WAVELET-BASED NUMERICAL METHODS ADAPTIVE MODELLING OF SHALLOW WATER FLOWS

By

Dilshad Abdul Jabbar Haleem

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

The University of Sheffield
Faculty of Engineering
Department of Civil and Structural Engineering

November 2015

Abstract

Mesh adaptation techniques are commonly coupled with the numerical schemes in an attempt to improve the modelling efficiency and capturing of the different physical scales which are involved in the shallow water flow problems. This work designs an adaptive technique that avails from the wavelets theory for transforming the local single resolution information into multiresolution information in which these data information became accessible. The adaptivity of wavelets was first comprehensively tested via using an arbitrary function in which the spatial resolution adaptivity was achieved from the local solution itself and it was based on a single user-prescribed parameter. Secondly, the adaptive technique was combined with two standard numerical modelling schemes (i.e. finite volume and discontinuous Galerkin schemes) to produce two wavelet-based adaptive schemes. These schemes are designed for modelling one dimensional shallow water flows and are referred to the Haar wavelets finite volume (HWFV) and multiwavelet discontinuous Galerkin (MWDG) schemes. Both adaptive schemes were systematically tested using hydraulic test cases. The results demonstrated that the proposed adaptive technique could serve as lucid foundation on which to construct holistic and smart adaptive schemes for simulating real shallow water flow.

Acknowledgments

It is my pleasure to express my gratitude to the following people/Institutions for their contributions and support to the work conducted as a part of the thesis.

1. My first supervisor **Dr Georges Kesserwani** for all his time, efforts, advice and helpful criticism. I feel I also must thank him for all the trust that he has endowed me with through these years. It has been my honor to work with him, and I am happy to not only count him as my supervisor, but also as my friend.

2. My second and third supervisors **Prof. Dr Harm Askes** and **Dr Chris Keylock** for their guidance and efforts, in particular in the first year of my study.

3. My funding body, Ministry of Higher Education and Scientific research, Kurdistan regional Government-Iraq and University of Duhok for their financial support that made this study possible.

4. My thanks go to **Dr Daniel Caviedes-Voullieme**, for our collaborations and his friendship.

5. Many thanks go to **Nil Gerhard** and **Prof. Dr Siegfried Muller** from RWTH Aachen university- Mathematics Department, Germany, for their friendship and broadening my knowledge in the subject of multi-resolution analysis, through interesting discussion.

6. Many thanks go to all staff members in Civil and Structural Engineering department and Kroto research institute at the Sheffield University for their friendship and help.

7. Thanks to my **parents** for all the valuable things they have taught me and for all the things that motivated me to do, as well as my **brothers** and **sisters** for their wise words and encouragement. I also thank all my friends and relatives for their help and support.

8. A special thank goes to **Mrs Bareen S. Tahir**, my love and partner for her support, and giving me the incredible joy of taking care of my children

(**Nipeal** and **Niyaz**) along these years.

Publications

Parts of the work presented in this thesis have been published in:

- Journal of Hydroinformatics as an article that accepted in revised form in 9 June 2015. Available online in 9 July 2015. Paper's title is "**Haar wavelet-based adaptive finite volume shallow water solver**".

- Proceedings of the International Workshop on Hydraulic Structures: Data Validation, Coimbra, Portugal, 8-9 May 2015. PDF version ISBN 978-989-98435-9-2. Book version ISBN 978-989-20-5792-7. Paper's title"**Adaptive wavelet-based finite volume shallow water solver**".

- Proceedings of Advances in Numerical Modelling of Hydrodynamics Workshop, Sheffield, UK, 24 - 25 March 2015. Paper's title "**A wavelet-based Adaptation of Finite Volume Method for Shallow Water Modelling**"

- Proceedings of the 3rd IAHR Europe Congress with the theme Water Engineering and Research. Porto, Portugal, 14 April 2014 - 16 April 2014. Paper's title is "**A multi-resolution Discontinuous Galerkin method for one dimensional shallow water flow modelling**".

Contents

List of Figures

List of Tables

Notation

A	cross sectional area
B	the width of channel
CPU	central processing unit
DG	discontinuous Galerkin method
1D	one dimension space
2D	two dimension space
FVM	finite volume method
FDM	finite difference method
FEM	finite element method
$\mathbf{F}$	flux vector
$\tilde{\mathbf{F}}_{i+1/2}$	discrete approximation of the flux at the cell interfaces
$\tilde{\mathbf{F}}^{n_i}_{j+1/2}$	discrete approximation of the flux at the sub-cell interfaces associated with j
HWFV	Haar wavelets finite volume solver
MWDG	multiwavelet Discontinuous Galerkin solver
MRA	multiresolution analysis
MW	mutiwavelets
I_1	hydrostatic pressure
I_2	pressure due to the width changing
L	left state
RK	Runge-Kutta stage
RPCPU	relative performance of CPU time
R	right state
SD	standard deviation
SWE	shallow water equation

$\mathbf{U}$	vector of conserved variables
$\tilde{\mathbf{U}}$	discrete approximated solution vector over cell i
S_0	friction term attributed to the slope of bed
S_f	friction term attributed to the roughness of bed
S	source term vector
$\tilde{\mathbf{S}}$	discrete approximate of sources for cell i
TVB	total variation bounded
TVD	total variation diminishing
V	space of scaling basis functions
W	space of wavelets basis functions
N	highest resolution level
N_0	baseline mesh
N_n	mesh resolution at level n
n	current resolution level
j	shifting index
ϕ	the scaling basis function
ψ	the wavelets basis function
Φ	vector of the scaling basis function
Ψ	vector of the wavelets basis function
H	the matrix of the low-pass filter coefficients
G	the matrix of the high-pass filter coefficients
$\mathbf{P}$	projection function operator
PDEs	partial differential equations
p	Legendre polynomial order
$RMSE$	root mean square error
RME	relative mean error
g	the acceleration gravity
q	the flow rate per unit width
h	water depth
M_t	the total mass of water at time t
M_0	the total initial discrete mass
η	the free surface of water elevation

t	time
k	the order of accuracy
λ_1	forward characteristic
λ_2	backward characteristic
c	wave celerity
u	average velocity within the cross-sectional area
ρ	density of water
Ω	computational domains
σ	channel width at a particular depth
x	coordinate direction along the channel
Δx	spatial step
Δt	time step
CFL	Courant-Friedrichs-Lewys criterion
$\mathbf{e}$	vector of eigenvalues of the Jacobian matrix
J	Jacobian matrix
α	wave strength from Roe's decomposition

Chapter 1

Introduction

1.1 Motivation

Numerical models based on the shallow water flow have become an important part of management and planning for water resources projects. It is considered a valuable tool for hydraulic engineers in particular, due to their use in providing real-time predictions about the specific conditions of the flow in watercourses without the need for field measurements. In many cases, field measurements can be expensive and time-consuming.

These numerical models often use classical space discretisation techniques, such as the finite difference method (FDM), the finite volume method (FVM) or the finite element method. However, nowadays the common numerical strategies for approximating the system of shallow water flow are based on finite volume methods, in particular the modern class of shock capturing methods, which is also known as the Godunov-type approach because of their conservation properties. However, the conceptual framework of this approach is restricted to first-order accuracy and generally needs a dense mesh to solve on. To retain the accuracy order with fewer cells for the same underlying problem, the adaptive techniques have been used. However, there are some problems with this technique such as error-sensors, multiple user-chosen parameters and data transfer/recovery between different inner resolution scales.

High order finite volume methods are based on performing reconstruction procedures to recover point values from the cell average, which might lose the advantage of locality. Thus, the Discontinuous Galerkin (DG) methods become good alternative tools, because their conceptual framework allowing for high order accuracy within a well-founded Godunov-type formulation, as well as retaining locality and conservation properties. However, these methods still suffer from high computational cost because the CFL condition is very restrictive. Thus, the adaptive techniques (i.e., considering h-adaptation or p-adaptation or both) are useful options when it comes to maintaining the efficiency. However, since the Godunov-type methods are sometimes problematic new adaptive multiresolution schemes, that can partially or fully address these issues, motivate this work. This is achieved by incorporating the Haar wavelets and their generalisation, called multiwavelets, into the conceptual design of Godunov-type and Discontinuous Galerkin methods respectively.

1.2 Background

There are a large number of numerical schemes that have been developed to solve fluid dynamics equations. This section presents a review of several works related to FVM and the DG method in the modelling of shallow flows, followed by a summary of previous works related to wavelets and multiwavelets in the context of numerical modelling.

1.2.1 Finite Volume Method

There are an enormous number of studies covering modelling shallow water flow in one and two dimensions using FVM. The better known studies are dependent upon the Godunov method because it is a conservative scheme with local flexibility for any chosen grid mesh, e.g., structural or non-structural meshes (Guinot, 2003). Moreover, one of the most important issues in the modelling of the SWE system is bore formation and, for capturing such phenomena, Godunov methods are a great option. This is because Godunov-type methods include the communication of the discontinuous flow across cell interfaces via incorporating the local approximate solution within the discretisation process (Delis and Kampanis, 2009). The majority of Godunov-type numerical solvers were first developed in the context of the homogeneous system of equations (Glaister, 1988). But for the practical engineering applications where the steep slope, roughness and changes in topography exist, the proper integration of source term within the Godunov-type solution should be carefully handled. For this purpose, Bermudez and Vazquez (Bermudez and Vazquez, 1994a) proposed an upwind treatment of the source term in 1D SWE using different types of flux-splitting techniques. However, the results were unable to satisfy the *well-balanced* property (i.e."numerical balance between the momentum flux and source term", (Caleffi and Valiani, 2009)) in the most considered test cases. To ensure this *well-balanced* property whilst also considering irregular geometries of mesh, this scheme was improved by Vázquez-Cendón (Vázquez-Cendón, 1999) in which the source terms were upwinded in the same way as the numerical flux. In the same area of research, this idea has been explored in (Garcia-Navarro and Vazquez-Cendon, 2000; Brufau et al., 2002, 2004; Murillo and Garca-Navarro, 2010). Moreover, this improvement has since been used to increase the accuracy of the numerical schemes to a high order. High order finite volume methods such as WENO (Liu et al., 1994; Caleffi et al., 2006) and MUSCL (Alcrudo and Garcia-Navarro, 1993; Murillo et al., 2007; Hou et al., 2013) are based on performing reconstruction procedures to recover point values from the cell average, in which the monotonicity should be preserved. This might lead to losing the advantage of locality when comparing with the original FV method (Zhou et al., 2001; Kesserwani and Wang, 2014) whereas, in the DG method (Section 1.2.2), the numerical solution over each element is not reconstructed artificially by extrapolating from neighbouring elements for obtaining a high order accurate solution, as the approximate solution is associated with a polynomial order (see Figure 1.1). Therefore, the DG method has a good advantage in terms of accuracy compared with FVM when the same order of accuracy and the same number of cells are considered (Shelton, 2009). Following decades of research, FV Godunov-type methods have become widely applied to simulate real-scale flooding and have been adopted into commercial hydraulic modelling software packages such as TUFLOW-FV and RiverFlow2D PLUS. Nevertheless, real large-scale shallow flows have complex flow features such as shocks, contact discontinuities and a wide range of spatial scales. Typically, the computational domain is discretised uniformly using a large number of cells, given that the position of flow features is usually unknown. Capturing certain small scales within a coarse mesh simulation maybe difficult without causing computational cost trade-off. Therefore, automated mesh adaptation comes in handy to improve modelling efficiency and capture the various physical scales involved in shallow water flows.

Various adaptive techniques have been developed within the FV framework intended to solve shallow water equations (SWE).These include moving mesh methods (Skoula et al., 2006) or static grids with local refinement methods (Nikolos and Delis, 2009; Caviedes-Voullième et al., 2012). However, most of the present techniques to date are achieved over patch of grid, in a decoupled manner, which controversially gives rise to many problematic effects (Nemec and Aftosmis, 2007). For instance, they require error-sensors and multiple user-chosen parameters (e.g. for setting up grid resolution coarsening vs. refinement), which introduce sensitivity (e.g. can lead to inadequate or excessive resolution), inflexibility and problem-dependency (e.g. due to the need to tune many parameters for each simulation problem). They also lack a rigorous strategy to accommodate flow data transfer and recovery between various inner resolution scales (given the changing nature of the mesh). Therefore, improving the conceptual design of the FVM to allow scaling in spatial-resolution should be considered and this motivates the current research.

Figure removed due to copyrights

Figure 1.1: Contrasting 1D high-order spatial approximations; left: extrinsic (non-local) finite volume polynomial estimates built from the original (local) piecewise-constant evolution data; right: discontinuous Galerkin (local) evolution data defining intrinsic piecewise-polynomials, (Shelton, 2009).

1.2.2 Discontinuous Galerkin method

In the 1970 s, the DG method was introduced by Reed and Hill (Reed and Hill, 1973) to improve the solution of steady-state neutron transport equation in which the approximate solutions were computed cell by cell and the sequence of cells was based on the characteristic direction. This is due to the neutron equation that is known as a time independent linear hyperbolic equation. Later many studies were conducted on the DG method with the intention of proving the mesh convergence of order k, such as Lesaint and Raviart (Lesaint and Raviart, 1974) who conducted a study choosing two types of meshes. The results showed that the convergence rate was $(\Delta x)^k$ for triangular mesh, while for Cartesian mesh the rate of convergence was $(\Delta x)^{k+1}$. Furthermore, Johson and Pitkäranta (Johnson and Pitkäranta, 1986) confirmed that the optimal converge rate was equivalent to $(\Delta x)^{k+1/2}$ for general meshes, and this was confirmed by Peterson (Peterson, 1991).

Early application of the DG method for solving 1D nonlinear scalar conservation laws was performed by Chavent and Salzano (Chavent and Salzano, 1982). They applied piecewise linear elements in DG space and the forward Euler approach for time step. The scheme was unconditionally unstable except if imposing a very restrictive time step. To solve this problem, a total variation diminishing means scheme (TVDM) and total variation bounded scheme (TVB) were introduced by Chavent and Cockburn (Chavent and Cockburn, 1987). In these schemes, the requirement of the Courant-Friedrichs-Lewy (CFL) number should be equal to or less than 1/2 for ensuring stability condition. However, they were first-order accurate in time and the slope limiter was activated globally. Thus, the quality of the solution was affected in the region where the solution was smooth. To overcome this problem, the Rung-Kutta discontinuous Galerkin method was introduced by Cockburn (Cockburn, 1987). They merged an improved version of Shu (Shu, 1987) slope limiter with the second-order total variation diminishing (TVD) of the Runge-Kutta Discontinuous Galerkin method. The resulting scheme showed a stability for (CFL $< 1/3$) and can preserve formal accuracy in the smooth region as well as ensuring sharp shock resolution without oscillations. Extending the RKDG method to high-order with a general slope limiter for the scalar conservation law was performed by Cockburn and Shu (Cockburn and Shu, 1989). Later, the integral framework of the RKDG method for convection-dominated problems was demonstrated by Cockburn *et al.* (Cockburn and Shu, 2001), which became the cornerstone in this field.

The RKDG method combines the finite volume method and finite element method. Therefore, it contains the advantageous aspects of both methods. Firstly, it can handle the boundary conditions and complex geometries. Secondly, it provides a high-order approximation by using the high-order interpolating functions. Thirdly, it has a local formulation. Therefore, application of the global assembly matrix is not required, which makes the method highly parallelisable and the implementation of the $hp-$ adaptive mesh refinements is straightforward.

For solving the conservation law of SWEs, the earliest implementation of the RKDG method was performed by Schwanenberg and Köngeter (Schwanenberg and Köngeter, 2000) with application to simulate shock wave problems including dam break flows and hydraulic jumps. Many others investigators applied the RKDG method on 1D and 2D to verify its accuracy, stability and convergence on 1D and 2D meshes, considering different types of test cases of various complexity (see, among others, (Tassi and Vionnet, 2003; Schwanenberg and Harms, 2004)). The most relevant works that contributed to the development of the RKDG to water flow modelling were focused on: **i.** the well-balanced property that introduced by Bermudez and Vezquez (Bermudez and Vazquez, 1994a) in which the numerical scheme was able to properly preserve a quiescent flow (e.g., see among others, (Audusse et al., 2004; Kesserwani and Liang, 2012b; Caleffi and Valiani, 2012)); **ii.** introducing local slope limiting to improve the conservation property (e.g., see among others, (Krivodonova et al., 2004; Qiu and Shu, 2005; Kesserwani and Liang, 2012b)); **iii.** the treatment of wet/dry interfaces in which a several techniques have been introduced,for instance, Bokhove (Bokhove, 2005) used a 1D moving mesh method to determine the wet/dry interfaces, Bunya *et al.* (Bunya et al., 2009) used the fixed mesh approach using the traditional thin water layer with applying a special treatments in the numerical flux and **iv.** gathering all these advanced topics (i.e., **i**, **ii** and **iii**) in a unified RKDG-scheme for realistic simulations. For instance, a new RKDG algorithm was presented by Kesserwani and Liang (Kesserwani and Liang, 2010a) to solve 2D SWE considering the bed and friction source terms.

Despite the progress that has been made in developing the RKDG methods, the issue of high computational cost associated with these methods is still obstructing their widespread application (Kesserwani and Liang, 2012a). Therefore, current attempts to improve RKDG shallow water solvers are mostly focused on reducing the computational cost. To this effects, introducing spatial resolution adaptivity (or h-adaptation) has commonly occupied researchers in the last decade affected by the locality property. This property further allows researchers to locally scale in the accuracy-order (or p-adaptivity). For instance, Kubatko *et al.* (Kubatko et al., 2006) compared global p-adaptation versus global h-adaptation in an RKDG method in terms of the computational efficiency for solving the SWE on unstructured triangular grids. The authors clearly concluded that the utilise of the p-adaptation is more efficient than the utilise of h-adaptation even in regions where the solution is highly non-linear. The reason was associated with the fact that p-adaptation works within the natural formulation of the RKDG method, whereas the h-adaptation is performed in a decoupled manner dictated by the external mesh. Therefore, Kubatko *et al.* (Kubatko et al., 2009) installed solely the dynamic p-adaptation in the RKDG for solving 2D SWE which, in terms of run-time efficiency, they found to be better than both the global h-adaptation and the global p-adaptation.

Concerning the local dynamic h-adaptation, Remacle *et al.* (Remacle et al., 2006) first investigated it in the high-order RKDG solution to the SWE. Bader *et al.* (Bader et al., 2010) reported a new dynamic h-adaptation mesh-generator in the context of a RKDG shallow flow solver, with a particular focus on minimizing memory demand. Both papers considered 2D triangular meshes, but do not provide information on the associated computational saving or the error generated due to the instalment of the dynamic h-adaptation process. Kubatko (Kubatko et al., 2009) coupled dynamic h-adaptation on quadrilateral meshes with an RKDG numerical solver for 2D SWE with an application to reproduce real-scale flood simulations. Based on a qualitative and quantitative analyses the use of h-adaptation in the RKDG numerical solver has shown a tendency to introduce uncertainties in the modelling. In addition, it has been found to compromise the design of an error-sensor and the quality of the initial mesh. Therefore, improving the conceptual design of the RKDG method to allow not only scaling in accuracy-order but also scaling in spatial-resolution is a complementary way forward.

1.2.3 The wavelets and multiwavelets introduced with numerical modelling

In the 1980s, wavelets became a topic of interest in many areas of science and engineering, for instance in signal processing, image compression, statistical analysis, speech recognition and other fields (Grossmann and Morlet, 1984). The first researcher who used the term wavelet (*Ondelette*) was Jean Morlet, while working on the analysis of seismic returns for Elf Aquitaine Oil company (Gargour et al., 2009). In fluid mechanics, wavelets were used at first to analyse turbulent flow. Later, much attractive work has been done using wavelet methods to simulate coherent vortices in 2D and 3D flow (Schneider et al., 1997; Farge and Schneider, 2001; Yoshimatsu et al., 2013). The principal idea of these methods is to apply wavelet decomposition to a turbulent flow so as to resolve the energetic eddies. Mehra and Kevlahan (Mehra and Kevlahan, 2008) used the wavelet adaptive collocation method for solving the partial differential equations (PDE) on a sphere with application to simulate geophysical flows. The first attempt to use multiresolution (MR) approach for hyperbolic conservation laws was performed by Harten (Harten, 1995). The main idea behind this method was to conceptualise data in a hierarchical form and use the multiscale wavelet basis as the approximation space. The author employed the MR approach to transform the cell average arrays associated with the FVM into a various form that reveals insight into the local approximate solution. The cell averages on the given highest resolution level were represented as cell averages on some coarse level where the fine scale information is encoded in arrays of detail coefficients of promoting resolution. By using the MR approach the computation is accelerated while controlling the flux evolution in regions where the solution is smooth, and the solution remains at the same level of accuracy as in the FVM. This method has been successfully implemented for 2D Cartesian meshes, Bihari and Harten (1995), curvilinear meshes, Dahmen et al. (2001) and unstructured meshes, Abgrall (1998). Following Harten's original ideas, Müller and Stiriba (Müller and Stiriba, 2007) and Cohen et al. (Cohen et al., 2003) have further improved the approach to minimize the computational cost while preserving the accuracy of solution as FVM. To increase the accuracy of the solution and in the meantime control the adaptive resolution of discontinuities, a more comprehensive multi-scale framework needs to be designed. Therefore, in 2008, Shelton illustrated the utility of merging the multiwavelets (MW) method with the DG method for unsteady compressible flow problems. The author successfully utilised multiwavelets to refine the basis of the approximation space of the local polynomial solution used in DG structure. Therefore, the MW basis is able to enhance the computational solution by zooming across the different scales within the computational solution framework (see Figure 1.2). In particular, in an area where further detail is not necessary (Shelton, 2009). This is a new concept that is only supported by some basic investigations and more investigation is needed. According to the current literature, apart from the recent papers by the team of Kesserwani and Müller (Kesserwani et al., 2014, 2015; Haleem et al., 2015), the implementation and implication of this idea in addressing practical aspects of shallow water flow simulation have not yet been fully explored.

Figure removed due to copyrights

Figure 1.2: Illustrates the comparison of the computation of the weak derivative operation in traditional local refinement and multi-resolution setting, (Shelton, 2009).

1.3 Objectives

The research formulates new HWFV and MWDG numerical solvers for simulation of one dimensional shallow water. These schemes are capable of achieving the dynamic adaptation from the solution itself and it is based on a single threshold value only. The adaptive HWFV formulation combines the Haar wavelets within the finite volume formulation, while the adaptive MWDG formulation combines the generalisation of Haar wavelets, called multiwavelets within the Discontinuous Galerkin formulation.

1.4 Outline of thesis

The next chapter presents an overview of the shallow water model in one dimensional flow. In Chapter 3, the multiresolution analysis and its mathematical properties are introduced, with particular focus on our choice of basis function, namely Haar wavelets and multiwavelts of (Alpert, 1993; Alpert et al., 2002) that are used to drive to two types of filters that used for performing the transformation for any two sequences of resolution levels. Moreover, an example of $\sin(2x)$ is presented to prove the feasibility of using these filter coefficients within the framework of FV and DG methods. Chapter 4 consists of two parts: the first part presents the standard FV and DG method, which helps to understand their mathematical conception and the conceptual design of these methods related to hydrodynamic modelling. The second part, based on the multiscale decomposition of the Legendre polynomial basis of DG, the merge of the MW and DG framework is reconstructed to provide complete solution of a hierarchy that can scale in both resolution and local scales. This is followed by presenting the HWFV formulation that can scale in resolution. In Chapter 5, the performance of these models is tested, analysed and discussed. Chapter 6 presents the conclusions and recommendations for future works.

Chapter 2

Shallow Water Equations

2.1 Introduction

The water over the Earth's surface includes overland run off in both natural (river, stream, oceans) and man-made environments (open channels, canals). They rely on gravitational acceleration force, so their movement is referred to as "free surface gravity flow" and their physics is usually represented by the mass (continuity equation) and momentum (dynamic equation) conservation in three space dimensions.

