galerkin method applied in shallow water equations. *Preprint submitted*, 2016.
- [118] M. Fišer, O. Bublík, and J. Vimmr. Novel limiting process of the discontinuous galerkin method applied to shallow water equations. *31st Conference With International Participation In Computational mechanics - Extended Abstracts*, pages 27–28, 2015.
- [119] Martin Fišer, Ilhan Özgen, Reinhard Hinkelmann, and Jan Vimmr. A mass conservative well-balanced reconstruction at wet/dry interfaces for the godunov-type shallow water model. *International Journal for Numerical Methods in Fluids*, 2016. fd.4246.