When modelling free surface flow, the shallow water equations (*Saint Venant equations*) are often considered the governing equations for the mathematical model. Here the term "shallow" refers to the depth of the water, which is usually much less than the horizontal scale of the flow length. This equation can be obtained by depth-averaging the Navier-Stokes equations based on the assumption that the vertical variation of the water is restricted via considering its importance only for the dynamic flow (Tan, 1992).

The objective of this chapter is to explicitly present one-dimensional shallow water equations. Firstly, the underlying assumptions of shallow water equations is presented below. Secondly the derivation of the shallow water is illustrated. Finally, we present the conservative form of the shallow water equations in which is often considered as the basis of numerical models.

2.2 The underlying assumptions of the shallow water equations

The hypothesis states that the shallow water is conceived between two interfaces and can be considered as boundary conditions for deducing the result of the shallow water equations. The fluid-fluid boundary (free surface) is denoted by $z = \overline{\eta}(x, y, t)$, while the solid-fluid boundary (bottom) is denoted by $z = -h + z_b(x, y)$, Figure 2.1 illustrates these conditions, where h is the water depth, z_b represents the height of bottom variation, and the interface variation is $\overline{\eta}$. Once the 3D free surface flow equations are determined, the derivation of 3D shallow water equations can be obtained via studying the characteristics scales of the problem. A number of fundamental assumptions should be defined to simplify the problem. These assumptions are inherent within the model and can be summarised as follows:

1. The distribution of the pressure with water depth is hydrostatics, i.e. the vertical component of the water acceleration is negligible.

2. The friction losses in unsteady flow are represented using the same empirical equations that used for steady flow, i.e. *Manning's equation*.

3. The bed slope is small so the tangent of the angle can be computed by the angle between the bed level and the horizontal plane.

4. The water has no viscosity and has a constant density (ρ), i.e. the temperature of water is constant during the flow of water.

2.3 The derivation of the shallow water equations

The shallow water equations have appeared in literature in many forms and can be written as a set of differential or integral relations. The following derivation considers the integral form which can be found in a book written by Cunge (Cunge et al., 1980) and applies to an arbitrary shaped channel such as that shown in Figure 2.2. Suppose that a control volume of water in the (x,t) plane is bounded

Figure removed due to copyrights

Figure 2.1: Schematic for the system of the shallow water equations.

between two sections $x = x_1$, $x = x_2$ and $t = t_1$, $t = t_2$ such as shown in Figure 2.3. By considering all assumptions listed in Section 2.2 and noting that there is no lateral inflow into the control volume, and by assuming that the velocity (u) in x-direction is constant over a cross-sectional area (A), then the change of mass in the control volume can be determined via computing the difference between the inflow mass at x_1 and outflow mass at x_2 and then performing the time integration between t_1 and t_2 which gives:

$$\int_{t_1}^{t_2} [(\rho u A)_{x_1} - (\rho u A)_{x_2}] dt \tag{2.1}$$

Due to mass conservation, the net inflow obtains from equation 2.1 is equivalent to the change of water volume between x_1 and x_2 during the time interval which is given by:

$$\int_{x_1}^{x_2} [(\rho A)_{t_2} - (\rho A)_{t_1}] dx \tag{2.2}$$

By substituting the discharge $Q = Au$ into the equations 2.1 and 2.2 gives the integral relation of the mass continuity for a channel has an arbitrary cross section.

$$\int_{x_1}^{x_2} [(A)_{t_2} - (A)_{t_1}] dx + \int_{t_1}^{t_2} [(Q)_{x_2} + (Q)_{x_1}] dt = 0 \tag{2.3}$$

For the dynamic equation, we apply Newton's second law to the control volume that states, the change of momentum in the control volume from t_1 to t_2 is equivalent to the net inflow of the momentum within the control volume plus the integral of the external forces which cause the acceleration in the control volume with respect to the same internal time. Hence, momentum and momentum flux through cross section can be defined as

$$Momentum = \rho A \times u \tag{2.4}$$

$$Momentumflux = Momentum \times u = \rho u^2 A \tag{2.5}$$

The difference of the momentum flux that is entered through the section x_1 and leaving through the section x_2 with respect to the time interval t_1 and t_2 can be given as follows:

$$M_f = \int_{t_1}^{t_2} [\left(\rho u^2 A\right)_{x_1} - \left(\rho u^2 A\right)_{x_2}]dt \tag{2.6}$$

At a particular time, we can define the net momentum in the control volume as

$$\int_{x_1}^{x_2} \rho u A dx \tag{2.7}$$

and it is increased (ΔM) over the time interval, which gives

$$\Delta M = \int_{x_1}^{x_2} [(\rho u A)_{t_2} - (\rho u A)_{t_1}]dx \tag{2.8}$$

It can be seen from the Figures 2.3 and 2.4 that only three important forces are acting on the control volume in x-direction. The first force is the pressure force (F_{p_1}) that is produced from the change in static pressure at boundaries in which the pressure force $F_{p_1}^{*}$ acts at section x_1 can be defined using equation 2.9. Furthermore, the pressure force acts at section x_2 is defined using $F_{p_1}^{**}$.

$$F_{p_1}^{*} = g \int_{0}^{h(x)} \rho[h(x) - \eta]\sigma(x,\eta)d\eta \tag{2.9}$$

where η = depth integration variable; $h(x,t)$ = water depth and (x,η) = width of the cross section at a depth η such that $\sigma(x,h) = B(x)$ at the free surface width. So the time integral of the net pressure force F_{p_1} can be expressed as follows:

$$\int_{t_1}^{t_2} F_{p_1} dt = \int_{t_1}^{t_2} (F_{p_1}^{*} - F_{p_1}^{**})dt = g \int_{t_1}^{t_2} [(\rho I_1)_{x_1} - (\rho I_1)_{x_2})]dt \tag{2.10}$$

where

$$I_1 = \int_{0}^{h(x)} = [h(x) - \eta]\sigma(x,\eta)d\eta \tag{2.11}$$

Consider the short length of channel dx. The pressure force is increased as the width of the corresponding wetted area $(d\sigma \cdot d\eta)$ increases for constant water depth $h = h_0$ and then multiplying its centroid by the distance from the free surface $h(x) - \eta$ gives:

$$\rho g \left[\left(\frac{\partial \sigma}{\partial x} \right) dx \cdot d\eta \right]_{h=h_0} [h(x) - \eta] \tag{2.12}$$

To compute all forces acting upon the control volume that is given between the section x_1 and the section x_2, this force is to be integrated between $\eta = 0$ and $\eta = h(x)$:

$$F_{p_2} = \int_{x_1}^{x_2} \rho g \int_0^{h(x)} [h(x) - \eta] \left[\frac{\partial \sigma(x, \eta)}{\partial x} \right]_{h_0} d\eta \, dx \tag{2.13}$$

and also F_{p_2} is integrated over the time interval and it can be written as:

$$\int_{t_1}^{t_2} F_{p_2} dt = g \int_{t_1}^{t_2} \int_{x_1}^{x_2} \rho \, I_2 \, dx \, dt \tag{2.14}$$

where

$$I_2 = \int_0^{h(x)} (h - \eta) \left[\frac{\partial \sigma}{\partial x} \right]_{h=h_0} \partial \eta \tag{2.15}$$

Since the slope of the bed channel has been assumed to be small (see Section 2.2), and can be obtained from equation 2.17, the total contribution of the gravitational force F_g, with respecting the time interval, can be expressed as

$$\int_{t_1}^{t_2} F_g \, dt = \int_{t_1}^{t_2} \int_{x_1}^{x_2} \rho \, g \, A \, S_0 \, dx \, dt \tag{2.16}$$

$$S_0 = -\frac{\partial z_b}{\partial x} = \tan \theta \approx \sin \theta \tag{2.17}$$

The frictional resistance force F_f is obtained due to the existence of shear force along the channel bed and banks. In most instances, it is expressed by following the expression of Ven Te Chow (Te Chow, 1959) which states that the energy gradient force is equivalent to the friction resistance force when flow is steady. The shear force per unit length $\rho g A S_f$, where S_f is the friction slope, is to be integrated over time interval to compute the total contribution of friction force applied to the control volume.

$$\int_{t_1}^{t_2} F_f \, dt = \int_{t_1}^{t_2} \int_{x_1}^{x_2} \rho \, g \, A \, S_f \, dx \, dt \tag{2.18}$$

Consider the momentum conservation law, which indicates that the change in momentum ΔM, is equivalent to the sum of external force and the resultant of momentum M_f, thus:

$$\Delta M = M_f + \int_{t_1}^{t_2} F_{p1}\, dt + \int_{t_1}^{t_2} F_{p2}\, dt + \int_{t_1}^{t_2} F_g\, dt - \int_{t_1}^{t_2} F_f\, dt \tag{2.19}$$

Due to the consistent density of water and taking into account all equations from 2.6 to 2.19, the general integral form of momentum equation becomes

$$\begin{aligned} \int_{x_1}^{x_2} [(uA)_{t_2} - (uA)_{t_1}]\, dx = \int_{t_1}^{t_2} [(u^2A)_{x_1} - (u^2A)_{x_2}]\, dt + g \int_{t_1}^{t_2} [(I_1)_{x_1} - (I_1)_{x_2}]\, dt \\ -g \int_{t_1}^{t_2} \int_{x_1}^{x_2} \rho\, I_2\, dx\, dt + g \int_{t_1}^{t_2} \int_{x_1}^{x_2} A(S_0 - S_f)\, dx\, dt \end{aligned} \tag{2.20}$$

Alternatively, the equation 2.20 can be rewriting using A and Q, then

$$\begin{aligned} \int_{x_1}^{x_2} [(Q)_{t_2} - (Q)_{t_1}]\ dx = \int_{t_1}^{t_2} \underbrace{\left[\left(\frac{Q^2}{A}\right)_{x_1} - \left(\frac{Q^2}{A}\right)_{x_2} \right]}_{\textcircled{A}} dt + g \int_{t_1}^{t_2} \underbrace{(I_1)_{x_1} - (I_1)_{x_2}]}_{\textcircled{B}}\ dt \\ -g \int_{t_1}^{t_2} \int_{x_1}^{x_2} \rho\, I_2\, dx\, dt + g \int_{t_1}^{t_2} \int_{x_1}^{x_2} A(S_0 - S_f)\, dx\, dt \end{aligned} \tag{2.21}$$

Suppose that the distance between x_2 and x_1 becomes very small and also assume that the flow variables are differential and continuous. The Tayor series expansion can then be applied to Q and A at t_2, so

$$(Q)_{t_2} = (Q)_{t_1} + \frac{\partial\, Q}{\partial t}\Delta t + \frac{\partial^2 Q}{\partial t^2}\frac{\Delta t^2}{2} + ... \tag{2.22a}$$

$$(A)_{t_2} = (A)_{t_1} + \frac{\partial\, A}{\partial t}\Delta t + \frac{\partial^2 A}{\partial t^2}\frac{\Delta t^2}{2} + ... \tag{2.22b}$$

By ignoring all terms in equation 2.22 that are greater than the first-order derivative term and performing the limit as Δx and Δt trend to zero, the continuity equation can be rewritten as:

$$\int_{x_1}^{x_2} \int_{t_1}^{t_2} \left[\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} \right] dt\, dx = 0 \tag{2.23}$$

Applying the Taylor series to the terms Ⓐ and Ⓑ in equation 2.21 gives

$$\left(\frac{Q^2}{A}\right)_{x_2} - \left(\frac{Q^2}{A}\right)_{x_1} = \frac{\partial\, (Q^2/A)}{\partial x}\Delta x + \frac{\partial^2\, (Q^2/A)}{\partial x^2}\frac{\Delta x^2}{2} + ... \tag{2.24a}$$

$$(I_1)_{x_2} - (I_1)_{x_1} = \frac{\partial\, (I_1)}{\partial x}\Delta x + \frac{\partial^2\, (I_1)}{\partial x^2}\frac{\Delta x^2}{2} + ... \tag{2.24b}$$

Figure removed due to copyrights

Figure 2.2: The cross section area of the non-prismatic open channel, (Cunge et al., 1980).

By considering only the first-order term in equations 2.24 and allowing that the limit of Δx and Δt goes to zero, the equation 2.21 can then be written as

$$\int_{x_1}^{x_2}\int_{t_1}^{t_2}\left[\frac{\partial Q}{\partial t}+\frac{(\partial(Q^2/A)}{\partial x}\right]dt\,dx=-g\int_{x_1}^{x_2}\int_{t_1}^{t_2}\left[\frac{\partial I_1}{\partial x}-I_2-A(S_0-S_f)\right]dt\,dx \tag{2.25}$$

The equations 2.23 and 2.25 must be performed for throughout the region. They can therefore be written in term of differential equation, thus:

$$\frac{\partial A}{\partial t}+\frac{\partial Q}{\partial x}=0 \tag{2.26a}$$

$$\frac{\partial Q}{\partial t}+\frac{\partial}{\partial x}\left(\frac{Q^2}{A}+gI_1\right)=gA(S_0-S_f)+gI_2 \tag{2.26b}$$

This form of the equation 2.26 is called the "*divergent form*" and it represents the system of conservation laws based on the de St Venant hypothesis.

A prismatic rectangular channel has been considered in this thesis in which the I_1 term can be simplified to $I_1 = A^2/2B$ and the I_2 becomes zero. In addition, it assumes that the flow parameters at a given instance in time are varied only in (x,t) plane, so the equation 2.26 becomes:

$$\frac{\partial h}{\partial t}+\frac{\partial q}{\partial x}=0 \tag{2.27}$$

$$\frac{\partial h}{\partial t}+\frac{\partial}{\partial x}\left(\frac{q^2}{h}+\frac{1}{2}gh^2\right)=gh(S_0-S_f) \tag{2.28}$$

It may be more convenient to write the equations 2.27 and 2.28 in vector form as

$$\frac{\partial \mathbf{U}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}=\mathbf{S} \tag{2.29}$$

Figure removed due to copyrights

Figure 2.3: The section view of the control volume, (Cunge et al., 1980).

Figure removed due to copyrights

Figure 2.4: The distribution of pressure forces, plan view, (Cunge et al., 1980).

where

$$\mathbf{U} = [\,h, q]^T \quad \mathbf{F}(\mathbf{U}) = \left[\,q, gh^2/2 + q^2/h\right]^T \quad \mathbf{S}(\mathbf{U}) = [\,0, gh(S_0 - S_f)]^T$$

Chapter 3

Wavelets and Multiwavelets

3.1 Introduction

The theory of wavelets is vast and it is being widely used in many disciplinary applications, ranging from signal processing and denoising to the fast solution of partial differential equations, since the theory allowed for the effective approximation of a large class of functions, including those with discontinuities and sharp spikes. However, it is considered a relatively new topic in the field of computational fluid dynamics and only appeared in the literature a few decades ago in relation to the analysis of turbulent flow (Schneider and Vasilyev, 2010). This is due to the sophistication of the theory and that the majority of available studies about wavelets was written by mathematicians at such level that is difficult for anyone to avail of them (Soman et al., 2010). Thus, in this chapter our goal is to describe the whole theory of discrete wavelets and multiwavelets in details and how they are construed within the target of integrating them into the framework of FV and DG. To do so, we will start with describing the concept of multiresolution analysis and along with how it can be exploited to construct the multiwavelets, and then we come to the end of the chapter by giving an example of representing function f in different resolutions.

3.2 Multiresolution Analysis

The multiresolution analysis concept (MRA) plays an important role in the context of wavelets and multiwavelets theory because it gives one the ability to drive their own families of wavelet bases (i.e. self-similar functions) without any restriction. For instance, obtaining the refinement equations that links the basis functions on one resolution level to their scaled version on higher resolution levels (Gargour et al., 2009). This allows one to access the finer details of the approximated function and manipulating them can be used to promote the representation of the function to higher resolution or demoted it to a lower resolution.

In fact, this idea was first proposed by Mallat (1989) and it was called multiresolution *approximation*. But in this thesis, the multiresolution *analysis* term is used to be consistent with the literature. More detail about MRA can be found in book written by Keinert (2003). Here we present the Alpert multiresolution analysis (Alpert, 1993) using the standard notation of wavelets and multiwavelets with the difference that our discussion is limited to the interval $[-1, 1]$ instead of the real line to be consistent with the compact-support of Legendre polynomial basis. Therefore, the first principle is to suppose that the MRA of $L^2([-1, 1])$ is orthogonal and has an *infinite nested sequence* of sub-spaces.

$$V_k^0 \subset V_k^1 \subset V_k^2 \subset \ldots V_k^n \subset \ldots \subset L^2\left([-1, 1]\right) \tag{3.1}$$

with the following properties:

1. $\text{clos}L_2\,(\bigcup_{n=0}^{\infty}\; V_k^n)$=$L^2([-1, 1])$.
2. $\bigcap_{n=0}^{\infty} V_k^n = \{0\}$.
3. $f(x) \;\in\; V_k^n \Longleftrightarrow f(2x) \;\in\; V_k^{n+1}, \forall_n \in \mathbb{N}$.
4. $f(x) \;\in\; V_k^n \Longleftrightarrow\; f(x - 2^{-n}j) \;\in\; V_k^n,\; \forall_n \in \mathbb{N}, 0 \leq\; j \;\leq\; 2^n - 1$.
5. There exists a vector function $\Phi \in L^2([-1, 1])$ of length $k + 1$ such that the vector components ϕ_i form an orthogonal basis of V_k^0.

This means that if we can construct a basis functions of V_k^0, which consists of $k + 1$ functions, the basis functions of any space V_k^n can be obtained via applying dilation (compression by a factor 2^n, property 3), and translation (to all shifting points at level n, property 4), on the original $k+1$ functions. Consider equation 3.1, the basis functions of space V_k^n can be expanded in V_k^{n+1} space as follows:

$$\phi_{i,j}^n(x) = 2^{n/2}\phi_i(2^n(x+1) - 2j - 1) \quad (j = 0, 1, .., 2^n - 1) \tag{3.2}$$

and the orthogonality condition of MRA is valid if

$$\langle \phi_{i,j}^n, \phi_{l,m}^n \rangle = \delta_{i,l}\, \delta_{j,m} \tag{3.3}$$

where δ is the delta function, j, m are the location index and i, l are the polynomial order. It means that the basis functions on one level are orthogonal both within one function vector and through all possible translation, but not through the different levels. This has an advantage of constructing the filter matrices that explained in the Subsection 3.3.4.

By considering the concept of nested sequence of sub-spaces, as in property 1, and they are non-overlapping, as in property 2, an orthogonal complement subspace called wavelets space (W_k^n) can be defined between any two sequences of sub-spaces V_k^{n-1} and V_k^n that is

$$V_k^{n+1} = V_k^n \oplus W_k^n \tag{3.4}$$

where the W_k^n spaces inherit the MRA properties from the space V_k^n. Thus given any vector basis functions Φ in V_k^n, there is another vector Ψ that contains basis function of the length of $k+1$ called wavelets. Similarly to equation 3.2, its translation and dilation at level n form a basis functions for W_k^n.

$$\psi_{i,j}^n(x) = 2^{n/2}\psi_i(2^n(x+1) - 2j - 1) \quad (j = 0, 1, .., 2^n - 1) \tag{3.5}$$

By considering the orthogonality condition of the MRA, the bases ψ fulfil the same orthogonal condition as in equation 3.3, and if we merge equation 3.1 and equation 3.4, they must be also orthogonal with respect to the different resolution levels. This is an important property of wavelet basis because of two reasons; first the wavelets transformation will be straightforward; second, the information captured by one wavelet basis ψ is completely independent from the information captured by another basis (Soman et al., 2010).

$$\langle \psi_{i,j}^n, \psi_{l,m}^{n'} \rangle = \delta_{i,l}\, \delta_{j,m}\, \delta_{n,n'} \tag{3.6}$$

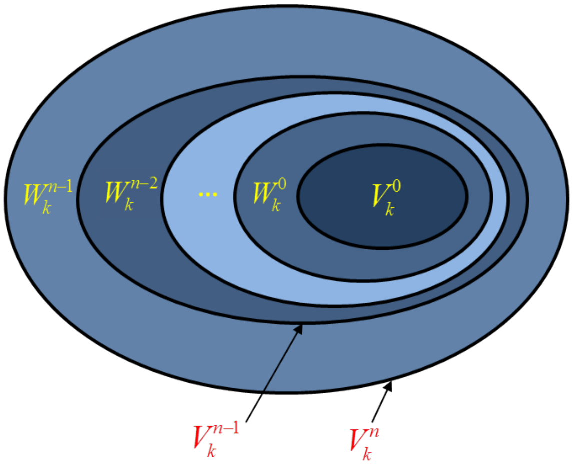


Figure 3.1: Decomposition of spaces V_k^n into the complement spaces W_k^n.

By applying equation 3.4 recursively, this yields an important relation (see Figure 3.1) that has the advantage of decomposing any space V_k^n as the sum of a single space V_k^0 along with a sequence of wavelet spaces W_k^{n-1}:

$$V_k^n = V_k^0 \oplus W_k^0 \oplus W_k^1 \oplus \cdots \oplus W_k^{n-1} \tag{3.7}$$

The concept in equation 3.7 is considered as a keystone of constructing the multiwavelets. The choice of multiwavelets as a tool for numerical purposes is due to two main reasons. First, they are sharing the same compact support. Thus, for a high order of approximation, the compact support length of these basis functions is not need to be increased. This aids to preserve the orthogonality condition which has an advantage in adaptive solvers of PDEs. Second, they are discontinuous and can be used for representing the integral operators of PDEs (Alpert et al., 2002).

3.3 Multiwavelets

Many approaches have been described for choosing the basis functions ϕ and ψ that are used to span the spaces V_k^n and W_k^n respectively. This leads to obtain different wavelet families such as Daubenchies's family in which the ϕ and ψ are compactly supported and constructed by using specific designed filter matrices (Daubechies et al., 1988). In contrast to the Alpert's family, where the basis functions ϕ and ψ are defined via applying the Alpert algorithms on the *legendre polynomial* basis functions. This leads to notation of multiwavelets.

Here, we define the scaling space V_k^n as a space of piecewise polynomial functions

$$\begin{aligned} V_k^n = \{f : is\, a\, polynomials\ of\ degree\ \leq\ k \\ on\, support of, (-1+2^{-n+1}j, -1+2^{-n+1}(j+1)) \\ for\, j = 0, 1, .., 2^n - 1\, and\, vanishes\ elsewhere\} \end{aligned} \tag{3.8}$$

and the equation 3.8 fulfils all conditions of MRA, provided the scaling basis functions ϕ are chosen to be orthogonal.

3.3.1 Scaling basis functions

The simplest way of constructing the scaling basis is to start with the standard $k+1$ polynomial functions $\{1, x, x^2, ..., x^k\}$ spanning the space of polynomial of degree $\leqslant k$ and then considering the orthogonality and normality conditions of the MRA with respect to the $L^2[-1,1]$ interval.

$$\int_{-1}^{1} P_l(x) P_i(x)\, dx \;=\; 0, \qquad l \neq i \tag{3.9}$$

Here, $\{P_l(x)\}_{l=0,1,\cdots,p\in\mathbb{N}}$ are the well-known *legendre polynomials* of order p shown in Figure 3.2 , and their orthogonality property is given in equation3.9. In addition, they are satisfied by the following recursive formula:

$$P_0(x) \;=\; 1 \tag{3.10}$$

$$P_1(x) \;=\; x \tag{3.11}$$

$$P_{l+1}(x) = \frac{2l+1}{l+1}\, x\, P_l(x) - \frac{l}{l+1} P_{l-1}(x) \tag{3.12}$$

The Legendre multi-scaling bases $\phi_{l,j}^n$ are obtained by dilation and translation to the interval $[-1,1]$, followed by $L^2[-1,1]$ normalization.

$$\phi_{l,j}^n(x) \;=\; 2^{\frac{n}{2}} \sqrt{\frac{2l+1}{2}}\, P_l\left(2^n(x+1) - 2j - 1\right), \quad x \in [-1,1] \tag{3.13}$$

In Table 3.1 the scaling bases are explicitly given for $p = 0, 1, 2$ in spaces V_2^0 and V_2^1 and then they are plotted in Figure 3.3.

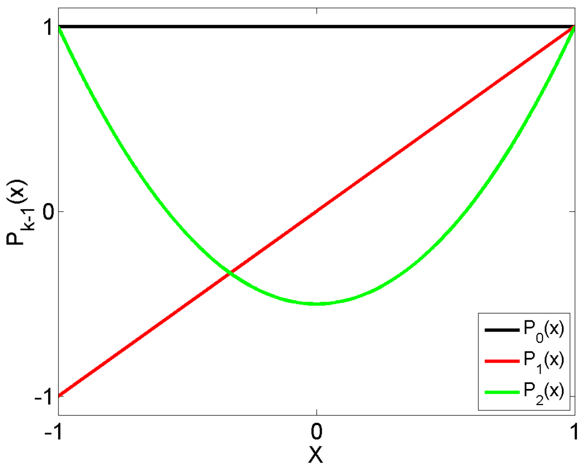


Figure 3.2: Basis of Legendre polynomial functions for V_2^0.

Table 3.1: Scaling bases for $p = 0, 1, 2$ in spaces V_2^0 and V_2^1 on $[-1, 1]$.

p	V_2^0	V_2^1	
	$x \in (-1, 1)$ $j = 0$	$x \in (-1, 0)$ $j = 0$	$x \in (0, 1)$ $j = 1$
0	$\sqrt{1/2}$	1	1
1	$\sqrt{3/2}\,x$	$\sqrt{3}\,(2x+1)$	$\sqrt{3}\,(2x-1)$
2	$\sqrt{5/8}\,(3x^2-1)$	$\sqrt{5}\,(6x^2+6x+1)$	$\sqrt{5}\,(6x^2-6x+1)$

3.3.2 Basis of wavelets

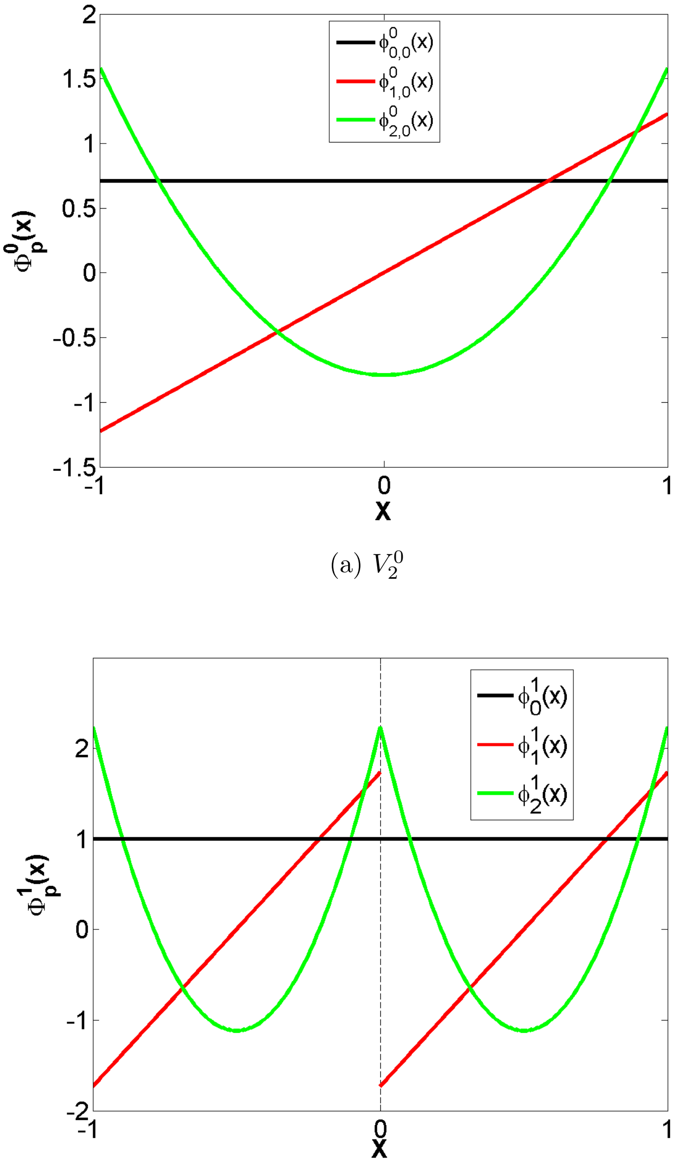
The wavelet basis functions of spanning W_k^n are defined to be the polynomial of degree $k-1$ on each of the two intervals at level $n+1$ that non-overlaps with discontinuities in the merging point. However, when $k = 1$ the Haar orthonormal wavelet family for $x \in [-1, 1]$ can be defined as follows:

$$\psi_0^n(x) = \begin{cases} 2^{\frac{n}{2}}\sqrt{\frac{1}{2}}, & x \in [\xi_1, \xi_2), \\ -2^{\frac{n}{2}}\sqrt{\frac{1}{2}}, & x \in [\xi_2, \xi_3), \\ 0 & \text{elsewhere}, \end{cases} \tag{3.14}$$

where

$\xi_1 = -1 + 2^{-n+1}j, \quad \xi_2 = -1 + 2^{-n+1}(j + \frac{1}{2}), \quad \xi_3 = -1 + 2^{-n+1}(j+1)$

The multiwavelet idea arises from the generalisation of Haar wavelets. Instead of single scaling and single wavelet function, several scaling and wavelet functions



(a) V_2^0

(b) V_2^1

Figure 3.3: The scaling bases of order $p = 0, 1, 2$; a) resolution level ($n = 0$); b) resolution level ($n = 1$).

.

are used. This leads to more degrees of freedom in the construction of multiwavelets conversely to Haar wavelets. Therefore, the properties such as high order of vanishing moments, compact support, and the orthogonality can be obtained simultaneously in multiwavelets. On the other hand, coupling multiwavelets system with high-resolution and high-accuracy approximation retains the locality of wavelet bases with discontinuous nature.

3.3.3 Construction of MW from scaling functions

Alpert's algorithm has been used for the construction of one dimensional bases $\psi_1, \psi_2, .., \psi_k$ as it appears in references (Alpert, 1993; Alpert et al., 1992) and (Shelton, 2009).

First, define the k functions g_1 , g_2 , g_k from $\mathbb{R}$ to $\mathbb{R}$, they support the interval $[-1, 1]$, and also satisfying the following properties:

1. The restriction of function g_i on the interval $(0, 1)$ is a polynomial of degree less than k.

2. The function g_i is extended to the interval $(-1, 0)$ as an odd or even function considering the following formula:

$$g_i(x) = (-1)^{i+k-1} g_i(-x) \tag{3.15}$$

in which the function $g_i(x)$ is zero outside the interval $(-1, 1)$.

3. The functions g_1 , g_2 , g_k have the following orthogonality and normality conditions:

$$\int_{-1}^{1} g_i(x) g_m(x)\, dx \equiv \langle\, g_i, g_m \rangle \;=\; \delta_{i,m}, \quad i, m = 0, 1, ..., k-1 \tag{3.16}$$

4. The function g_i has the following vanishing moments:

$$\int_{-1}^{1} g_i(x)\, \phi_m(x)\, dx \;=\; 0, \qquad m = 0, 1, .., i + k - 2 \tag{3.17}$$

Second, suppose we have $2k$ functions which span the space of polynomials of degree $(k - 1)$ on the intervals $(0, 1)$ and $(-1, 0)$. Then, we first orthogonalise k

of them to the functions ϕ_0, $\phi_1, \dots, \phi_{k-1}$, then to the functions ϕ_k, $\phi_{k+1}, \dots, \phi_{2k-1}$ and finally among themselves. The function g_i^1 can be defined as follows:

$$g_i^1(x) = \begin{cases} \phi_i(2x-1), & x \in (0,1) \\ -\phi_i(2x+1), & x \in (-1,0) \\ 0 & \text{otherwise} \end{cases} \tag{3.18}$$

Note that the $2k$ function ϕ_0, ϕ_1 , ... , ϕ_{k-1}, g_0^1 , g_2^1 , ... , g_{k-1}^1 are linearly independent. Thus, they span the space of functions, which are polynomial of degree less than k on the intervals $(0,1)$ and $(-1,0)$. Then:

1. By the Gram-Schmidt process we orthogonalise g_i^1 with taking into account the sequence ϕ_0, ϕ_1, , , , , ϕ_{k-1}, to obtain g_i^2 for $i = 1,..,k$. This orthogonality is retained by keeping orthogonalizations, which only produce linear combinations of g_i^2.

2. By considering the following steps, $k-1$ functions orthogonal to the ϕ_0, ϕ_1, ... , ϕ_{k-1} can be obtained. In which $k-2$ functions are orthogonal to ϕ_{k+1}, and so forth, down to one function that is orthogonal to ϕ_{2k-2}.

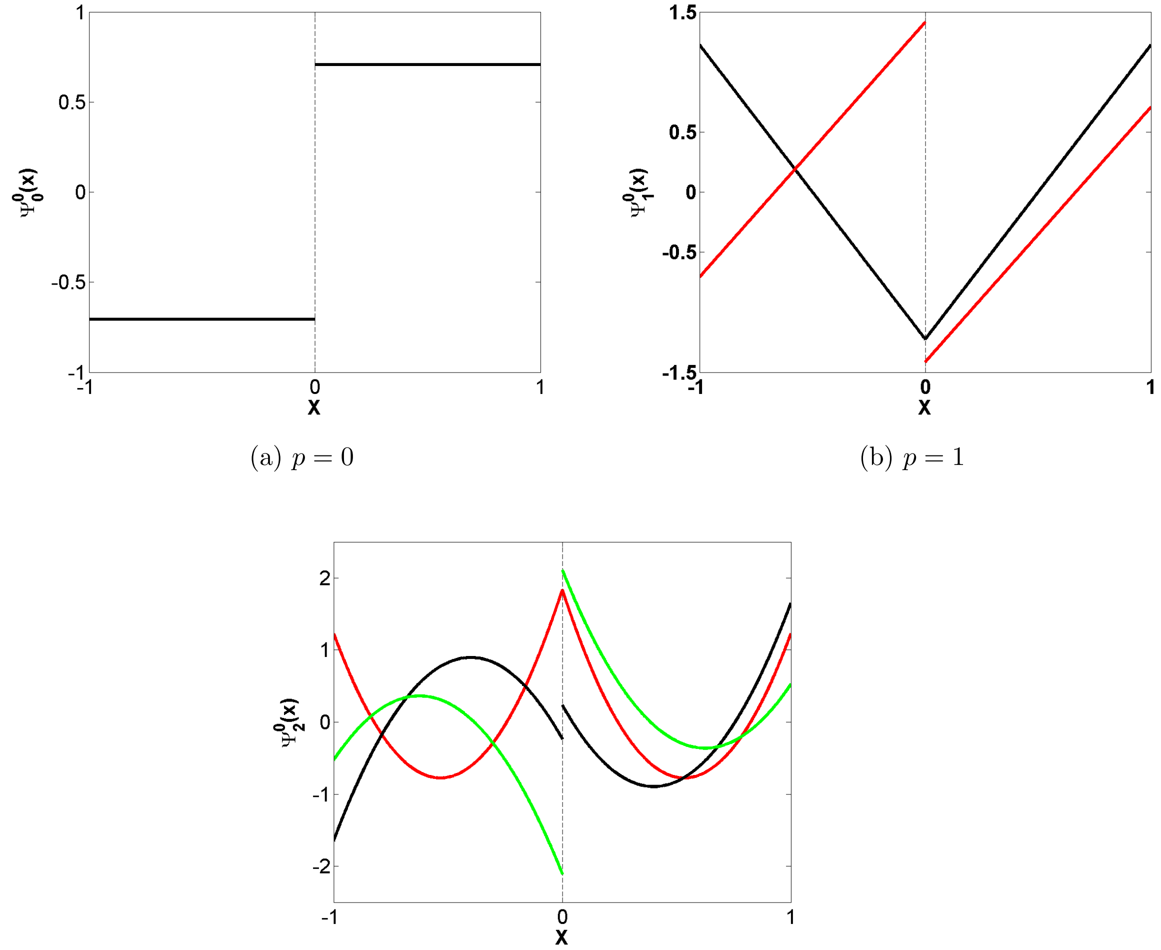
(i) First step, if at least one of the function g_i^2 is not orthogonal to ϕ_k, we have to reorder the function so that it appears first,$\langle \phi_k, g_0^2 \rangle \neq 0$.

(ii) Define $g_i^3 = g_i^2 - a_i\, g_0^2$, where a_i is chosen so $\langle \phi_k, g_i^3 \rangle = 0$ for $i = 1, .. , k-1$, satisfying the desired orthogonality to ϕ_k.

(iii) In the same way, orthogonalise to $\phi_{k+1}, \dots, \phi_{2k-2}$, each in turn, to obtain g_0^2, g_1^3, ... , g_{k-1}^{k+1} such that $\langle \phi_m, g_i^{i+2} \rangle = 0$ for $m \leqslant i + k - 1$.

3. Final step, perform Gram-Schmidt orthogonalisation on $g_{k-1}^{k+1}, g_{k-2}^{k}, \dots, g_0^2$, in that order, and normalize to obtain ψ_{k-1}, ψ_{k-2}, ..., ψ_0.

In Table 3.2, an example of multiwavelet bases are explicitly given for $k = 1, 2, 3$ in space W_k^0 and plotted in Figure 3.4.



(a) $p = 0$

(b) $p = 1$

(c) $p = 2$

Figure 3.4: Multiwavelet bases of order $p : p = k - 1$; *black, red and green* lines represent the ψ_0, ψ_1 and ψ_2 for $j = 0, 1$ respectively.

Table 3.2: Wavelets for $p = 0, 1, 2$ in space W_k^0 on $[-1, 1]$

p	$x \in (-1, 0)$	$x \in (0, 1)$
0	$-\sqrt{\frac{1}{2}}$	$\sqrt{\frac{1}{2}}$
1	$-\sqrt{\frac{3}{2}}(2x+1)$ $\sqrt{\frac{1}{2}}(3x+2)$	$\sqrt{\frac{3}{2}}(2x-1)$ $\sqrt{\frac{1}{2}}(3x-2)$
2	$-\frac{1}{3}\sqrt{\frac{1}{2}}(30x^2+24x+1)$ $\frac{1}{2}\sqrt{\frac{3}{2}}(15x^2+16x+3)$ $-\frac{1}{3}\sqrt{\frac{5}{2}}(12x^2+15x+4)$	$\frac{1}{3}\sqrt{\frac{1}{2}}(30x^2-24x+1)$ $\frac{1}{2}\sqrt{\frac{3}{2}}(15x^2-16x+3)$ $\frac{1}{3}\sqrt{\frac{5}{2}}(12x^2-15x+4)$

3.3.4 Filter matrices relations

With the multiscaling and multiwavelet bases defined, we can construct the filter matrices that achieve the transformation between any two sequence of resolution levels. The locality of this transformation is important for numerical implementation, as it leads to efficient algorithms. Basically, four type of matrices with size $(p + 1) \times (p + 1)$ are considered: two of them, H^0 and H^1, are driven from the scaling bases and called here "*lowpass filter*" matrices. While the other two matrices G_0 and G_1 are driven from inner product between the scaling and multiwavlet bases and are called " *highpass filter*" matrices.

3.3.4.1 Lowpass filter matrices

Let the vector scaling bases $\Phi_{l,j}^0 \in V_p^0$ are given, where $l \in \{0, 1, .., p\}$. Note the nested sequence of spaces V_p^0 and V_p^1, so $\Phi_{l,j}^0 \in V_p^1$. This means that the bases vector at resolution level $n = 0$ will overlap with two basis vectors at resolution level $n = 1$, therefore it is possible to write ϕ_l as follows:

$$\phi_l(x) = \sum_{r=0}^{p} \underbrace{\langle \phi_l, \phi_{r,0}^1 \rangle}_{\textcircled{A}} \phi_{r,0}^1(x) + \sum_{r=0}^{p} \underbrace{\langle \phi_l, \phi_{r,1}^1 \rangle}_{\textcircled{B}} \phi_{r,1}^1(x) \tag{3.19}$$

By using the equation 3.2, $\phi^1_{r,0}$ and $\phi^1_{r,1}$ become

$$\phi^1_{r,0}(x) = \sqrt{2}\,\phi_r(2^1(x+1)-0-1) \tag{3.20a}$$

$$\phi^1_{r,0}(x) = \sqrt{2}\,\phi_r(2x+1) \tag{3.20b}$$

$$\phi^1_{r,1}(x) = \sqrt{2}\,\phi_r(2^1(x+1)-2-1) \tag{3.21a}$$

$$\phi^1_{r,1}(x) = \sqrt{2}\,\phi_r(2x-1) \tag{3.21b}$$

Also, the *inner product* terms $\textcircled{A}$ and $\textcircled{B}$ in equation 3.19 form the so-called lowpass filer coefficients when $j = 0, 1$; $\{l, r\} \in \{0, 1, .., p\}$, and can be computed as follows

$$h^{(j)}_{l,r} = \langle \phi_l, \phi^1_{r,0} \rangle \qquad (j = 0); \tag{3.22a}$$

$$= \int_{-1}^{0} \phi_l(x)\,\sqrt{2}\,\phi_r(2x+1)\,dx \tag{3.22b}$$

$$h^{(j)}_{l,r} = \langle \phi_l, \phi^1_{r,1} \rangle \qquad (j = 1); \tag{3.23a}$$

$$= \int_{0}^{1} \phi_l(x)\,\sqrt{2}\,\phi_r(2x-1)\,dx \tag{3.23b}$$

Consider $p = 0, 1, 2$; $j = 0, 1$, then by gathering all coefficients that obtained from equation 3.22, the $H^{(j)}$ matrices that associated with the chosen Legendre polynomial order are given as follows:

For $l = 0$ and $r = 0$;

$$H^0 = \begin{pmatrix} \langle \phi^0_{0,0}, \phi^1_{0,0} \rangle \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} \end{pmatrix} \quad H^1 = \begin{pmatrix} \langle \phi^0_{0,0}, \phi^1_{0,1} \rangle \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} \end{pmatrix}$$

For $l = 1$ and $r = 1$;

$$H^0 = \begin{pmatrix} \langle \phi^0_{0,0}, \phi^1_{0,0} \rangle & \langle \phi^0_{0,0}, \phi^1_{1,0} \rangle \\ \langle \phi^0_{1,0}, \phi^1_{0,0} \rangle & \langle \phi^0_{1,0}, \phi^1_{1,0} \rangle \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{6}}{4} & \frac{\sqrt{2}}{4} \end{pmatrix}$$

$$H^1 = \begin{pmatrix} \langle \phi^0_{0,0}, \phi^1_{0,1} \rangle & \langle \phi^0_{0,0}, \phi^1_{1,1} \rangle \\ \langle \phi^0_{1,0}, \phi^1_{0,1} \rangle & \langle \phi^0_{1,0}, \phi^1_{1,1} \rangle \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{6}}{4} & \frac{\sqrt{2}}{4} \end{pmatrix}$$

For $l = 2$ and $r = 2$;

$$H^0 = \begin{pmatrix} \langle \phi^0_{0,0}, \phi^1_{0,0} \rangle & \langle \phi^0_{0,0}, \phi^1_{1,0} \rangle & \langle \phi^0_{0,0}, \phi^1_{2,0} \rangle \\ \langle \phi^0_{1,0}, \phi^1_{0,0} \rangle & \langle \phi^0_{1,0}, \phi^1_{1,0} \rangle & \langle \phi^0_{1,0}, \phi^1_{2,0} \rangle \\ \langle \phi^0_{2,0}, \phi^1_{0,0} \rangle & \langle \phi^0_{2,0}, \phi^1_{1,0} \rangle & \langle \phi^0_{2,0}, \phi^1_{2,0} \rangle \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} & 0 & 0 \\ -\frac{\sqrt{6}}{4} & \frac{\sqrt{2}}{4} & 0 \\ 0 & -\frac{\sqrt{30}}{8} & \frac{\sqrt{2}}{8} \end{pmatrix}$$

$$H^1 = \begin{pmatrix} \langle \phi^0_{0,0}, \phi^1_{0,1} \rangle & \langle \phi^0_{0,0}, \phi^1_{1,1} \rangle & \langle \phi^0_{0,0}, \phi^1_{2,1} \rangle \\ \langle \phi^0_{1,0}, \phi^1_{0,1} \rangle & \langle \phi^0_{1,0}, \phi^1_{1,1} \rangle & \langle \phi^0_{1,0}, \phi^1_{2,1} \rangle \\ \langle \phi^0_{2,0}, \phi^1_{0,1} \rangle & \langle \phi^0_{2,0}, \phi^1_{1,1} \rangle & \langle \phi^0_{2,0}, \phi^1_{2,1} \rangle \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} & 0 & 0 \\ \frac{\sqrt{6}}{4} & \frac{\sqrt{2}}{4} & 0 \\ 0 & \frac{\sqrt{30}}{8} & \frac{\sqrt{2}}{8} \end{pmatrix}$$

By considering all definitions given above, equation 3.19 can be generalised to obtain any scaling basis functions of V^n_P from their basis functions which are expanded in V^{n+1}_p

$$\phi^n_{l,j}(x) = \sum_{m=0}^{2^n-1} \sum_{r=0}^{p} H^{(m)}_{l,r}\, \phi^{n+1}_{r,m}(x) \tag{3.24}$$

3.3.4.2 Highpass filter matrices

The same approach has been considered for multiwavelets bases with respect to W^0_p. Let $\Psi^0_{l,j} \in W^0_p$ are given, $l \in \{0, 1, .., p\}$. Since $W^0_p \in V^1_p$ (see equation 3.7) so $\psi_l(x)$ can be represented as follows:

$$\psi_l(x) = \sum_{r=0}^{p} \underbrace{\langle \psi_l, \phi^1_{r,0} \rangle}_{\textcircled{C}} \phi^1_{r,0}(x) + \sum_{r=0}^{p} \underbrace{\langle \psi_l, \phi^1_{r,1} \rangle}_{\textcircled{D}} \phi^1_{r,1}(x) \tag{3.25}$$

Also, the *inner product* terms $\textcircled{C}$ and $\textcircled{D}$ in equation 3.25 form the so-called highpass filer coefficients when $j = 0, 1$; $\{l, r\} \in \{0, 1, .., p\}$, and can be computed as follows:

$$g^{(j)}_{l,r} = \langle \psi_l, \phi^1_{r,0} \rangle \qquad (j = 0); \tag{3.26a}$$

$$= \int_{-1}^{0} \phi^1_{l,j}(x)\, \sqrt{2}\, \psi_r(2x + 1)\, dx \tag{3.26b}$$

$$g^{(j)}_{l,r} = \langle \psi_l, \phi^1_{r,1} \rangle \qquad (j = 1); \tag{3.27a}$$

$$= \int_{0}^{1} \phi^1_{l,j}(x)\, \sqrt{2}\, \psi_r(2x - 1)\, dx \tag{3.27b}$$

Consider $p = 0, 1, 2$; $j = 0, 1$, then by gathering all coefficients that obtained from equation 3.26. The $G^{(j)}$ matrices that associated with the chosen Legendre

polynomial order, are given as follows:
For $l = 0$ and $r = 0$;

$$G^0 = \begin{pmatrix} \langle \psi^1_{0,0}, \phi^1_{0,0} \rangle \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{2} \end{pmatrix} \quad G^1 = \begin{pmatrix} \langle \psi^1_{0,0}, \phi^1_{0,1} \rangle \end{pmatrix} = \begin{pmatrix} -\frac{\sqrt{2}}{2} \end{pmatrix}$$

For $l = 1$ and $r = 1$;

$$G^0 = \begin{pmatrix} \langle \psi^1_{0,0}, \phi^1_{0,0} \rangle & \langle \psi^1_{0,0}, \phi^1_{1,0} \rangle \\ \langle \psi^1_{1,0}, \phi^1_{0,0} \rangle & \langle \psi^1_{1,0}, \phi^1_{1,0} \rangle \end{pmatrix} = \begin{pmatrix} 0 & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{4} & \frac{\sqrt{6}}{4} \end{pmatrix}$$

$$G^1 = \begin{pmatrix} \langle \psi^1_{0,1}, \phi^1_{0,1} \rangle & \langle \psi^1_{0,1}, \phi^1_{1,1} \rangle \\ \langle \psi^1_{1,1}, \phi^1_{0,1} \rangle & \langle \psi^1_{1,1}, \phi^1_{1,1} \rangle \end{pmatrix} = \begin{pmatrix} 0 & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{4} & \frac{\sqrt{6}}{4} \end{pmatrix}$$

For $l = 2$ and $r = 2$;

$$G^0 = \begin{pmatrix} \langle \psi^1_{0,0}, \phi^1_{0,0} \rangle & \langle \psi^1_{0,0}, \phi^1_{1,0} \rangle & \langle \psi^1_{0,0}, \phi^1_{2,0} \rangle \\ \langle \psi^1_{1,0}, \phi^1_{0,0} \rangle & \langle \psi^1_{1,0}, \phi^1_{1,0} \rangle & \langle \psi^1_{1,0}, \phi^1_{2,0} \rangle \\ \langle \psi^1_{2,0}, \phi^1_{0,0} \rangle & \langle \psi^1_{2,0}, \phi^1_{1,0} \rangle & \langle \psi^1_{2,0}, \phi^1_{2,0} \rangle \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{2}}{6} & \frac{\sqrt{6}}{6} & -\frac{\sqrt{10}}{6} \\ 0 & \frac{\sqrt{2}}{8} & \frac{\sqrt{30}}{8} \\ -\frac{\sqrt{10}}{12} & -\frac{\sqrt{30}}{12} & -\frac{\sqrt{2}}{3} \end{pmatrix}$$

$$G^1 = \begin{pmatrix} \langle \psi^1_{0,1}, \phi^1_{0,1} \rangle & \langle \psi^1_{0,1}, \phi^1_{1,1} \rangle & \langle \psi^1_{0,1}, \phi^1_{2,1} \rangle \\ \langle \psi^1_{1,1}, \phi^1_{0,1} \rangle & \langle \psi^1_{1,1}, \phi^1_{1,1} \rangle & \langle \psi^1_{1,1}, \phi^1_{2,1} \rangle \\ \langle \psi^1_{2,1}, \phi^1_{0,1} \rangle & \langle \psi^1_{2,1}, \phi^1_{1,1} \rangle & \langle \psi^1_{2,1}, \phi^1_{2,1} \rangle \end{pmatrix} = \begin{pmatrix} -\frac{\sqrt{2}}{6} & \frac{\sqrt{6}}{6} & \frac{\sqrt{10}}{6} \\ 0 & -\frac{\sqrt{2}}{8} & \frac{\sqrt{30}}{8} \\ \frac{\sqrt{10}}{12} & -\frac{\sqrt{30}}{12} & \frac{\sqrt{2}}{3} \end{pmatrix}$$

By considering all definitions given above and analogously to equation 3.24, the multiwavelet bases vector can be expanded in basis of space V_p^{n+1}

$$\psi^n_{l,j}(x) = \sum_{m=0}^{2^n-1} \sum_{r=0}^{p} G^{(m)}_{l,r} \, \phi^{n+1}_{r,m}(x) \tag{3.28}$$

3.3.4.3 The combination of High-Low pass filter matrices

With the filter matrices defined, we can perform the two-scale transformation, which is local and important for numerical implementation; as it leads to efficient algorithms. Here, some important properties of filter matrices are given after collecting the both types of filters to obtain a matrix transformation (U), which is orthogonal and can describe a *linear unitary* transformation between the two sets of bases.

$$U = \begin{pmatrix} H^0 & H^1 \\ G^0 & G^1 \end{pmatrix} \tag{3.29}$$

Since $UU^T = I$, where I is the *identity matrix* and ^{T} is the transpose matrix index. we can satisfy $U^T = U^{-1}$. This condition introduces an additional set of relations (Alpert et al., 2002):

$$H^0\, H^{0^T} + H^1\, H^{1^T} = I, \tag{3.30a}$$

$$G^0\, G^{0^T} + G^1\, G^{1^T} = I, \tag{3.30b}$$

$$H^0\, G^{0^T} + H^1\, G^{1^T} = 0, \tag{3.30c}$$

$$G^0\, H^{0^T} + G^1\, H^{1^T} = 0, \tag{3.30d}$$

Consequently, equations 3.24 and 3.28 can be reduced to

$$\begin{pmatrix} \phi^n_{l,j} \\ \psi^n_{l,j} \end{pmatrix} = \begin{pmatrix} H^0 & H^1 \\ G^0 & G^1 \end{pmatrix} \begin{pmatrix} \phi^{n+1}_{l,2j} \\ \phi^{n+1}_{l,2j+1} \end{pmatrix} \tag{3.31}$$

The transformation in equation 3.31 is called bases decomposition, while its inverse is called the 'bases reconstruction'.

3.4 Function representation

The multiwavelets formalism presented in section 3.3 gives prospects of efficient representation of any arbitrary function, and in this section we describe how this is achieved in practice by considering an example function such as $f(x) = \sin(2\pi x)$; $x \in (-1, 1)$.

3.4.1 Single scale representation

The orthogonal projection of function $\mathbf{P}^n_p\, f(x) = f(x)$ in V^n_p with respect to the scaling bases $\{\phi^n_{l,j}\}$ is given by:

$$P^n_p\, f(x) = \sum_{j=0}^{2^n-1} \sum_{l=0}^{p} s^n_{l,j}\, \phi^n_{l,j}(x) \tag{3.32}$$

where the expansion coefficients $s^n_{l,j}$ indicate the location of $f(x)$ at which resolution level, the so-called "scaling coefficients". It can be computed (or initialised) as:

$$s^n_{l,j}(i) = \langle f, \phi^n_{l,j} \rangle = \int_{-1+2^{-n+1}(j)}^{-1+2^{-n+1}(j+1)} f(x)\, \phi^n_{l,j}(x)\, dx \tag{3.33a}$$

$$= 2^{\frac{n}{2}} \int_{-1+2^{-n+1}(j)}^{-1+2^{-n+1}(j+1)} f(x)\, \phi_l(2^n(x+1) - 2j - 1)\, dx \tag{3.33b}$$

To apply the standard Gauss-Legendre quadrature integration (see appendix A), the integration limitations in equation 3.33 are replaced by (-1,1) using the equation 3.34. Therefore, the i^{th} Gauss points ξ_{g_i}, the root of Legendre polynomial order, and its weight ω_{g_i} can be straightforwardly used.

$$\xi = 2^n(x+1) - 2j - 1, \tag{3.34a}$$

$$x = 2^{-n}(\xi + 2j + 1) - 1, \tag{3.34b}$$

$$dx = 2^{-n}\,\xi\, d\xi \tag{3.34c}$$

By substituting equation 3.34 in equation 3.33 with consideration for the Gauss points, their weighted functions yield

$$s^n_{l,j}(i) \;= 2^{-\frac{n}{2}} \int_{-1}^{1} f(2^{-n}(\xi + 2j + 1) - 1)\, \phi^n_{l,j}(\xi)\, d\xi \tag{3.35a}$$

$$s^n_{l,j}(i) \;= 2^{-\frac{n}{2}} \sum_{l=0}^{p} \omega_{g^l_i} f(2^{-n}(\xi_{g^l_i} + 2j + 1) - 1)\, \phi_l(\xi_{g^l_i})\, d\xi \tag{3.35b}$$

where the index i indicates to the local cell within the discrete domain $[-1, 1]$. The accuracy of this approximation is dependent on to which resolution level the projection is performed, and to the order p of polynomial basis.

3.4.2 Multi-scale representation

The function $f(x)$ also has the projection operator $\mathbf{Q}^n_p$ that projects in W^n_p with respect to the multiwavelet bases $\{\psi^n_{l,j}\}$.

$$\mathbf{Q}^n_p\, f(x) = \sum_{j=0}^{2^n-1} \sum_{l=0}^{p} d^n_{l,j}\, \psi^n_{l,j}(x) \tag{3.36}$$

For which the detail coefficients are given as

$$d^n_{l,j} \;= \int_{-1+2^{-n+1}(j)}^{-1+2^{-n+1}(j+1)} f(x)\, \psi^n_{l,j}(x)\, dx \tag{3.37}$$

According to equation 3.4 the following relation can be obtained between the projection operators

$$\mathbf{P}^{n+1}_p f(x) \;=\; \mathbf{P}^n_p f(x) + \mathbf{Q}^n_p f(x) \tag{3.38}$$

which means that the multiwavelets projection should not be considered as an approximation of the function $f(x)$, but only the difference between two approximations.

$$\mathbf{Q}^n_p f(x) = (\mathbf{P}^{n+1}_p - \mathbf{P}^n_p) f(x) = f(x)^{n+1} - f(x)^n \tag{3.39}$$

This means that the multiwavelets projection can be used as a measure of the accuracy of projection $f^n(x)$ at level n. It holds an individual fluctuation of the solution, which allows the escalation of $f(x)$ to the level $n+1$ of resolution once it's added to the projection function at level n (Gerhard et al., 2015). By applying the equation 3.39 recursively, an approximation of $f^N(x)$ in the finest level (N) can be expressed as the approximation solution on the coarsest level (i.e. $n = 0$) with a number of encoded detail information.

$$\mathbf{P}_p^N f(x) = \mathbf{P}_p^0 f(x) + \sum_{n=0}^{N-1} Q_p^n f(x) \tag{3.40a}$$

$$= \sum_{l=0}^{p} s_{l,0}^0 \phi_{l,0}^0 + \sum_{n=0}^{N-1} \sum_{j=0}^{2^n-1} \sum_{l=0}^{p} d_{l,j}^n \, \psi_{l,j}^n(x) \tag{3.40b}$$

Usually we need all the entries of $s_{l,0}^0$ and $d_{l,j}^n$ to obtain an accurate representation of f^N. However, many entries of $d_{l,j}^n$ are small and replacing them with zero will not directly affect the approximating solution. Also the function still permits a sufficient accuracy. Therefore, to take advantage of both representations, we need an algorithm to convert one method into another and this is achieved by using the high and low pass filter matrices.

3.4.3 Application for a function $\sin(2\pi x)$

In this section, an example $u(x)$ =sin$(2\pi x)$; $x \in (-1, 1)$ is presented to prove how it is straightforward to decompose and reconstruct $u(x)$ across different resolution levels. The domain $[-1, 1]$ is divided into 8 uniform and non-overlapping cells. The cell size at resolution $n = 0$ is 0.25.

In the algorithm 1, the decomposition of $u(x)$ over each cell is presented. Lines 2 and 3 of Algorithm 1 give the single scale projection at the highest levels (here chosen to be $n = 2$) with considering the Legendre polynomial $p = 2$. The approximate solutions over each cell projected into V_p^2 (algorithm 1, line 4) has been shown in Figure 3.5, in which the accuracy of the approximation decreases as the scaling coefficients associated with the polynomial order are excluded from the construction of the solution. In Figure 3.6, the projection of $u(x)$ into V_p^0 is given, in which its scaling coefficients are obtained from the algorithm 1, line 13. It is noted that the accuracy of the solution is locally preserved and transmitted from level $n = 2$ to level $n = 0$ with considering the associated p value that is considered in building the local approximate solution at level $n = 2$. Furthermore, the flow information between the resolution levels are accessible, one can manipulate them to change the accuracy of approximate solution and the resolution level independently over each cell, as shown in Figure 3.7 as well as the projection that each cell belonged is given in Table 3.3. On the other hand, by following steps in the algorithm 2, the function $u(x)$ can be reconstructed along with preserving the initial accuracy of function.

It can be concluded from the presented example, that the local solution-based criterion can be established by omitting the details coefficient from the construction of local solution. This concept gives an optimistic mechanism to be employed within the numerical scheme to introduce an adaptive scheme. Hence, this will be our aim to incorporate this technique within the FV Godunove and DG formulation. More details will be given in the following chapters.

Table 3.3: The projection of $\sin(2\pi x)$ over each cell

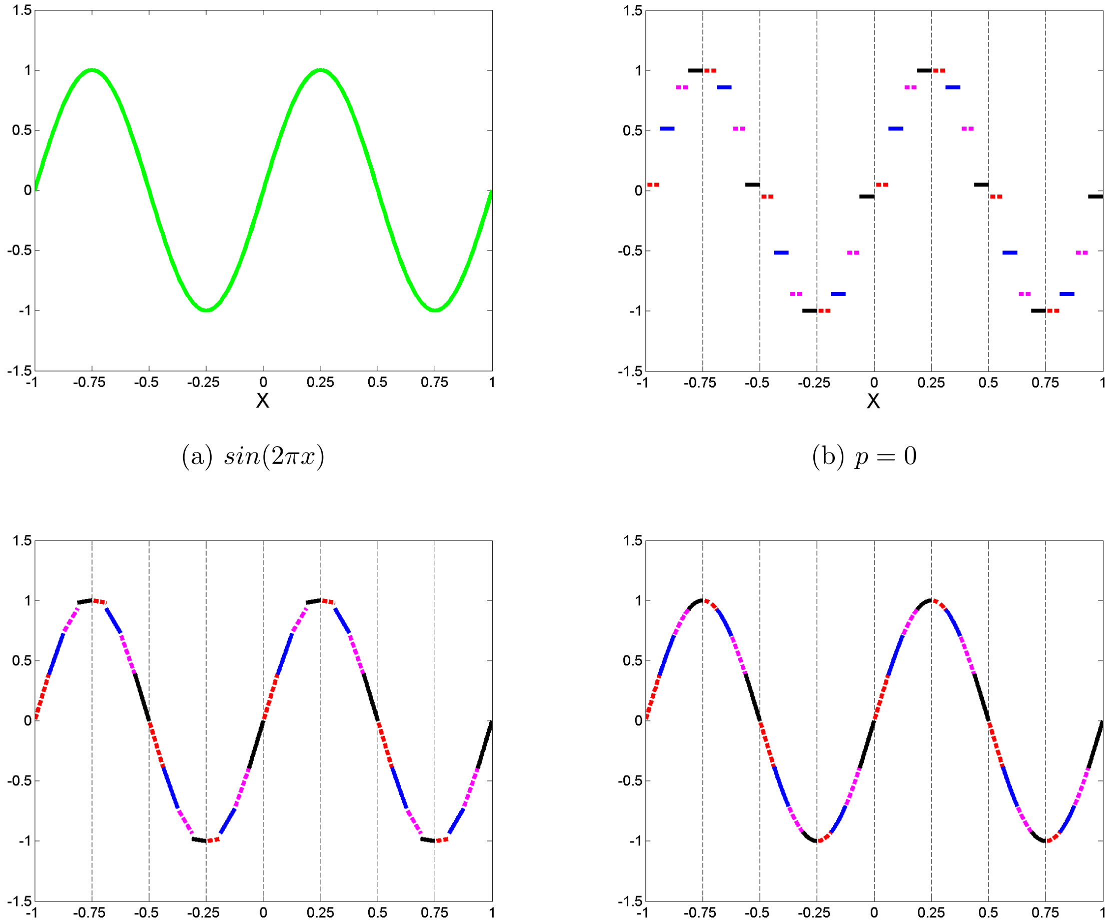
cell	space (V_p^n)	cell	space (V_p^n)
1	V_0^0	5	V_1^0
2	V_1^1	6	V_0^2
3	V_2^2	7	V_1^2
4	V_0^2	8	V_2^1

Algorithm 1 Decomposition of $sin(2\pi x)$, into V_2^0

```
1:  for all Cells do
2:      Use Eq. 3.2 to obtain the scaling bases $\phi_{0,j}^2$,$\phi_{1,j}^2$,$\phi_{2,j}^2$ in $V_2^2$ from the scaling bases in $V_2^0$ given in table 3.1.
3:      Compute $s_{0,j}^2$, $s_{1,j}^2$, $s_{2,j}^2$; $j = 0, 1, 2, 3$, from Eq. 3.33.
4:      Perform the projection of $u(x)$ in $V_2^2$ using the Eq. 3.32.
5:      Use transformation matrix 3.29 to decompose $s_{0,j}^2$, $s_{1,j}^2$, $s_{2,j}^2$ into $s_{0,j}^1$, $s_{1,j}^1$, $s_{2,j}^1$ and $d_{0,j}^1$, $d_{1,j}^1$, $d_{2,j}^1$.
6:      Encode the detail coefficients $d_{0,j}^1$, $d_{1,j}^1$, $d_{2,j}^1$.
7:      Determine $\phi_{0,j}^1$, $\phi_{1,j}^1$, $\phi_{2,j}^1$ and $\psi_{0,j}^1$,$\psi_{1,j}^1$,$\psi_{2,j}^1$ using Eq. 3.31.
8:      Encode $\psi_{0,j}^1$, $\psi_{1,j}^1$, $\psi_{2,j}^1$.
9:      Re-use transformation matrix 3.29 to decompose $s_{0,j}^1$, $s_{1,j}^1$, $s_{2,j}^1$ into $s_{0,j}^0$, $s_{1,j}^0$, $s_{2,j}^0$ and $d_{0,j}^0$, $d_{1,j}^0$, $d_{2,j}^0$.
10:     Encode the detail coefficients $d_{0,j}^0$, $d_{1,j}^0$, $d_{2,j}^0$.
11:     Determine $\phi_{0,j}^0$, $\phi_{1,j}^0$, $\phi_{2,j}^0$ and $\psi_{0,j}^0$, $\psi_{1,j}^0$, $\psi_{2,j}^0$ using Eq. 3.31..
12:     Encode $\psi_{0,j}^0$, $\psi_{1,j}^0$, $\psi_{2,j}^0$.
13:     Perform the projection of $u(x)$ in $V_2^0$ using $s_{0,j}^0$, $s_{1,j}^0$, $s_{2,j}^0$ and the scaling bases in $V_2^0$ see table 3.1; then substitute them in Eq. 3.32.
14: end for
```

Algorithm 2 Reconstruction of $sin(2\pi x)$, upto V_2^2

```
1:  for all Cells do
2:      Use the inverse of transformation matrix 3.29 to construct $s_{0,j}^1$, $s_{1,j}^1$, $s_{2,j}^1$ from $s_{0,j}^0$, $s_{1,j}^0$, $s_{2,j}^0$ and $d_{0,j}^0$, $d_{1,j}^0$, $d_{2,j}^0$ ( Algorithm 1,line 9).
3:      Perform the inverse of Eq.3.31 to obtain $\phi_{0,j}^1$, $\phi_{1,j}^1$, $\phi_{2,j}^1$ from the bases in Algorithm 1,line 11.
4:      Re-Use the inverse of transformation matrix 3.29 to construct $s_{0,j}^2$, $s_{1,j}^2$, $s_{2,j}^2$ from $s_{0,j}^1$, $s_{1,j}^1$, $s_{2,j}^1$ and $d_{0,j}^1$, $d_{1,j}^1$, $d_{2,j}^1$ ( Algorithm 1, line 5).
5:      Perform the inverse of Eq. 3.31 to obtain $\phi_{0,j}^2$, $\phi_{1,j}^2$, $\phi_{2,j}^2$ from the bases in Algorithm 1, line 7.
6:      Perform the projection of $u(x)$ in $V_2^2$ from lines 4 and 5 using Eq. 3.32.
7:  end for
```



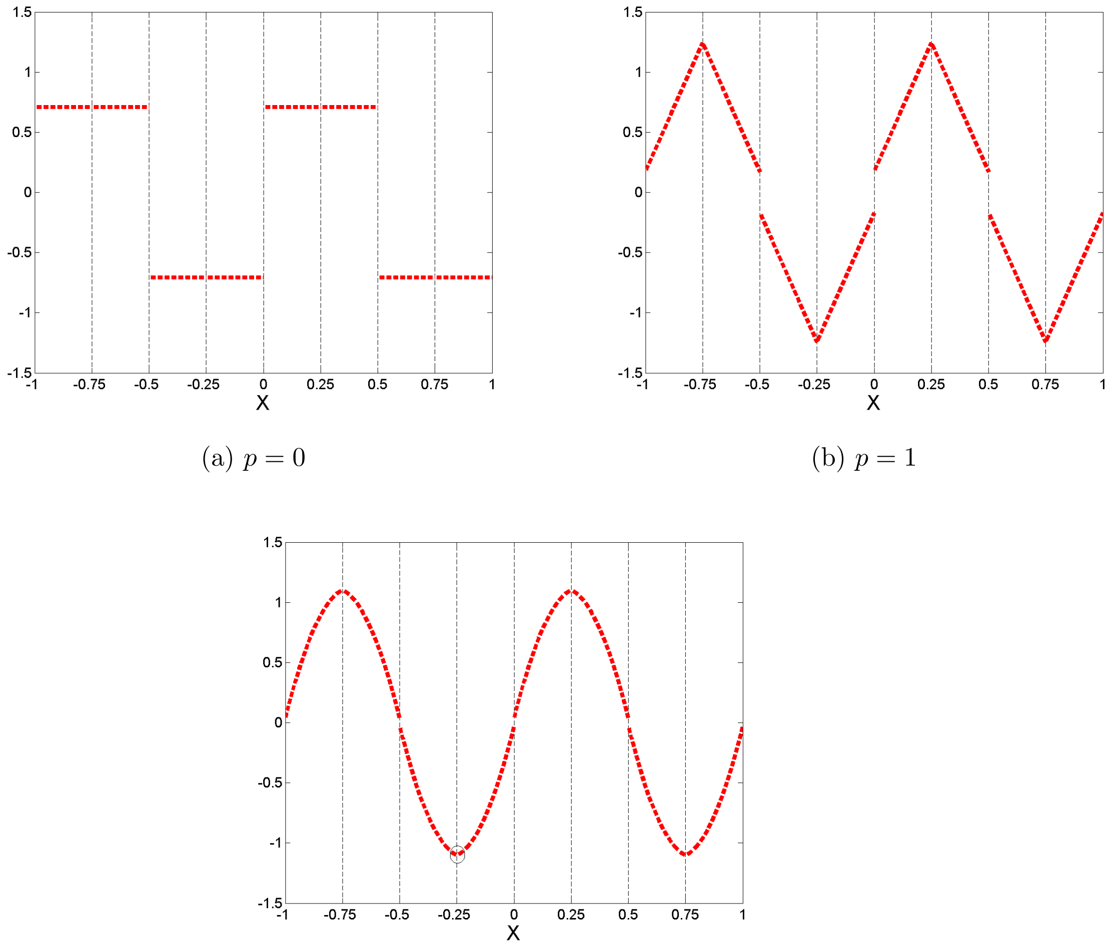
(a) $sin(2\pi x)$

(b) $p = 0$

(c) $p = 1$

(d) $p = 2$

Figure 3.5: The approximation of $u(x)$ =sin$(2\pi x)$ into V_p^2.



(a) $p = 0$

(b) $p = 1$

(c) $p = 2$

Figure 3.6: The approximation of $u(x) = \sin(2\pi x)$ into V_p^0.

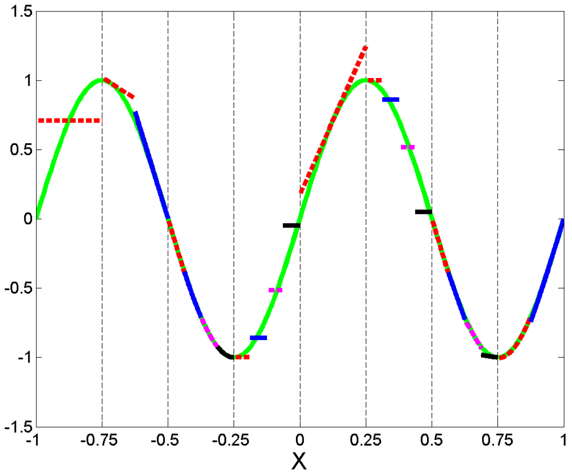


Figure 3.7: The approximate solution of $\sin(2\pi x)$ considering different resolution levels and accuracy orders compared with the exact solution.

Chapter 4

Numerical Methods

This chapter consists of two parts. In the first part, the general procedures of the finite volume method and the discontinuous Galerkin method for the SWE are presented. Before delving into these methods, a brief review of the mathematical model of shallow water flow is given and how its principles can be interpreted within the context of open channel flow. The second part presents how we can exploit the useful features of Haar wavelets and multiwavelets when they are incorporated into the aforementioned numerical schemes to obtain an adaptive version of these methods. The presentation and discussion herein is based on the following sources (Henderson, 1966; Cunge et al., 1980; Roe and Pike, 1985; Toro, 2001, 2009, 2012; Glaister, 1988; Cockburn and Shu, 2001, 1998; Cockburn, 1998, 1999; Osher, 1984; Quirk, 1994; Kesserwani and Liang, 2012b; Kesserwani, 2013; Hirsch, 2002; Roe, 1981; Hovhannisyan et al., 2014; Kesserwani et al., 2015)

Part I: Non-adaptive numerical schemes

4.1 Mathematical model

A number of mathematical models for describing the shallow water flow have been developed, In the case of one dimensional flow, the Saint Venant equations that given in chapter is the most commonly used for solving open channel low problems. By gathering equations 2.27 and 2.28, the conservation matrix form for 1D SWE can be obtained as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = \mathbf{S} \tag{4.1}$$

$$\mathbf{U} = \begin{bmatrix} h \\ q \end{bmatrix} \qquad \mathbf{F} = \begin{bmatrix} q \\ gh^2/2 + q^2/h \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} 0 \\ -gh(S_0 - S_f) \end{bmatrix} \tag{4.2}$$

where t is the time (s), x is space (m) and $\mathbf{U}$, $\mathbf{F}$ and $\mathbf{S}$ are the vectors containing the conserved variables, the fluxes and the bed source terms, respectively, in which h is the water depth (m), q is the flow rate per unit width $(m^3/s.m)$,g is the acceleration gravity (m/s^2) and z is the bed elevation (m). For a rectangular channel, S_f, in terms of Manning's equations and S_0 have the following expressions:

$$S_f = \frac{n^2|q|q}{h^2 R^{4/3}} \tag{4.3}$$

$$S_0 = \partial_x z \tag{4.4}$$

4.2 Finite volume framework

The finite volume methods are based on writing equation 4.1 in integral form. By considering the homogeneous scalar conservation law that is valid for any arbitrary closed volume on the Cartesian mesh, the equation 4.1 can be represented by

$$\int_\Omega \frac{\partial \mathbf{U}}{\partial t} \, d\Omega + \int_\Omega \frac{\partial \mathbf{F}}{\partial x} \, d\Omega = 0 \tag{4.5}$$

In order to obtain a numerical solution, each of the closed volume is represented by a cell i of the mesh, which used for the discretisation of simulated domain, see Figure 4.1. Thus $\Omega_i = [x_{i-1/2}, x_{i+1/2}]$ of size Δx. Integrating 4.5 over a control volume $[x_{i-1/2}, x_{i+1/2}] \times [t^n, t^{n+1}]$ where $t^{n+1} = t^n + \Delta x$. It yields the following equation.

$$\begin{aligned} &\int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}(x, t^{n+1}) dx - \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}(x, t^n) dx + \\ &\int_{t^n}^{t^{n+1}} \mathbf{F}[\mathbf{U}(x_{i+1/2}, t)] dt - \int_{t^n}^{t^{n+1}} \mathbf{F}[\mathbf{U}(x_{i+1/2}, t)] dt = 0 \end{aligned} \tag{4.6}$$

This form ensures that the approximate solution of the finite volume is conservative within each cell which is important when attempting to accurately resolve the shock waves.

Here, the discrete values of $\mathbf{U}$ are taken to be the cell average value at time t^n, defined as

$$\mathbf{U}_i^n = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}(x,t^n)dx \tag{4.7}$$

Similarly, at the time t^{n+1}

$$\mathbf{U}_i^{n+1} = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}(x,t^{n+1})dx \tag{4.8}$$

Also, the time average fluxes $\mathbf{F}[\mathbf{U}(x,t)]$ across the cell interfaces $x = x_{i\mp1/2}$ can by defined as

$$\mathbf{F}_{i\mp1/2}^n = \frac{1}{\Delta x}\int_{t^n}^{t^{n+1}} \mathbf{F}[\mathbf{U}(x_{i\mp1/2},t)]dt \tag{4.9}$$

Consequently, the explicit numerical algorithm of 4.5 for the updating of a single cell i from t^n to t^{n+1} can be written generally by means of

$$\mathbf{U}_i^{t^{n+1}} = \mathbf{U}_i^{t^n} + \frac{\Delta t}{\Delta x}[\tilde{\mathbf{F}}_{i+1/2} - \tilde{\mathbf{F}}_{i-1/2}] \tag{4.10}$$

where $\tilde{\mathbf{F}}_{i\mp1/2}$ are numerical fluxes, which represent the numerical approximation of the exact flux integral that across the cell interfaces. However, at the interfaces, the discrete values of $\mathbf{U}_i$ manifest a jump, therefore a sequence of Riemann problems is generated and their construction solution leads to several expressions for numerical fluxes. Substituting these fluxes in equation 4.10 will lead to different numerical methods, which are described in the literature.

In this thesis, a Godunov-type method is reformulated to create an adaptive multiresolution scheme therefore our discussion employs a Godunov-type method in general form, and its reformulation is given in Part II.

A Godunov-type method is a first order *upwind* method that provides a best solution to the discontinuity problem. It satisfies the entropy conditions, which means all shock waves are physically correct. The term *upwind* refers to the flow direction in which the upstream values is used to evaluate the property on the cell interfaces and then use the evaluation values to compute the value at the centre of the cell. The origin Godunov method can be generalized in three steps. The first step is to assume that the numerical solution ($\mathbf{U}_i^n$) of the dependent variables within each cell i is *piece-wise* constant, instead of considering $\mathbf{U}_i^n$ as a set of point values (see Figure 4.2). Therefore, if the averages of the equations 4.7 and 4.9 hold; the $\mathbf{U}_i^n$ satisfies the integral form of the conservation law 4.6 exactly. The second step is to compute the numerical fluxes $\tilde{\mathbf{F}}_{i\mp 1/2}$ via using the exact solution of local Riemann problems at the cell interfaces (see Figure 4.3). A Riemann problem consists of solving the system of SWE 4.1 (regardless the source term vector) for a single jump discontinuity, such as at $x_{i+1/2}$, together with piecewise initial data state that has the following form:

$$\mathbf{U}(x,t^n) = \begin{cases} \mathbf{U}_i^n & x < x_{i+1/2} \\ \mathbf{U}_{i+1}^n & x > x_{i+1/2} \end{cases} \tag{4.11}$$

The Riemann solution with the right data state $\mathbf{U}_{i+1}^n$ and the left data state $\mathbf{U}_i^n$ is denoted by $\mathbf{U}_{i+1/2}^n$. Similarly the $\mathbf{U}_{i-1/2}^n$ is computed from the right data state $\mathbf{U}_i^n$ and the left data state $\mathbf{U}_{i-1}^n$ at point $x_{i-1/2}$. The last step is linking these Riemann solutions together on such a way that the waves belong to the neighbouring cells are not interacting.To achieve this, for any Δx one has to keep the time step Δt within bounds, usually achieved by imposing the Courant-Friedrichs-Lewys (CFL) criterion ($0 < \text{CFL} \leq 1.0$). By performing the equation 4.10 , the updated numerical solution values over cell i is obtained.

For the system of SWE, solving the Riemann problem exactly often requires extensive computational time. Therefore the exact solution of the Riemann problems can be replaced by the approximate Riemann solvers with no significant effect in practice. This is due to the third step in which the information that is driven from the exact solution is partially lost. Generally, two approaches are used for the approximation of Riemann solvers. The first approach is to approximate the Riemann state and apply the physical flux, while the second approach is to directly approximate the numerical flux. In this thesis, we will use the Roe Riemann solver as a numerical flux function.

4.2.1 Roe Riemann solver

In this subsection, the presentation of the Roe Riemann solver for the Euler equations is given together with the details of the application method to open channel flow. The Roe solver is considered one of the simplest and most used Riemann solvers because it employs a local linearisation of the system of conservation law written in quasi-liner form.

$$\frac{\partial \mathbf{U}}{\partial t} + J\frac{\partial \mathbf{U}}{\partial x} = 0 \tag{4.12}$$

Figure removed due to copyrights

Figure 4.1: The control volume of cell i with associated interface fluxes.

.

Figure removed due to copyrights

Figure 4.2: The piecewise constant representation of data at t^n.

Figure removed due to copyrights

Figure 4.3: The sequence of Riemann problems and their averaging.

where J is the Jacobian matrix given by

$$J = \frac{\partial \boldsymbol{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 \\ gh - q^2/h^2 & 2q/h \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ c^2 - u^2 & 2u \end{bmatrix} \tag{4.13}$$

and $u = q/h$ is the average water velocity within the cross-sectional area of the channel and $c = \sqrt{gh}$ is the celerity of the small amplitude water surface wave.

Roe's technique linearises the system 4.12 by replacing the Jacobian matrix in each interval such as (x_i, x_{i+1}), which represents the original Riemann problems, with a matrix $\tilde{J} = \tilde{J}(\mathbf{U}_i, \mathbf{U}_{i+1})$ that represents an approximate Riemann problem with initial data state. Thus, for any two neighboring states $\mathbf{U}_L$ and $\mathbf{U}_R$, the matrix $\tilde{J} = \tilde{J}(\mathbf{U}_L, \mathbf{U}_R)$ should satisfy the following properties:

i. $\tilde{J}(\mathbf{U}_L, \mathbf{U}_R)$ is diagonalisable with real eigenvalues (Hyperbolicity).

ii. $\tilde{J}(\mathbf{U}_L, \mathbf{U}_R) \Longrightarrow J(\mathbf{U})$ smoothly as $\mathbf{U}_L, \mathbf{U}_R \Longrightarrow \mathbf{U}$ (Consistency).

iii. $\tilde{J}(\mathbf{U}_L, \mathbf{U}_R)\,(\mathbf{U}_R - \mathbf{U}_L) = \mathbf{F}(\mathbf{U}_R)\text{-}\mathbf{F}(\mathbf{U}_L) = \Delta\mathbf{F}$ (Conservation).

The first and second conditions are achieved, if $\tilde{J}$ is chosen to be the Jacobian evaluated at an averaged state $(\tilde{\mathbf{U}})$,i.e. $\tilde{J}(\mathbf{U}_L, \mathbf{U}_R) = \tilde{J}(\tilde{\mathbf{U}})$. However, an average value is generally unable to meet the last condition. Therefore a specific kind of geometric average is used to satisfy this condition and it is also demonstrated by Roe for Euler equations which reconstructs $\tilde{J}$, such that

$$\tilde{J}\,\Delta\mathbf{U} = \Delta\,\mathbf{F} \tag{4.14}$$

where

$$\Delta \mathbf{U} = \sum \tilde{\alpha_k}\, \tilde{\mathbf{e}_k} \tag{4.15}$$

and

$$\Delta \mathbf{F} = \sum \tilde{\lambda_k}\, \tilde{\alpha_k}\, \tilde{\mathbf{e}_k} \tag{4.16}$$

where $\tilde{\alpha_k}$ corresponds to the strength of the k^{th} wave in the Riemann solution that transmits with speed $\tilde{\lambda}^k$. The $\tilde{\mathbf{e}}^k$ are the right eigenvetors of the matrix $\tilde{J}$ with associated eigenvalues $\tilde{\lambda}^k$. By obtaining the approximated quantities, the numerical flux at the cell interface can be defined as:

$$\tilde{\mathbf{F}}(\mathbf{U}_L, \mathbf{U}_R) = \frac{1}{2}(\mathbf{F}_L + \mathbf{F}_R) - \frac{1}{2}\sum |\tilde{\lambda}^k|\, \tilde{\alpha}^k\, \tilde{\mathbf{e}}^k \tag{4.17}$$

Thus, the Roe flux function for 1-d Saint Venant equation in a rectangular channel is given by 4.18 using the definitions

$$\mathbf{U} = \begin{pmatrix} h \\ q \end{pmatrix} \qquad \mathit{and} \qquad \mathbf{F} = \begin{pmatrix} q \\ q^2/h + gh^2/2 \end{pmatrix}$$

$$\tilde{\mathbf{F}}^{Roe}_{i+1/2}(\mathbf{U}^L_{i+1/2}, \mathbf{U}^R_{i+1/2}) = \frac{1}{2}(\mathbf{F}^L_{i+1/2} + \mathbf{F}^R_{1+1/2}) - \frac{1}{2}\sum_{k=1}^{2} |\tilde{\lambda}^k_{i+1/2}|\, \tilde{\alpha}^k_{i+1/2}\, \tilde{\mathbf{e}}^k_{i+1/2} \tag{4.18}$$

From this, the approximate eigenvalues and eigenvectors of Jacobian $\tilde{J}$ for the homogeneous Saint Venant equation at $x_{i+1/2}$ are defined as

$$\tilde{\lambda}^k_{i+1/2} = \tilde{u}_{i+1/2} \pm \tilde{c}_{i+1/2} \quad \mathit{and} \quad \tilde{\mathbf{e}}^k_{i+1/2} = \begin{pmatrix} 1 \\ \tilde{u}_{i+1/2} \pm \tilde{c}_{i+1/2} \end{pmatrix}$$

The variables used in the equation 4.18 are given by equations 4.19 to 4.22 that satisfy all of the aforementioned conditions.

$$\begin{aligned} \tilde{\lambda}^1_{i+1/2} &= \frac{(\tilde{c}_{i+1/2} - \tilde{u}_{i+1/2})\Delta h_{i+1/2} + \Delta q_{i+1/2}}{2\tilde{c}_{i+1/2}} \\ \tilde{\lambda}^2_{i+1/2} &= \frac{(\tilde{c}_{i+1/2} + \tilde{u}_{i+1/2})\Delta h_{i+1/2} - \Delta q_{i+1/2}}{2\tilde{c}_{i+1/2}} \end{aligned} \tag{4.19}$$

$$\tilde{u}_{i+1/2} = \frac{u^R_{i+1/2}\sqrt{h^R_{i+1/2}} + u^L_{i+1/2}\sqrt{h^L_{i+1/2}}}{\sqrt{h^R_{i+1/2}} + \sqrt{h^L_{i+1/2}}} \tag{4.20}$$

$$\tilde{c}_{i+1/2} = \sqrt{\frac{g(h^R_{i+1/2} + h^L_{i+1/2})}{2}} \tag{4.21}$$

$$\begin{aligned}\Delta h_{i+1/2} &= h^R_{i+1/2} - h^L_{i+1/2} \\ \Delta q_{i+1/2} &= q^R_{i+1/2} - q^L_{i+1/2}\end{aligned} \tag{4.22}$$

The Roe expression in the form 4.18 under certain conditions leads to entropy violating the solution. Therefore a number of *entropy fixes* (to ensure capturing of transcritical flows) have been introduced to fix this problem and these are often based to modify the $|\tilde{\lambda}^k_{i+1/2}|$ term in the equation 4.18. The following modification has been used throughout this work

$$|\tilde{\lambda}^k_{i+1/2}|^* = \begin{cases} |\tilde{\lambda}^k_{i+1/2}| & if \quad |\tilde{\lambda}^k_{i+1/2}| \geq \varepsilon^k_{i+1/2} \\ (\tilde{\lambda}^k_{i+1/2})^2/2\varepsilon^k + \varepsilon^k/2 & if \quad |\tilde{\lambda}^k_{i+1/2}| < \varepsilon^k_{i+1/2} \end{cases} \tag{4.23}$$

where

$$\varepsilon^k_{i+1/2} = min[\tilde{c}_{i+1/2},\, max\,[0, 2(\lambda^{k,L}_{i+1/2}, \lambda^{k,R}_{i+1/2})]]$$

4.2.2 Source terms

Discretisation of the source terms creates some well-known problems in the context of shallow water modelling. Firstly, the issue of well-balancing between the convective terms and the bed source term. Secondly, the existence of wet/dry fronts when the bed source term is non-zero. As a consequence, the numerical scheme needs to resolve these issues in order to avoid any non-physical solution.

The simplest and most common way of incorporating the source terms within the Godunov-type schemes for Saint Venant equations is a cell average value that satisfies the following relationship

$$\mathbf{S}_i = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} S(x, \mathbf{U}_i)\, dx \tag{4.24}$$

Thus, the explicit expression 4.10 becomes

$$\mathbf{U}^{t^{n+1}}_i = \mathbf{U}^{t^n}_i + \frac{\Delta t}{\Delta x}[\tilde{\mathbf{F}}_{i+1/2} - \tilde{\mathbf{F}}_{i-1/2}] + \Delta t\, \mathbf{S}^{t^n}_i \tag{4.25}$$

By following the same way in which the numerical flux functions are defined, a source function $\mathbf{S}$ is obtained, which depends on the local variables such that

$$\mathbf{S}^{t^n}_i = \underbrace{\mathbf{S}(x_{i-1}, x_i, x_{i+1}, \mathbf{U}^{t^n}_{i-1}, \mathbf{U}^{t^n}_i, \mathbf{U}^{t^n}_{i+1})}_{Ⓐ} \tag{4.26}$$

where the term Ⓐ can be written in term of left and right form to give

$$\mathbf{S}(x_{i-1}, x_i, x_{i+1}, \mathbf{U}^{t^n}_{i-1}, \mathbf{U}^{t^n}_{i}, \mathbf{U}^{t^n} n_{i+1}) = \mathbf{S}^L(x_{i-1}, x_i, \mathbf{U}^{t^n}_{i-1}, \mathbf{U}^{t^n}_{i}) + \mathbf{S}^R(x_i, x_{i+1}, \mathbf{U}^{t^n}_{i}, \mathbf{U}^{t^n}_{i+1}) \tag{4.27}$$

The definition of $\mathbf{S}^L$ and $\mathbf{S}^R$ depend on the numerical method being used. Hence, the *upwind decomposition* approach that is developed by Bermudez (Bermudez and Vazquez, 1994b) depended on this work in which the source term vector $\mathbf{S}$ decompose into the characteristic basis using the parameters involved within the Roe flux, such that

$$\mathbf{S}_i = \frac{1}{2}\left[(I + |\tilde{J}_{i-1/2}|\,\tilde{J}^{\,-1}_{i-1/2})\,\mathbf{S}_{i-1/2} + (I - |\tilde{J}_{i+1/2}|\,\tilde{J}^{\,-1}_{i+1/2})\,\mathbf{S}_{i+1/2}\right] \tag{4.28}$$

where the variables used in the equation 4.28 are given as follows:

$$\tilde{J}_{i\pm1/2} = \mathbf{R}_{i\pm1/2}\begin{pmatrix}\tilde{\lambda}^1_{i+1/2} & 0\\ 0 & \tilde{\lambda}^2_{i+1/2}\end{pmatrix}\mathbf{R}^{-1}_{i\pm1/2}$$

$$\mathbf{R}_{i\pm1/2} = \left(\tilde{\mathbf{e}}^1_{i+1/2}, \tilde{\mathbf{e}}^2_{i+1/2}\right) = \begin{pmatrix}1 & 1\\ \tilde{\lambda}^1_{i+1/2} & \tilde{\lambda}^2_{i+1/2}\end{pmatrix}$$

$$\mathbf{S}_{i-1/2} = \begin{pmatrix}0\\ -g\left[\frac{h_i+h_{i-1}}{2}\right]\left[\frac{z_i-z_{i-1}}{\Delta x_i}\right]\end{pmatrix}$$

$$\mathbf{S}_{i+1/2} = \begin{pmatrix}0\\ -g\left[\frac{h_i+h_{i+1}}{2}\right]\left[\frac{z_{i+1}-z_i}{\Delta x_i}\right]\end{pmatrix}$$

4.2.3 Wet/Dry bed treatment

As already mentioned, the existence of the bed implies the possibility of having the wetting and drying fronts with the evolution of flow over non-flat beds. Hence, the positivity of water depth is needed within a Godunov-type scheme for solving the expression 4.1. Typically, the Roe solver works well in wet/wet front problems, but in wet/dry front problems, it may produce a negative water depths due to wrong estimation of the source quantity, and this usually leads to crash the computation. In this thesis, the positivity of water depth is achieved by following the strategy of Liang (Liang, 2010), which is illustrated in Appendix B. The strategy is based to reconstruct the free-surface water (η) from the Riemann state before computing the numerical flux, so that the physical representation of the topography is guaranteed and accounts for wet/dry front.

4.2.4 Friction source term

The friction source term (S_f) is discretised following the finite volume pointwise approach which is widely applied in numerical models because of its simplicity. It is locally evaluated as the same way of evaluating the conserved variables and it is integrated explicit with time.

4.2.5 Initial and boundary conditions

The initial and boundary conditions play an important role for the solution of SWE. The flow conditions at the beginning of the computation are referred to as initial conditions, while the boundary conditions govern the flow conditions at the boundaries of the computational domain. These conditions are associated with the characteristics directions of the flow which can be mathematically described by two real eigenvalues, $\lambda^1 = u - c$ and $\lambda^2 = u + c$. They provide the flow information to every point inside the domain of the solution of h and q in the (x, t) plane shown in Figure 4.4. However, at the boundaries of a flow domain, the information about the characteristics may be unknown. Thus, for the solution of 1D SWE, two physical boundary conditions need to be defined independently from the governing equations such as h and q or h and u and also should be independent of each other (see Table 4.1).

Figure removed due to copyrights

(a) Subcritical Flow

Figure removed due to copyrights

(b) Supercritical Flow

Figure 4.4: Characteristics at boundaries for flow regime (Khan and Lai, 2014).

Table 4.1: Boundary Conditions needed for modeling the 1D SWE.

Flow Type	Initial Condition	Inflow Boundary	Outflow Boundary
Critical	2	1	0
Subcritical	2	1	1
Supercritical	2	2	0

4.3 Discontinuous Galerkin method

The original Godunov-type method is conceptually limited to the first-order of accuracy in space as presented in the previous section. But with the appearance of the discontinuous Galerkin method, it has been possible to extend the Godunov-based finite volume method to a higher order of accuracy. It merges the properties of both the finite element method and the finite volume method. It exploits the weak formation of the finite element method to shape a local polynomial solution, which is stored and evolved locally over a cell via a finite element coefficient. In the meantime, it maintains the conservative and stabilising properties that feature the finite volume method, particularly the conservative flux communication across inter-cells.

4.3.1 Discontinuous Galerkin framework

In this section, the DG method for numerically solving the one dimensional shallow water equations is presented. The basic idea is to multiply the system 4.1 by a test function, which is chosen to be a polynomial of degree p within each cell (I_i), and by performing the integration by parts over the domain $\Omega_i = [x_{i-1/2}, x_{i+1/2}]$ of size Δ_i, also removing the flux derivative term from the system often performed in the conventional finite element method. In this way the weak formulation form is obtained:

$$\begin{aligned}
&\int_{I_i} \frac{\partial \tilde{\mathbf{U}}(x,t)}{\partial t}\,\phi_{i,k}(x)\,dx - \int_{I_i} \frac{\mathbf{F}(\tilde{\mathbf{U}}(x,t))}{x}\,\partial_x \phi_{i,k}(x)\,dx \\
&+ \Big[\mathbf{F}(\tilde{\mathbf{U}}(x_{i+1/2},t))\,\phi_{i,k}(x_{i+1/2}) - \mathbf{F}(\tilde{\mathbf{U}}(x_{i-1/2},t))\,\phi_{i,k}(x_{i-1/2})\Big] \\
&= \int_{I_i} \frac{\partial \mathbf{S}(\tilde{\mathbf{U}}(x,t))}{\partial x}\,\phi_{i,k}(x)\,dx
\end{aligned} \tag{4.29}$$

The DG approximate solution $\tilde{\mathbf{U}}$ is sought in the space (V_p) of polynomials of degree $\leq p$ on I_i and can be locally expanded using finite element coefficients, i.e.

$$\tilde{\mathbf{U}}(x,t)|_{I_i} = \sum_{k=0}^{p} \mathbf{U}_{i,k}(t)\, \phi_{i,k}(x) \tag{4.30}$$

where $\phi_{i,k}$ are the polynomial basis function locally supported on cell i, with polynomial degree $0 \leq\; k \;\leq p$, and $\{\mathbf{U}_{i,k}\}_{0\leq\,k\,\leq p}$ are unknown coefficients associated with the local polynomial degree in cell i and can be described as degree of freedom (Cockburn and Shu, 1998). These coefficients can be initialised as

$$\mathbf{U}_{i,k}(x,0) = \frac{1}{a_{i,j}} \int_{I_i} \mathbf{U}^0 \phi_{i,k}(x)\, dx \tag{4.31}$$

where $a_{i,k}$ is the dot product coefficient and can be obtained as follow

$$a_{i,j} = \int_{I_i} \phi_{i,k}\phi_{i,k} dx \tag{4.32}$$

The approximation in equation 4.30 has an order of accuracy $k+1$. Thus, if the basis function $(k=0)$ is used, the first order accuracy is achieved, and in consequence the scheme can be referred to as Godunov-type scheme, while when the linear basis functions $(k=1)$ are used, the second order accuracy is achieved. Here the scheme is referred to as DG2.

The test function are taken to be the same as the basis functions, i.e. $v(x) = \{\phi_{i,k}\}_{k=0}^{p}$ and we particularly select the Legendre polynomial due to their orthogonality property and also due to the multiwavelets which use the same basis functions, as given in Chapter two.

$$\int_{-1}^{1} \phi_m(\xi)\phi_k(\xi)\, dx \;= \frac{2}{2k+1}\,\delta_{mk} \tag{4.33}$$

where

$$\xi = \frac{2(x-x_i)}{\Delta x_i} \qquad\qquad \delta_{mk} = \begin{cases} 1, & m=k, \\ 0, & \text{elsewhere}, \end{cases}$$

By invoking the orthogonality property of the Legendre's polynomial, the weak formulation 4.29 simplifies to

$$\begin{aligned}\frac{d}{dt}\mathbf{U}_{i,k}(t) = &-\frac{2}{\Delta x_i}\frac{(2k+1)}{2}\Bigg\{\underbrace{\left[\tilde{\mathbf{F}}_{i+1/2}-(-1)^k\,\tilde{\mathbf{F}}_{i-1/2}\right]}_{\mathcal{M}_{i,k}}\\ &-\underbrace{\frac{\Delta x_i}{2}\int_{-1}^{+1}\mathbf{F}\big(\tilde{\mathbf{U}}(x_i+\frac{\Delta x}{2}\xi,t)\big)\frac{d\phi_{ik}}{d\xi}(\xi)d\xi}_{\mathcal{K}_{i,k}}\\ &-\underbrace{\frac{\Delta x_i}{2}\int_{-1}^{+1}\mathbf{S}\big(\tilde{\mathbf{U}}(x_i+\frac{\Delta x}{2}\xi,t)\big)\phi_{ik}(\xi)d\xi}_{\mathcal{S}_{i,k}}\Bigg\}\end{aligned} \tag{4.34}$$

In equation 4.34, the local integral term $\mathcal{K}_{i,k}$ is computed by the Gauss-Legendre rule with $(p+1)$ quadrature points (see Appendix A). Numerical flux $\tilde{\mathbf{F}}_{i\pm1/2}$ is obtained by approximately Roe Riemann solver. The approximate source term requires the existence of bed z projected onto the orthogonal basis (i.e. onto V_k). The construction of this function is to ensure the well-balancing and wet/dry, and is discussed in the next section.

Integration in time is performed by the strong stability preserving Runge-Kutta procedure with $(k+1)$ stage to solve the *Ordinary Differential Equations* in equation 4.34 with a CFL number less than $1/(2k+1)$.

$$\begin{aligned}\mathbf{U}^{(1)} &= \mathbf{U}^{t^n} + \Delta t\, L(\mathbf{U}^{t^n})\\ \mathbf{U}^{(2)} &= \frac{3}{4}\mathbf{U}^{t^n} + \frac{1}{4}\big(\mathbf{U}^{(1)} + \Delta t\, L(\mathbf{U}^{(1)})\big)\\ \mathbf{U}^{t^{n+1}} &= \frac{1}{3}\mathbf{U}^{t^n} + \frac{2}{3}\big(\mathbf{U}^{(2)} + \Delta t\, L(\mathbf{U}^{(2)})\big)\end{aligned} \tag{4.35}$$

4.3.2 Well-balancing treatment and wet/dry front

In the DG context, several strategies have been proposed for discretising the bed source term $\mathbf{S}$ in a well-balanced way, most of which have risen from the finite volume context. The issue of well-balancing arises from approximating the numerical flux $\tilde{\mathbf{F}}$ at the cell interfaces. This is because the approximate Riemann solver assumes that the source term does not exist, i.e. $\mathbf{S} = 0$ but in fact that is not true, i.e. $\mathbf{S} \neq 0$.

In this work, well-balancing is achieved by following the strategy of Kesserwani and Liang (Kesserwani and Liang, 2010b). This strategy can be generalised in two steps. The first step is to project the bed function into the same space of the local polynomial function (i.e. the space of conserved variables). The second step is to ensure that the projected bed function is continuous across the cell interface. Here the setting for DG2 can be given as:

$$\tilde{z}(x)\big|_{I_i} = z_{0,i} + z_{1,i}\Big(\frac{x - x_i}{\Delta x/2}\Big) \qquad (\forall\, x \in\, I_i) \tag{4.36}$$

where

$$z_{0,i} = \int_{x_{i-1/2}}^{x_{i+1/2}} z(x)\, dx \approx \frac{z(x_{i+1/2}) + z(x_{i-1/2})}{2}$$

and

$$z_{1,i} = \int_{x_{i-1/2}}^{x_{i+1/2}} \Big(\frac{x - x_i}{\Delta x/2}\Big) z(x)\, dx \approx \frac{z(x_{i+1/2}) - z(x_{i-1/2})}{2}$$

With this representation of the bed, the local gradient of bed becomes

$$\begin{aligned} \partial_x \tilde{z}(x)\big|_{I_i} &= \partial_x \Bigg[z_{0,i} + z_{1,i}\Big(\frac{x - x_i}{\Delta x/2}\Big)\Bigg] \\ &= \frac{\big[z(x_{i+1/2}) - z(x_{i-1/2})\big]/2}{\Delta x/2} \\ &= \frac{\big[z(x_{i+1/2}) - z(x_{i-1/2})\big]}{\Delta x} \end{aligned} \tag{4.37}$$

and also the continuity step is obtained at interface, such as at $x_{i+1/2}$, which can be given as

$$\tilde{z}(x^L_{i+1/2})\big|_{I_i} = z_{0,i} + z_{1,i} = z_{0,i} - z_{1,i} = \tilde{z}(x^R_{i+1/2})\big|_{I_{i+1}} \tag{4.38}$$

To cope with the issue of wetting and drying fronts, the strategy presented by Kesserwani (Kesserwani and Liang, 2012b) (see Appendix C) is used in this thesis. It is based on the continuous bed well balancing strategy described in equation 4.38, and also it enforces that the numerical solution must remain positive in the entire cell. This implies with DG2 context it would be enough to check the cell interfaces. If the value of water depth at the interface is smaller than a chosen dry-threshold value ε_{dry}, the slope coefficient of the projected depth $h_{1,i}$ is modified to ensure positivity. Accordingly, the momentum at the interface is also modified.

The descretisation of friction source term in DG is obtained via generalising the approach described in Subsection 4.2.4 in which the friction value is evaluated at each Gaussian quadrature point and then integrating it explicitly in time.

4.3.3 Slope limiter

The high-order schemes provided by RKDG need slope limiters to be activated at the discontinuities to maintain stability and avoid spurious oscillations. But perhaps the accuracy of the reference scheme is not preserved at the such portion of the computational domain where the slope limiter is activated. Limiting technique affects on the data as the scale difference of the point value at the cell boundary and the mean value of the cell (see Figure 4.5). In this work, the strategy suggested by Kesserwani (Kesserwani and Liang, 2011) is used which is based on the idea of Krivodonova (Krivodonova et al., 2004) applied to the numerical solution of the homogeneous Euler equations. It uses the generalised TVD-*minmod* and applied to the coefficients $\mathbf{U}_{i,k}(t^n)$ at time (t^n). Since it is not robust tool to use in detecting the *troubled-cells* or *troubled slope* a user depending parameter is used for detection process. The variation of the local k^{th}-order slope coefficients $(\widehat{\mathbf{U}}_{i,k})$ is reconstructed from $k-1^{th}$-order coefficients of local slope, upstream and downstream gradients, i.e.

$$\widehat{\mathbf{U}}_{i,k} = \frac{1}{(2k-1)}\, minmod\Big[(2k-1)\mathbf{U}_{i,k}, \mathbf{U}_{i+1,k-1}-\mathbf{U}_{i,k-1}, \mathbf{U}_{i,k-1}-\mathbf{U}_{i-1,k-1}\Big] \quad (4.39)$$

The discontinuity detector for DG2 can be performed by first computing the following expression

$$\mathbf{DS}^L_{i+1/2} = \frac{\big|\mathbf{U}^R_{i+1/2} - \mathbf{U}^L_{i+1/2}\big|}{|\frac{\Delta x}{2}|max\{\big|\mathbf{U}_{0,i} - \mathbf{U}_{1,i}\big|, \big|\mathbf{U}_{0,i} + \mathbf{U}_{1,i}\big|\}} \quad (4.40)$$

in which

$$\begin{aligned} \mathbf{U}^L_{i+1/2} =& \tilde{\mathbf{U}}(x^L_{i+1/2}, t) = \mathbf{U}_{0,i} + \mathbf{U}_{1,i} \\ \mathbf{U}^R_{i+1/2} =& \tilde{\mathbf{U}}(x^R_{i+1/2}, t) = \mathbf{U}_{0,i+1} + \mathbf{U}_{1,i+1} \end{aligned} \quad (4.41)$$

And second, comparing the $\mathbf{DS}^L_{i+1/2}$ with the unity value, when $\mathbf{DS}^L_{i+1/2} \to 0$, the DG solution is smooth and no need to activate the slope limiter whereas the DG solution must be limited when $\mathbf{DS}^L_{i+1/2} \to \infty$. In the same way, the $\mathbf{DS}^R_{i-1/2}$ can be evaluated by using these coefficients $\mathbf{U}_{0,i}$,$\mathbf{U}_{1,i}$,$\mathbf{U}_{0,i-1}$ and $\mathbf{U}_{1,i-1}$ in equation 4.40. Ultimately, the slope coefficient $\mathbf{U}_{i,1}$ at cell I_i is limited once a discontinuity is detected at $x_{i+1/2}$ or $x_{i-1/2}$.

Figure removed due to copyrights

Figure 4.5: Illustration of limiter on point value at cell boundary, (Hovhannisyan et al., 2014).

Part II: Adaptive numerical schemes

4.4 Introduction

In the context of shallow water flow, the real large-scale flows usually have complex features such as shocks, contact discontinuities and a wide range of spatial scales. Typically, the computational domain is uniformly discretised using a large number of cells, given that the position of flow features is usually unknown and capturing certain small scales within a coarse mesh simulation may be difficult without a trade-off of computational cost. Therefore, the mesh adaptation technique comes into play to improve modelling efficiency and capture the various physical scales involved in shallow water flows.

From our discussion in chapter 3 and previous sections, it should now be clear that the theory of discrete wavelets offer a good mechanism for transformation of single-resolution local basis into multiresolution and can be incorporated into the DG and FV Godunov-type methods, since their conceptual frameworks are based on using the same Legendre polynomial bases. The adaptivity strategy here depends on exploiting the information embedded within the local solution of both aforementioned numerical methods and performing a multiresolution analysis, which decomposes the local solution into a set of coarse information and details information at a different resolution level. The details may be negligible in smooth regions of the solution, while they may be very significant in non-smooth regions of the solution. Hence, the adaptive solution can be established to trigger adaptivity according to the magnitude of such details (Alpert et al., 2002). In case of using the Haar wavelet bases to reformulate a FV Godunov-type method, the new adaptive Haar wavelet FV scheme (HWFV) is obtained, while the adaptive multiwavelets DG scheme (MWDG) is obtained when incorporating multiwavelets into the DG method.

4.5 The DG discretisation with multiresolution-based mesh adaptivity

As previously stated, we seek to speed up the convergence of a reference FV Godunov-type and the DG schemes on a uniform mesh by utilising the adaptive mesh. In the following we start to reformulate the original DG scheme. The reformulation of FV Godunov-type is straightforward, and only needs to reduce the local accuracy of the adaptive DG scheme to the first-order accurate solution.

The equation 4.34 represents a single resolution approximation of DG in level $n = 0$. To introduce the muliresolution transformation of the DG approximation, we have to assume that the cell exists within the hierarchy of meshes, therefore, $I^n_{i,j} = \left[x_i + \Delta x/2\left(1 + 2^{-n+1}j\right), x_i + \Delta x/2\left(1 + 2^{-n+1}(j+1)\right)\right] \subset I_i$ of a sub-cell centre spatial resolution $x^n_{ij} = x_{i-1/2} + 2^{-n}\Delta x(j + 1/2)$ (see Figure 4.6), and the local approximation $\tilde{\mathbf{U}}$, is sought into a space of polynomial of degree at most k.i.e. $\tilde{\mathbf{U}} \subset V^n_k$, which is locally expanded via using all *orthonormal* Legendre polynomial basis functions that are scaled and translated on cell I^n_i. Consequently, the local DG approximation given in equation 4.30 becomes

$$\begin{aligned}\tilde{\mathbf{U}}(x,t)\big|_{I^n_i} &= \sum_{n=0}^{N}\sum_{j=0}^{2^n-1}\sum_{k=0}^{p} \mathbf{U}^n_{ijk}(t)\,\phi^n_{ijk}\Big(\frac{2}{\Delta x}(x - x_i)\Big) \qquad \big(x \in I_i\big)\\ &= 2^{n/2}\sum_{n=0}^{N}\sum_{j=0}^{2^n-1}\sum_{k=0}^{p} \mathbf{U}^n_{ijk}(t)\,\phi_{ik}\left(2^n\Big(\frac{2}{\Delta x}(x - x_i) + 1\Big) - 2j - 1\right)\end{aligned} \tag{4.42}$$

where $\mathbf{U}^n_{ijk}$ is an approximate coefficients and can be initialized as

$$\begin{aligned}\mathbf{U}^n_{ijk}(0) = \langle \mathbf{U}, \phi^n_{ijk}\rangle = &\int_{-1}^{+1} \mathbf{U}\big(x_i + \frac{\Delta x}{2}\,\xi\big)\,\phi_{ijk}(\xi)\,d\xi\\ =&2^{n/2}\int_{-1+2^{-n+1}(j)}^{-1+2^{-n+1}(j+1)} \mathbf{U}(x_i + \frac{\Delta x}{2}\xi)\,\phi_{ik}\Big(2^n(\xi+1) - 2j - 1\Big)\,d\xi\end{aligned} \tag{4.43}$$

It can simplify the limits of integration in equation 4.43 to support $[-1, +1]$ instead of $[-1+2^{-n+1}j, -1+2^{-n+1}(j+1)]$ by substituting the following expressions into 4.43, such that

$$\begin{aligned}\sigma &= 2^n(\xi + 1) - 2j - 1 \qquad \sigma \in [-1, +1]\\ \xi &= = 2^{-n}(\sigma + 2j + 1) - 1\\ d\xi &= 2^{-n}d\sigma\end{aligned} \tag{4.44}$$

Thus, equation 4.43 becomes

$$\mathbf{U}^n_{ijk} = 2^{-n/2} \int_{-1}^{+1} \mathbf{U}\Big(x_i + \frac{\Delta x}{2}\big(2^{-n}(\sigma + 2j + 1) - 1\big)\Big)\, \phi_{ik}(\sigma)\, d\,\sigma \tag{4.45}$$

Now, we can use the Gauss Legendre points $\chi_0, \cdots, \chi_g$ and their associated weights $\omega_0, \cdots, \omega_g$ to approximate $\mathbf{U}^n_{ijk}$

$$\mathbf{U}^n_{ijk} \approx 2^{-n/2} \sum_{r=0}^{g} \omega_r \mathbf{U}\Big(x_i + \frac{\Delta x}{2}\big(2^{-n}(\chi_r + 2j + 1) - 1\big)\Big)\, \phi_{ik}(\chi_r) \tag{4.46}$$

By considering the relation 3.7, an equivalent expression of projection of $\tilde{\mathbf{U}}$ into space V^n_k exists by means of using the multiwavelets that described in section 3.3, such that

$$\tilde{\mathbf{U}}(x,t)\big|_{I^n_i} = \sum_{k=0}^{p} \mathbf{U}_{ik}\, \phi_{ik}\Big(\frac{2}{\Delta x}(x - x_i)\Big) + \sum_{n=1}^{N-1} \sum_{j=0}^{2^n-1} \sum_{k=0}^{p} \mathbf{D}^n_{ijk}\, \psi^n_{ijk}\Big(\frac{2}{\Delta x}(x - x_i)\Big) \tag{4.47}$$

where $\mathbf{D}^n_{ijk}$ represent the detail coefficients that hold all the multiscale information across the mesh hierarchy up to level $n-1$. This means if we add them to the lowest-resolution information such as at level n, the local approximate solution will promote to level $n+1$, see Figure 4.7. However, for practical reasons and for computational efficiency, the expression 4.47 is not applied directly, since what is of interest is to demote and promote the coefficients, and not the solution itself. Thus, the filter matrices relations described in subsection 3.3.4 can be used to achieve this. To better clarify the promotion and demotion transformation of the coefficients across resolution levels, two successive resolution levels such as $n = 0$ and $n = 1$ are chosen. The first step is to decompose the coefficients at the finest level and here is level one (referred to as demoting):

$$\mathbf{U}^0_{i\,0\,k} = \sqrt{1/2}\,\big(\mathbf{H}^0\, \mathbf{U}^1_{i0k} + \mathbf{H}^1\, \mathbf{U}^1_{i1k}\big) \tag{4.48a}$$

$$\mathbf{D}^0_{i\,0\,k} = \sqrt{1/2}\,\big(\mathbf{G}^0\, \mathbf{U}^1_{i0k} + \mathbf{G}^1\, \mathbf{U}^1_{i1k}\big) \tag{4.48b}$$

Combined use of both filters $\big($i.e.$\mathbf{H}^{0,1}$ and $\mathbf{G}^{0,1}\big)$ allows us to compute $\{\mathbf{U}^1_{i\,j\,k}\}_{j=0,1}$ from $\mathbf{U}^0_{i0k}$ and heir (stored) complementary detail coefficients $\mathbf{D}^0_{i0k}$ (referred to as promoting):

$$\mathbf{U}^1_{i0k} = \sqrt{2}\,\big(\mathbf{H}^{0^T}\, \mathbf{U}^0_{i0k} + \mathbf{G}^{0^T}\, \mathbf{D}^0_{i0k}\big) \tag{4.49a}$$

$$\mathbf{U}^1_{i1k} = \sqrt{2}\,\big(\mathbf{H}^{1^T}\, \mathbf{U}^0_{i0k} + \mathbf{G}^{1^T}\, \mathbf{D}^0_{i0k}\big) \tag{4.49b}$$

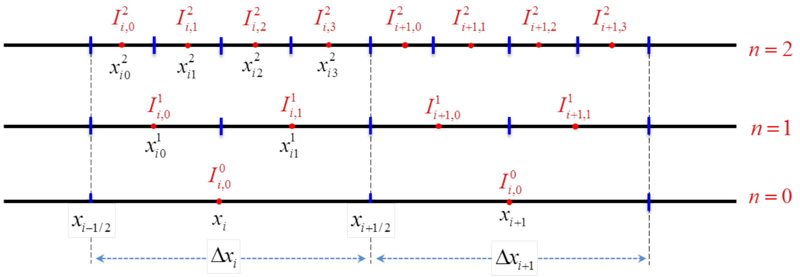


Figure 4.6: Nested mesh hierarchy up to level $n = 2$.

.

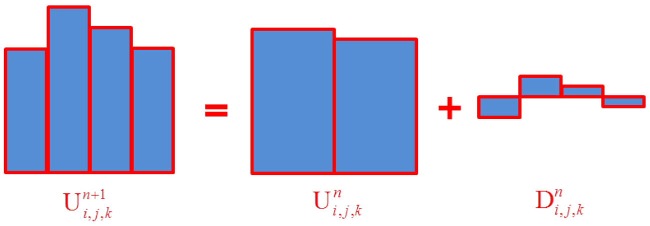


Figure 4.7: Two scale transformation (Hovhannisyan et al., 2014)

.

By recursive application of the two-scale transform equations, the expansion of scaling coefficients describing a local solution can be demoted or promoted across different resolutions (see Figure 4.8). These transformations have very appealing and convenient property that allow us to conserve the accuracy of the solution when cycling over the transformation. This is proven in the example given in chapter two.

4.5.1 The DG multisclae formulation

The equation 4.29 can be reworked by considering these equations 4.42, 4.43 and 4.45. Furthermore, by using the *orthonormal* basis functions along with reworking, the following scale-dependent semi-discrete local DG operators are obtained:

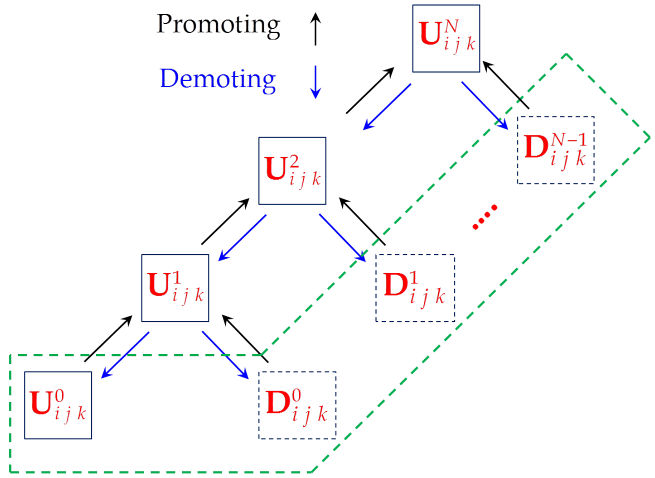

Promoting
Demoting
$\mathbf{U}_{i\,j\,k}^{N}$
$\mathbf{D}_{i\,j\,k}^{N-1}$
$\mathbf{U}_{i\,j\,k}^{2}$
$\mathbf{U}_{i\,j\,k}^{1}$
$\mathbf{D}_{i\,j\,k}^{1}$
$\mathbf{U}_{i\,j\,k}^{0}$
$\mathbf{D}_{i\,j\,k}^{0}$

Figure 4.8: The promoting and demoting of the scaling coefficients numerical solution across different resolutions.

$$\begin{aligned}\frac{d}{dt}\mathbf{U}_{ijk}^{n_i}(t) = &-\frac{2^{n_i+1}}{\Delta x_{ij}^{n}}\underbrace{\Bigg\{2^{n_i/2}\sqrt{\frac{2k+1}{2}}\left[\tilde{\mathbf{F}}_{j+1/2}^{n_i}-(-1)^k\,\tilde{\mathbf{F}}_{j-1/2}^{n_i}\right]}_{\mathcal{M}_{i,k}}\\ &-2^{-n_i/2}\sqrt{\frac{2k+1}{2}}\frac{\Delta x_{ij}^{n_i}}{2}\underbrace{\int_{-1}^{+1}\mathbf{F}\Big(\tilde{\mathbf{U}}\Big(x_i+\frac{\Delta x}{2}\big(2^{-n_i}(\xi+2j+1)-1,t\big)\Big)\Big)\frac{d\phi_{ik}}{d\xi}(\xi)d\xi}_{\mathcal{K}_{i\,j\,k}}\\ &-2^{-n_i/2}\sqrt{\frac{2k+1}{2}}\frac{\Delta x_{ij}^{n_i}}{2}\underbrace{\int_{-1}^{+1}\mathbf{S}\Big(\tilde{\mathbf{U}}\Big(x_i+\frac{\Delta x}{2}\big(2^{-n_i}(\xi+2j+1)-1,t\big)\Big)\Big)\phi_{ik}(\xi)d\xi}_{\mathcal{S}_{i\,j\,k}}\Bigg\}\end{aligned} \tag{4.50}$$

In equation 4.50, the local integral term $\mathcal{K}_{i\,j\,k}$ is computed by the Gauss-Legendre rule with $(p+1)$ quadrature points. The approximate source term requires the existence of bed z projected onto the *orthonormal* basis functions (i.e. onto V_k^n) and its integral term $\mathcal{S}_{i\,j\,k}$ is computed in the same way of term $\mathcal{K}_{i\,j\,k}$. Numerical flux term $\mathcal{M}_{i\,j\,k}$ through the boundaries is obtained by approximately Roe Riemann solver. To ensure the well-balancing and wet/dry, the same strategy described for DG has been applied to the multiscale version because the locality of solution is preserved. Integration in time is performed by using Runge-Kutta procedure with $(k+1)$ stage to solve the *ordinary differential equations* in 4.50 with a CFL number less than $1/(2k+1)$.

4.5.2 The FV Godunov-type multiscale formulation

The adaptive version of FV Godunov-type can be obtained via omitting the flux term (i.e. $\mathcal{K}_{i\,j\,k}$) from the expression 4.50 ($\because \frac{d\phi_i}{d\xi}(\xi)d\xi = 0$) and also put $k = 0$. This yields:

$$\begin{aligned}\frac{d}{dt}\mathbf{U}_{ij}^{n_i}(t) = -\frac{2^{n_i+1}}{\Delta x_{ij}^{n}} \Bigg\{ & \underbrace{2^{n_i/2}\sqrt{\frac{1}{2}}\Big[\tilde{\mathbf{F}}_{j+1/2}^{n_i} - \tilde{\mathbf{F}}_{j-1/2}^{n_i}\Big]}_{\mathcal{M}_{ij}} \\ & - 2^{-n_i/2}\sqrt{\frac{1}{2}}\frac{\Delta x_{ij}^{n_i}}{2} \underbrace{\int_{-1}^{+1} \mathbf{S}\bigg(\tilde{\mathbf{U}}\Big(x_i + \frac{\Delta x}{2}\big(2^{-n_i}(\xi + 2j + 1) - 1, t\big)\Big)\bigg)\phi_i(\xi)d\xi}_{\mathcal{S}_{i\,j}} \Bigg\}\end{aligned} \tag{4.51}$$

In equation 4.51, the local integration terms $\mathcal{M}_{ij}$ and $\mathcal{S}_{i\,j}$ follows the same strategy that is described for Godunove-type methods in Part I Section 4.2.

4.6 Adaptivity process

One of the most important issues in the adaptive numerical schemes is to design robust criteria that identify which regions have solutions that need further mesh refinement because of their spatial variability. The adaptive strategy here is based on measuring the magnitude of detail coefficients according to prescribed threshold value and this is the only value that must be selected by user to control the adaptivity of mesh.

In general, the adaptive version of Godunov-type FV and DG used here mainly consist of three key steps. In the first, prediction is needed for refinement. In this context, the refinement means to promote the local solution to the higher resolution. The second step is performing the multi-scale FV/DG update on the predicted mesh. Finally, hard thresholding is performed.

4.6.1 Prediction step for mesh refinement

Since the flow field is evolving in time, the prediction step must be performed after each time step to refine the mesh and guarantee no significant (future) features of the numerical solution are omitted in the next time step. The prediction strategy is only based on the information available at the current time level (Müller, 2003;

Müller and Stiriba, 2007). Generally, the detail coefficients of predicted cells are not available. Thus, the local solution over the predicted cells is promoted by simply setting zero detail coefficients (Harten, 1995). To do this, Algorithm 3 is used to identify those neighbourhood cells that needs to be further refined:

Algorithm 3 Predictions for mesh refinement

```
for every Time steps do
    1: Compute the local solution of the conserved variables at $n = 0$.
    2: Use Eq. 4.52 to compute the normalized gradient $\tau$ between the
    local cell and its neighbor cell.
    3: Introduce two indicators to compare values and decide the resolution
    levels of neighboring cells that may have significant information.
    4: Choose $\tau = 0.1$ and $\tau = 0.05$ as indicators for all numerical test cases.
    5: The decision for the local refinement mesh will take the following
    if $\tau \geq 0.1$ then
        The adaptive mesh is in as Figure 4.9
    else if $0.1 > \tau \geq 0.05$ then
        The adaptive mesh is in as Figure 4.10
    else
        The adaptive mesh is in as Figure 4.11
    end if
end for
```

$$\tau = \frac{\left|\tilde{\mathbf{U}}|_{I_i^0} - \tilde{\mathbf{U}}|_{I_{i+1}^0}\right|}{max\left(1, \left|\tilde{\mathbf{U}}|_{I_i^0}\right|\right)} \tag{4.52}$$

4.6.2 Multi-scale update

After the prediction step, The RKDG evolution and FV are performed as a single time step using equation 4.50 and 4.51 respectively accounting different resolution levels that are taking over each cell (i.e. n_i). In this work, the both adaptive formulation are applied in conjunction with a proper discretisation of the topography source term (more details will be given in the following subsection), slope limiter and a Roe Riemann solver. Algorithm 4 summarises the entire adaptive solution process for the MWDG and HWFV schemes.

4.6.2.1 Adaptivity of topography

The topography is projected into the highest resolution in the same way as the conserved variables. Thus the compressed data set of bed information over each cell is obtained. Finally, the adaptive bed mesh is obtained by applying the thresholding step. However, the topography adaptation is performed once (i.e., initially when $t = 0$) and remains constant throughout the simulation time. To ensure the preservation of an accurate water surface elevation and mass conservation across levels, the bed mesh should be considered as a reference; this means that the adaptive schemes do not allow demoting the local numerical solution to coarsen the mesh to a resolution level lower than the topography refinement, even if the local numerical solution allows for a lower refinement level.

4.6.3 Hard thresholding

Hard thresholding is applied on the detail coefficients after each update to decompress the mesh. The values of detail coefficients ($\mathbf{D}^n_{i\,j\,k}$) become small when the numerical solution is smooth. Therefore, they can be cancelled without substantially affecting the accuracy of the numerical solution. To do so, all detail coefficients that their absolute values are below a normalised level-dependent threshold value are discarded, i.e. if they satisfy

$$\check{\mathbf{D}}^n_{i\,j\,k} = \begin{cases} \mathbf{D}^n_{i\,j\,k} \quad if \quad \underset{r\,\in\,\mathbf{U}}{max}\left(\dfrac{\left|\left(\mathbf{D}^n_{i\,j\,k\,r}\right)\right|}{max\left(\underset{i}{max}\left(\left|\mathbf{U}^0_{i\,0\,0\,r}\right|,1\right)\right)}\right) > \varepsilon^n \\ \\ 0, \qquad otherwise \end{cases} \tag{4.53}$$

where $\mathbf{U}^0_{i\,0\,0\,r}$ is the cell average of r^{th} conserved variable in cell $I^0_{i\,k}$ (i.e. index r spans the components of the conserved variables), $\check{\mathbf{D}}^n_{i\,j\,k}$ is the significant details coefficients in which their components are associated with the components of the conserved variables in $\mathbf{U}^n_{ijk}$. The ε^n is a level-dependent threshold value defined as

$$\varepsilon^n = \varepsilon\, 2^{n-N} \tag{4.54}$$

where ε is the prescribed threshold value for adaptivity and this is the only value that must be defined by the user in order to control the adaptivity process. In real computations, it is impossible to know the optimal threshold value. But a range of options is feasible (Hovhannisyan et al., 2014; Gerhard et al., 2015) as choosing it in such way that can retain the same quality of solution as the reference fine uniform mesh provided. This strategy is investigated in the numerical results chapter. In this work, the threshold value (at the coarsest level) is set to $\varepsilon = 0.01$ and the reference uniform mesh can be expressed in a single scale expansion at the highest resolution level.

Algorithm 4 Full adaptive solution for the HWFV and MWDG schemes

```
for Initial condition do
    1: Initial projection of the solution and the bed source term at highest
       resolution level following Eq. 4.45 considering the associated
       basis functions.
    2: Use Eq. 4.48 to demote the initial projection from step (1)
       to coarse level $n = 0$, considering the associated filter matrices
    3: Initial thresholding of topography and the solution to obtain adaptive
       mesh.
    4: Use Eq. 4.49 to obtain an adaptive mesh.
end for
for all Time steps in simulation do.
    5: Use Eq. 4.48 to see the details coefficients.
    6: Perform the Prediction (Algorithm 3).
    7: Find and flag significant details.
    8: Use Eq. 4.49 to get the adaptive mesh.
    procedure UPDATE(MWDG)
        for Runge-Kutta stages do
            9: Impose boundary conditions
            10: Perform limiting and ensure depth-positivity at edges of the
                cells (Appendix C).
            11: Perform Eq. 4.50 to update the operators.
        end for
    end procedure
end for
Continue
```

```
for all Time steps in simulation do.
    procedure UPDATE(HWFV)
        for Euler stage do
            9: Impose boundary conditions
            10: ensure depth-positivity at edges of the
                cells (Appendix B).
            11: Perform Eq. 4.51 to update the operators.
        end for
    end procedure
    procedure HARD THRESHOLDING
        12: Perform Eq. 4.53 on $\mathbf{D}_{ijk}^{N-1}$ obtained from Eq. 4.48).
        13: Find the non-significant detail coefficients and flag the significant
            detail coefficients.
        14: Zeroing All Non-significant detail coefficients
    end procedure
end for
```

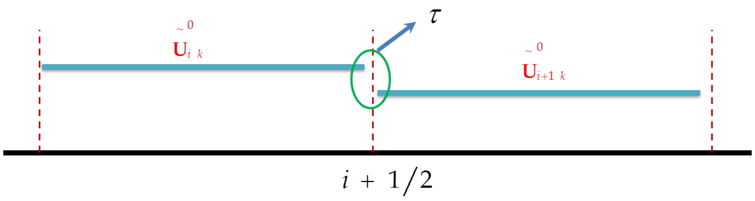

τ
$\tilde{\mathbf{U}}^0_{i\;k}$
$\tilde{\mathbf{U}}^0_{i+1\;k}$
$i + 1/2$

Figure 4.9: Mesh prediction $\tau < 0.05$.

.

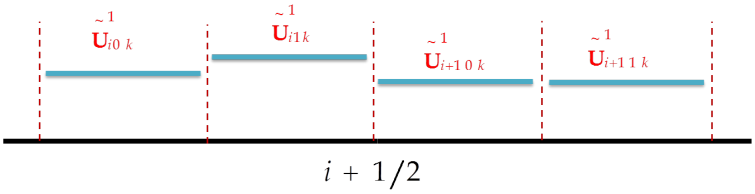
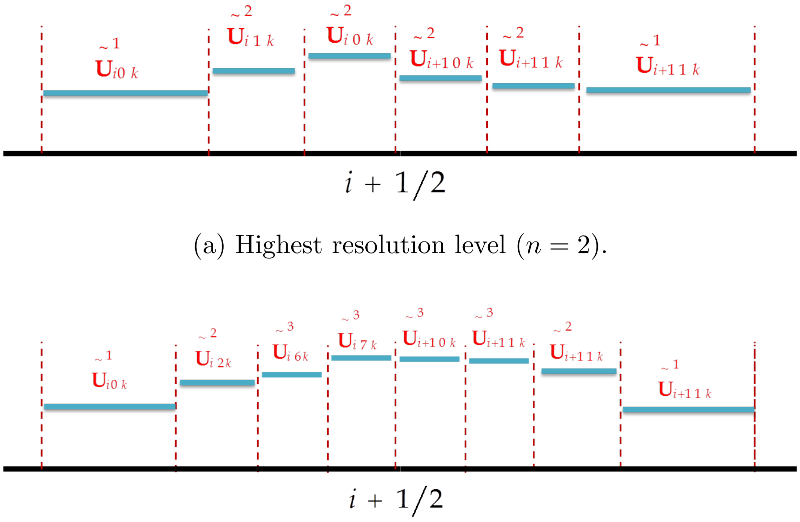

$\tilde{\mathbf{U}}^1_{i0\;k}$
$\tilde{\mathbf{U}}^1_{i1k}$
$\tilde{\mathbf{U}}^1_{i+1\,0\;k}$
$\tilde{\mathbf{U}}^1_{i+1\,1\;k}$
$i + 1/2$

Figure 4.10: Mesh prediction $0.1 \geq \tau \geq 0.05$.

.


$\tilde{\mathbf{U}}^1_{i0\;k}$
$\tilde{\mathbf{U}}^2_{i\,1\;k}$
$\tilde{\mathbf{U}}^2_{i\,0\;k}$
$\tilde{\mathbf{U}}^2_{i+1\,0\;k}$
$\tilde{\mathbf{U}}^2_{i+1\,1\;k}$
$\tilde{\mathbf{U}}^1_{i+1\,1\;k}$
$i + 1/2$
$\tilde{\mathbf{U}}^1_{i0\;k}$
$\tilde{\mathbf{U}}^2_{i\;2k}$
$\tilde{\mathbf{U}}^3_{i\;6k}$
$\tilde{\mathbf{U}}^3_{i\;7\;k}$
$\tilde{\mathbf{U}}^3_{i+1\,0\;k}$
$\tilde{\mathbf{U}}^3_{i+1\,1\;k}$
$\tilde{\mathbf{U}}^2_{i+1\,1\;k}$
$\tilde{\mathbf{U}}^1_{i+1\,1\;k}$
$i + 1/2$

(a) Highest resolution level ($n = 2$).

(b) Highest resolution level ($n = 3$).

Figure 4.11: Mesh prediction when $\tau \geq 0.1$.

.

Chapter 5

Numerical results

5.1 Introduction

In this chapter different test cases are used to assess the wavelet-based adaptivity with two different schemes (i.e., HWFV and MWDG2). The aim is to verify the capability of these two schemes in performing simulation of flow problems with fewer cells when compared to the standard FV and DG2 (i.e., non-adaptive schemes). In addition, the numerical test cases reported herein, seek to study the impact of the adaptivity on the resolution accuracy, the conservation of mass in the system, show the properties of well-balancing, and the positivity of the water depth are transferable into the adaptivity. The quantitative analyses are also computed in some test cases via calculating the **R**oot **M**ean **S**quare **E**rror (*RMSE*), **R**elative **M**ean **E**rror (*RM*E), maximum error, mean and the **S**tandard **D**eviation (SD) for both the water depth and flow rate to confirm the numerical results. It is worth mentioning that the adaptivity process is not seeking to improve the modelling over the uniform FV and DG schemes, but it seeks to conserve the properties of the reference schemes at the same quality as they are delivered on the uniform counterpart schemes.

The benchmark hydraulic tests are divided into two parts, part I contains test cases for investigating key issues, which are identified for the relevance to wavelet-based adaptivity. In part II, more validations and comparisons of both adaptive schemes are performed using popular hydraulic benchmark test cases.

In general, the results of the adaptive and non-adaptive schemes are computed using the strategies that were detailed in Chapter 4 for well-balancing, wetting and drying and friction discretization. In Table 5.1, several standard parameters are listed which are used in all numerical computations. If any of them are changed, it will be stated in the test case.

Table 5.1: Input parameters used in the adaptive schemes and their counterpart non-adaptive schemes.

Parameters	HWFV and FV	MWDG2 and DG2
CFL condition in fully wet domain	0.98	0.33
CFL condition in wet/dry domain	0.5	0.28
The dry threshold parameter (ε_{dry})	1×10^{-5}	1×10^{-5}
The local limiter detection parameter	No need	1.0
The adaptive threshold parameter (ε)	0.01	0.01

Part I- Key issues relevant to both schemes

This part presents two test cases to address two detected issues that are relevant to both adaptive schemes. The first issue is to explore an optimal choice of the adaptive threshold value so as to conserve the quality of the numerical solution. For this purpose, the oscillatory flow in a parabolic bowl test case is considered using only the adaptive HWFV scheme rather than the adaptive MWDG2 scheme. This choice is associated with the importance of maintaining the average coefficients as compared to the slope coefficients. This due to the fact that the adaptivity should be based on the average coefficients, because of their responsibility for the conservative behavior of the solution. Therefore, an optimal threshold value that can preserve the significant average detail coefficients, would be assumed to be sufficient for also dealing with slope detail coefficients that are further involved in the MWDG2 scheme. Based on this, the same threshold value will be used in both adaptive schemes.

The second issue addresses the effect of machine precision (i.e., truncating number of digits from right side of the decimal point) on the adaptivity process. The machine precision generally plays an important role in the computer systems and for many numerical schemes. This is due to the accuracy of computation results that are obtained via using different types of floating-points precision (Goldberg, 1991). Every computer language has a specific representation of the floating-point precision. For example, MATLAB language (all codes here are written in MATLAB) provides several fundamental data types according to the types of numbers (e.g., irrational numbers or rational numbers) that are involved in computations (Moler, 2008).

Since the filter coefficients (used in the multiscale transformation processes) are irrational numbers, the precision digits for the irrational numbers are known as the single and double precisions. These precision digits will affect the conservation of the adaptivity information due to the repetitive use of filter matrices (equations 4.47 and 4.48) to achieve the adaptive solution. Thus this effect will be presented here in terms of the mass conservation in a closed system. For this purpose, a test case that has a dam-break over a triangular hump is used.

5.2 Oscillatory flow in a parabolic bowl

This case is well known and recognized as a challenging test case for numerical models because it involves both moving wet/dry interfaces and it has an uneven topography. It has been extensively used for benchmarking hydraulic models (see, among others, (Kesserwani and Liang, 2012b; Liang and Borthwick, 2009; Xing et al., 2010)). Thus, it is also selected here with aiming to obtain an optimal threshold value in which the adaptive solution can track accurately the transient flow conditions in a system. In addition, there is another reason for choosing this test case over others. This is because the transient flow does not create any significant shock wave throughout the entire simulation, instead of generating the series of the tiny waves at the wet/dry fronts with high velocities. Therefore, any threshold value that is sensitive to those tiny waves will certainly be sensitive to shock waves in a test case generating shock waves throughout the simulation. This test also studies the resolution accuracy and the mesh convergence abilities of the adaptive HWFV scheme.

The test consists of an oscillatory flow taking place inside a parabolic bowl. The transient analytical solution was proposed by Thacker (1981) and can be computed using equations 5.1 to 5.4. The frictionless bed is described by $z(x)$ given in equation 5.5 with constants $h_0 = 10$ and $a = 3000$.

$$\eta\,(x,t) = h_0 - \frac{B^2}{4g}\cos\left(2\,S\,t\right) - \frac{x}{g}\,B\,s\cos\left(S\,t\right) \tag{5.1}$$

$$u\,(x,t) = B\,\sin\left(2\,S\,t\,\right) \tag{5.2}$$

where $B = 5$ is a constant value and S is the frequency that defined as

$$S = \frac{1}{2\,a}\sqrt{8\,g\,h_0} \tag{5.3}$$

and the position of the wet/dry front face (x_{wd}) is given by

$$x_{wd} = -\frac{B\,S\,a^2}{2\,gh_0}\cos(S\,t) \mp a \tag{5.4}$$

$$z(x) = h_0\left(\frac{x}{a}\right)^2 \tag{5.5}$$

Under these conditions, the oscillation period is $T = 1345.94\,s$. The case is simulated on the domain $[-5000\,m, 5000\,m]$ using different computational mesh cells at coarse level (N_0). Simulations are run up to $1.5\,T$. Boundary conditions are irrelevant because the flow never reaches the boundaries. They are set as transmissive boundaries.

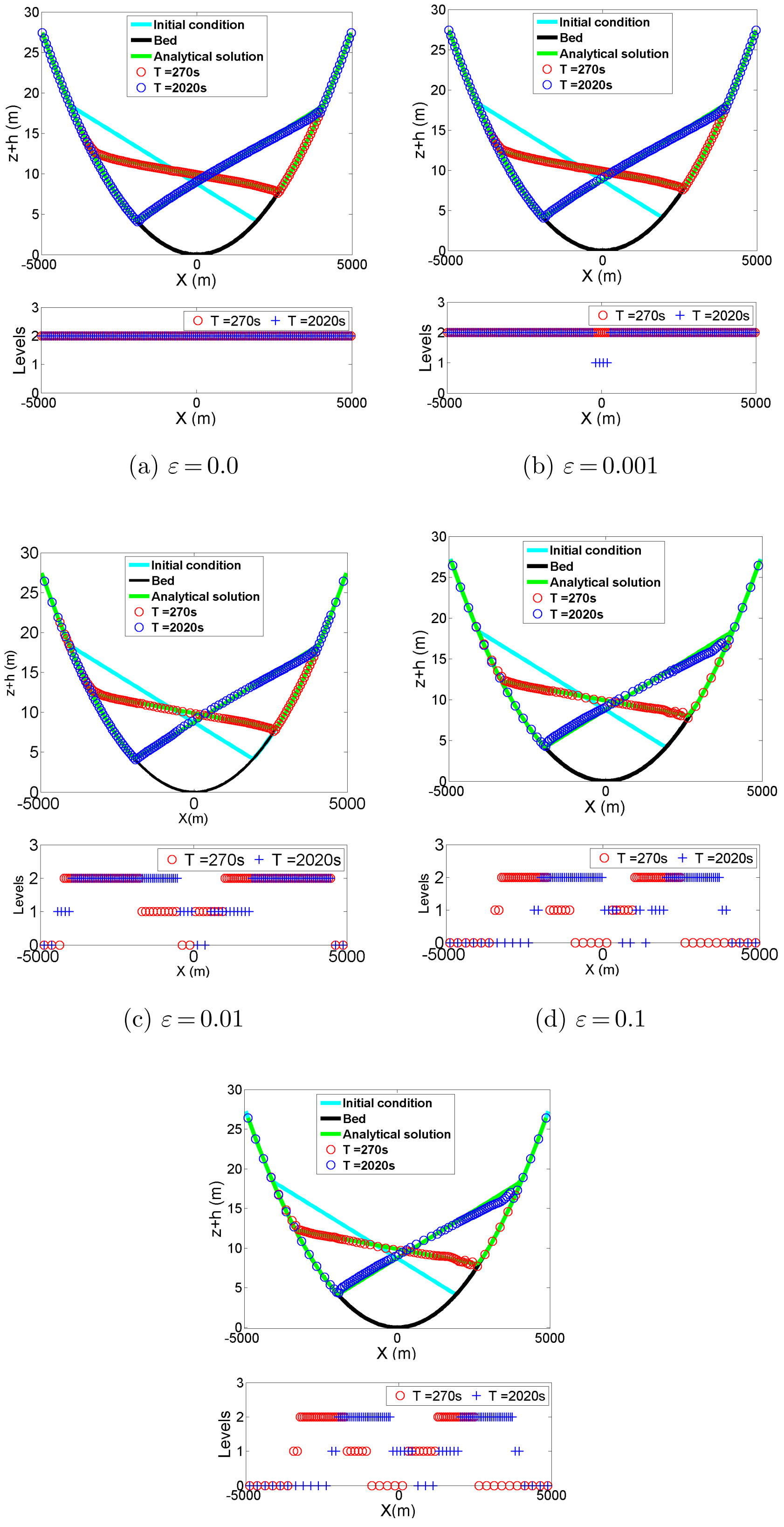
5.2.1 Threshold sensitivity

To understand the effect of the threshold value parameter on the adaptivity process, a baseline mesh with $N_0 = 40$ cells is fixed, while considering the following threshold values $\varepsilon = 0.0$, 0.001, 0.01, 0.1 and 1.0. In Figure 5.1, the numerical water surface profiles at $t = 275\,s$ ($T/5$) and $t = 2020\,s$ ($1.5\,T$) are compared with the analytical solution. As seen in Figure 5.1, the HWFV scheme refines in the region where the wet/dry interface is moving through. Meanwhile, other parts of the domain stay at the coarsest and the intermediate levels of refinement. Furthermore, it can be seen that varying the threshold value leads to different refinement patterns. In particular, it is shown that, as ε increases fewer cells are refined to higher levels during the simulation, since smaller detail coefficients are selectively omitted. In Figure 5.1a, it is clear that the adaptive HWFV scheme activates all the detail coefficients when $\varepsilon = 0.0$ and all the computational cells go to the highest level, thus resulting in a uniform mesh. Figure 5.1b shows an almost uniform mesh prediction when $\varepsilon = 0.001$. Hence, it is not reasonable, in terms of efficiency, to use a threshold value $\varepsilon < 0.001$. A sensible refinement is obtained with $\varepsilon = 0.01$ (Figure 5.1c). For $\varepsilon = 0.1$ and $\varepsilon = 1.0$ (shown in Figure 5.1d and 5.1e respectively) poor predictions at wet/dry interfaces appear which may be caused by the fact that a large threshold value leads to omitting some small detail coefficients, relevant to the wet/dry front, during the promotion process.

5.2.2 Baseline meshes

The choice of a baseline mesh with $N_0 = 40$ is rather arbitrary, and it might be already too fine, or on the contrary, never allow for a fine enough mesh for this test case. This may affect the performance of the adaptivity process. Therefore, the influence of the baseline mesh should be studied. Several baseline meshes at coarse level $N_0 = 20$, 40, 80, 160 and 320 are introduced to address this. The same settings of the threshold value as reported in Figure 5.1 are used. The evolution of the number of active cells is presented in Figure 5.2. The results confirm that all considered combinations of N_0 and ε are able to perform the adaptive solutions. Moreover, when refining the baseline mesh, the HWFV requires a reduction of the threshold value to better perform adaptivity process. This is due to the fact that most of the flow region results in rather smooth solutions; therefore the value of the detail coefficients is small.

In Figure 5.2, N/N_0 reduces as the baseline mesh is refined regardless of the varying threshold value. However, in Figure 5.2d and 5.2e for $\varepsilon = 0.1$ and $\varepsilon = 1.0$, the magnitude of N/N_0 is relatively the same and with values bounded between 1.0 and around 2.5. These values are less when compared to other threshold values, but they strongly influence the quality of the numerical solution. Notably, with $\varepsilon = 0.01$, regardless of N_0, seems to obtain optimal results, in comparison to other threshold values. The value of N/N_0 is bounded between around 1 to 4. This case shows a particular trend of how the pattern of active cells varies with N_0. This trend indicates that $\varepsilon = 0.01$ is the most sensitive to N_0. Owing to this sensitivity, this threshold value allows for a wide, automatic response of the adaptive process



(a) $\varepsilon = 0.0$

(b) $\varepsilon = 0.001$

(c) $\varepsilon = 0.01$

(d) $\varepsilon = 0.1$

(e) $\varepsilon = 1.0$

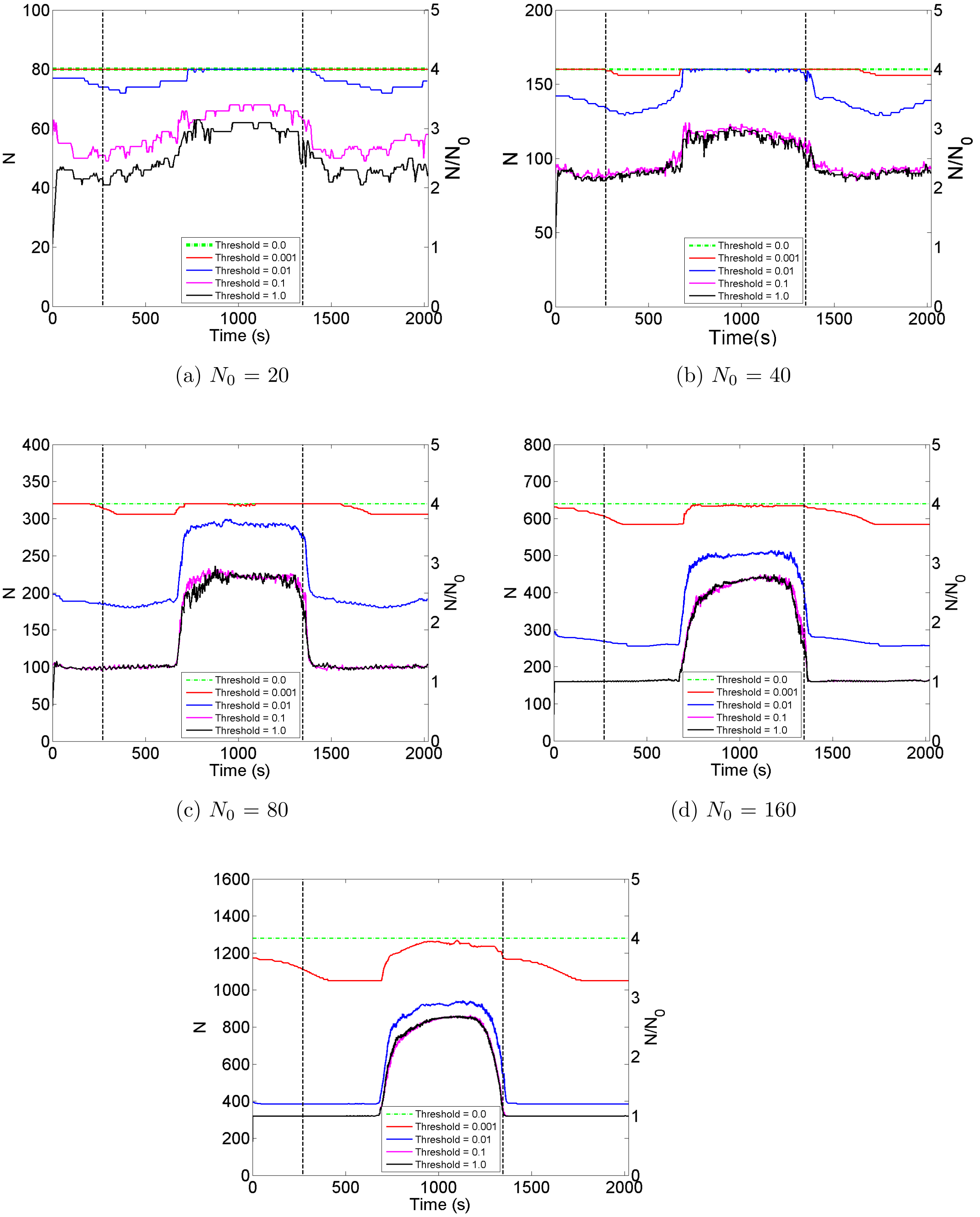
Figure 5.1: Numerical solution against the analytical solution in parabolic bowl flow ($N_0 = 40$), considering different threshold values.

(contrary to, for example $\varepsilon = 0.1$) and therefore is likely to be the best to perform adaptivity in a prompt way. It is clear that an inefficient adaptive process is obtained with $\varepsilon = 0.001$ except in Figure 5.2e, when a very fine baseline mesh is used.

5.2.3 Mesh convergence

Mesh convergence is also studied in terms of the L^1-*norm* as defined in equation. 5.6. Simulations are performed up to $2020\,s$ and for analysis the $t = 270\,s$ $(T/5)$ and $t = 2020\,s$ $(1.5\,T)$ are selected. Several baseline meshes at coarse level $N_0 = 10, 20, 30, 40, 80, 160$ and 320 are introduced with using the same threshold values as defined previously. It can be seen in Figure 5.3 that the behavior of the L^1-*norm* is asymptotic regardless of N_0 and ε. The uniform mesh ($\varepsilon = 0.0$) is taken as a reference curve for comparison. Each point within a curve is associated with each baseline mesh. The difference between the convergence curve for $\varepsilon = 0.001$ and, the reference is relatively small. For $\varepsilon = 0.01$, the convergence is slightly better than all as shown in Figure 5.3a and 5.3b. The large error obtained with $\varepsilon = 1.0$ and $N_0 = 20$, 30 cells and this is due to too few cells being activated during the adaptivity process (over-filtering). Thus, the quality of the numerical solution is affected. Nevertheless, the magnitude of the error becomes smaller as N_0 increases but still remains larger than the error of the reference curve. Furthermore, the better results are obtained for $\varepsilon = 0.1$ but with sensible differences compared to $\varepsilon = 1.0$. The magnitude of the L^1-*norm* is increased from Figures 5.3a to 5.3b. This is merely because of the numerical diffusion which is an anticipated issue for finite volume first-order schemes.

In Figure 5.4, the same analysis, as the one reported in Figure 5.3, is considered to illustrate the relative performance of CPU time (RPCPU). Figure 5.4a shows the ratio of the CPU times of the adaptive schemes to the CPU time obtained from the associated fine uniform reference schemes (i.e., $\varepsilon = 0.0$); while Figure 5.4b further describes the normalized CPU times, which are obtained by dividing all CPU time by the maximum one (i.e., $N_0 = 320$, $\varepsilon = 0.0$). The results show clearly that when $\varepsilon > 0.0$, less time is required for achieving a simulation as the baseline mesh N_0 density increases. They also show that the efficiency of the



(a) $N_0 = 20$

(b) $N_0 = 40$

(c) $N_0 = 80$

(d) $N_0 = 160$

(e) $N_0 = 320$

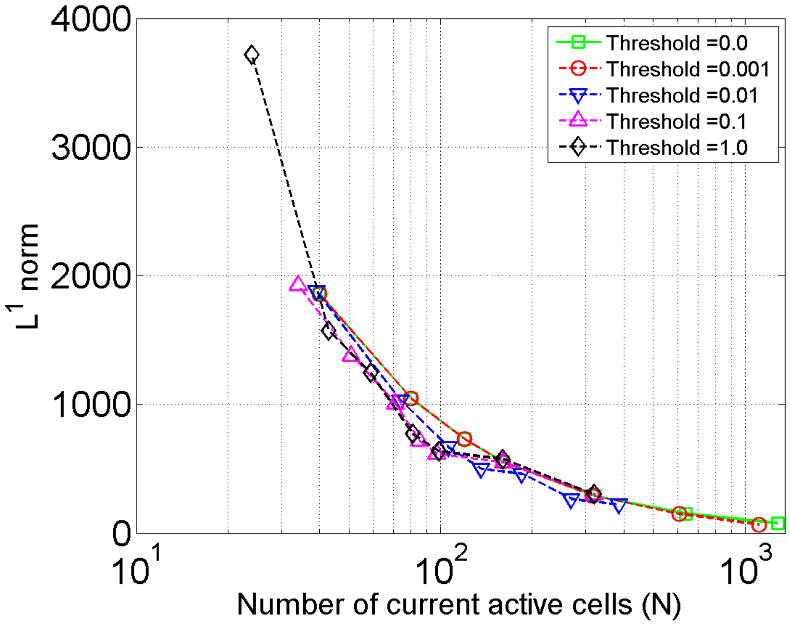
Figure 5.2: Time evolution of active cells for various baseline meshes in parabolic bowl flow.

adaptive HWFV schemes is near their equivalent uniform mesh FV schemes when the baseline mesh has a size $N_0 \leq 40$ and despite the choice of ε.

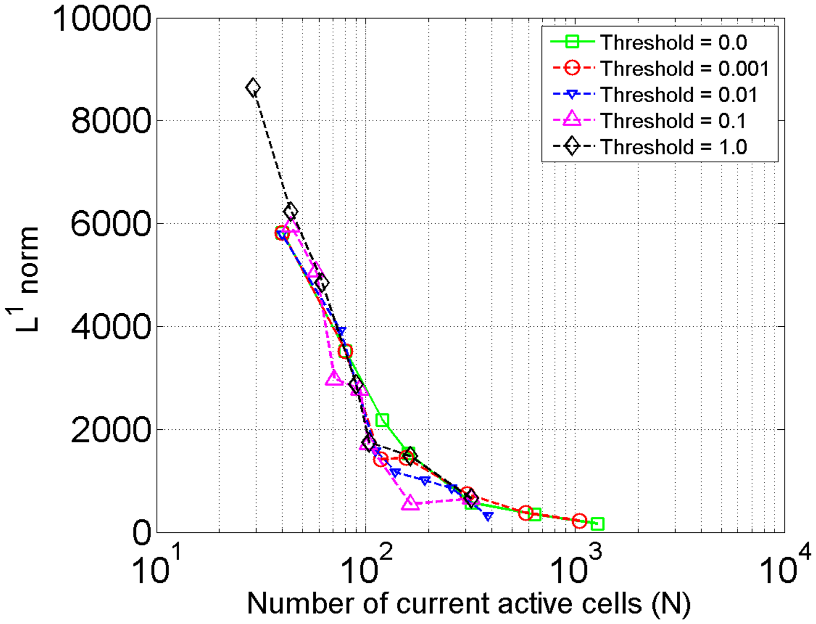
For $\varepsilon\ = 0.001$, the RPCPU and normalized CPU time are noted to be inefficient despite the choice of the baseline mesh N_0. In contrast, for $\varepsilon = 0.01$, 0.1 and 1.0 they start to significantly decrease in proportion with an increase in the density of the baseline mesh N_0. However, for $\varepsilon\ = 0.1$ and 1.0, the RPCPU tend to remain close for $N_0 \geq 40$; whereas, with $\varepsilon\ = 0.01$ the RPCPU shows consistent decrease in line with the refinement of the baseline mesh N_0; this suggests that a threshold value of $\varepsilon\ = 0.01$ enables best selection among the magnitude of the detail coefficients, and so allows optimal efficiency and accuracy in the context of the proposed HWFV scheme for a baseline mesh around 40-100 cells.

$$L^1 - norm = \sum_{i=1}^{N_0} \sum_{n=0}^{n_l} \sum_{j=0}^{2^n-1} |dx_{i,j}^n(h_{i,j}^n - \hat{h}_{i,j}^n)| \tag{5.6}$$

Based on the aforementioned conclusions in the previous sections, and also for comparison purpose, the $\varepsilon =\ 0.01$ is used as a default value in all test cases belong to both proposed adaptive schemes.

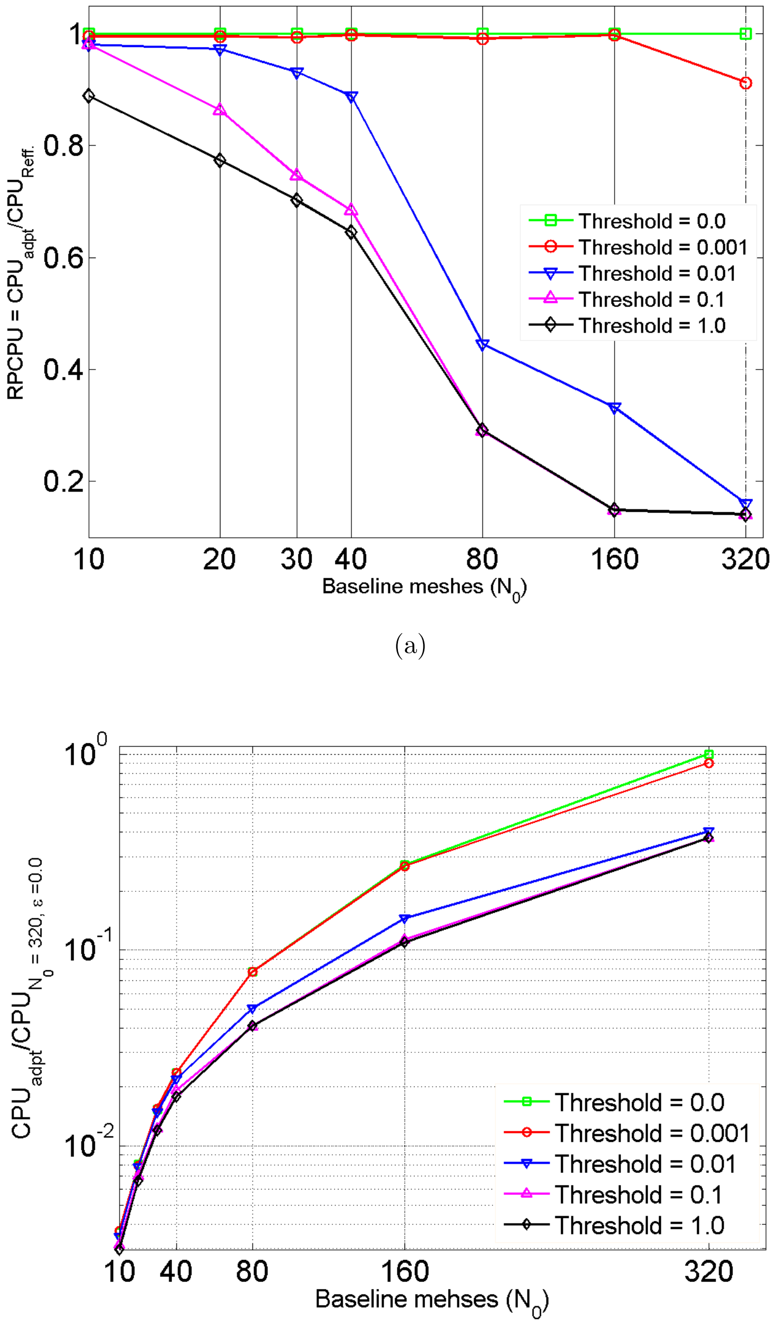


(a) $t = 270\,s$



(b) $t = 2020\,s$

Figure 5.3: Comparisons of L^1-*norm* for parabolic bowl. Each highlight point is associated with the initial cell number at coarse level.



(a)

(b)

Figure 5.4: Comparisons of the relative CPU time for parabolic bowl.

5.3 Effect of machines precision on the adaptive schemes output

This test case studies how the two chosen precisions for the filter coefficients affect the adaptive scheme output, in terms of conserving the total numerical mass in a closed system. Therefore, a hypothetical dam-break over a triangular hump is set up, as shown in Figure 5.5. The length of the horizontal flume and the location of the dam from the upstream end are $38\,m$ and $15.5\,m$, respectively. A reservoir with a water surface elevation of $0.75\,m$ is located upstream from the dam and the rest of the domain is assumed to be dry. Mesh resolution at the coarse level is consisted of 13 cells for both adaptive schemes. Wall boundary conditions are set at both ends of the channel to conserve the total mass in the system. The dam is assumed to fail at $t\ = 0\,s$, causing violent wave preparations; namely, the wetting front rushes into the floodplain, over-tops and interacts with the obstacle creating a reflected wave that will be reflected again by the boundary walls. During the simulation, the total mass of water at time t (M_t) is computed and compared with the real physical mass (M_R) which is considered as a reference mass for all computations.

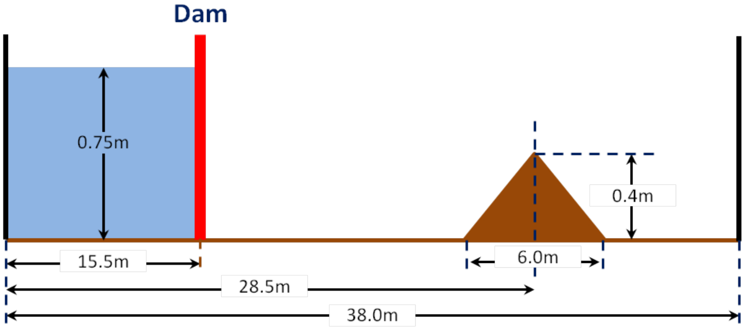


Figure 5.5: Dam-break over a triangular hump.

The adaptive computations are performed with $n = 3$ and the simulation time is set to $300\,s$. The relative mass error is computed by invoking equation 5.7 and used twice independently, for each of the adaptive schemes (i.e., HWFV and MWDG2). In the first adaptive computations, the single precision for the filter coefficients (associated with each adaptive schemes) is fixed. While, in the second adaptive computations, the double precision is fixed throughout the entire simulations.

$$RME = \frac{\mid M_R - M_t \mid}{M_R} \tag{5.7}$$

Figure 5.6 shows the results of the *RME* profiles. Note that both double precision simulations provide relatively better preserving of the real physical mass compared to the counterpart single precision. This implies that the computed values of the conservative variables involved in the multiscale transformation (i.e., equations 4.47 and 4.48) are affected by the precision of the filter coefficients. Irrespective to the accuracy of both schemes in reproducing the real mass (i.e., the magnitude value of *RMEs*). The difference between the *RME* profiles obtained from the single and double precision for the MWDG2 and the HWFV schemes are (1.099×10^{-3}) and (2.299×10^{-4}), respectively. It is clear that the difference in the MWDG2 computation is significantly bigger by approximately fivefold than the difference in the HWFV computation. This is reasonable due to the bigger size of the filter matrices that are used in the multiscale transformation. In addition, the results suggest greater concern regarding the precision of filter coefficients when the adaptive formulation seeks a higher order of accuracy. This is because more irrational filter coefficients will be involved in the adaptivity process (see Subsections 3.3.4). Thus, to minimize this effect, it is suggested to perform the initial projection of the conserved variable solutions with fewer cells (i.e., choose coarse baseline mesh as much as possible). This is attributed to decreasing the total number of calls of the multiscale transformations which results to reduce the involvement of filter coefficients in computations. Arguably, this is a reasonable suggestion and it is acceptable with the conceptual framework of the adaptive formulation. Because the adaptive formulation is fully local and the increasing of the accuracy order is associated with the number of degrees of freedom over the computational cell. For example, the HWFV formulation needs one degree of freedom per cell, while the MWDG2 formulation needs two degrees of freedom per cell. Therefore, the initial projection will not be affected by following the aforementioned suggestion.

Based on the *RMSEs* results, the multiscale transformations must be performed by choosing filter coefficients with a minimum setting to the double precision.

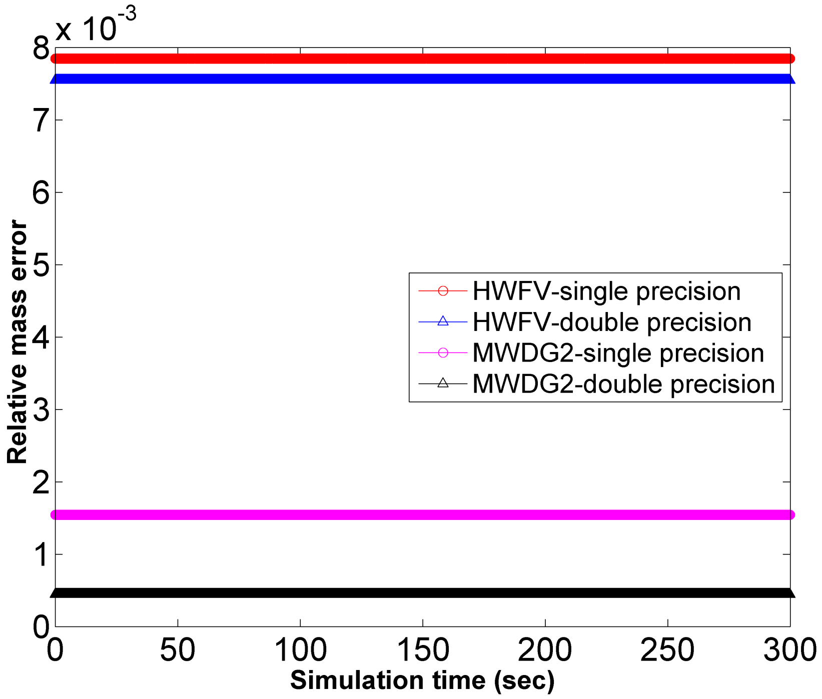


Figure 5.6: Evolution of the RME in the dam-break over the triangular hump test case considering the single and double-precision floating point arithmetic.

Part II- Validations and comparisons of the adaptive schemes

This part presents six well-known benchmark test cases for validations and comparisons of both adaptive schemes. Two of the test cases are idealized dam-break cases over frictionless beds (regular and irregular) and the third test case considers the quiescent flow over an irregular bed. The forth and fifth test cases consider steady flow over a hump (transcritical flow with shock and supercritical flow performed only for the HWFV scheme) and the last test case is a steady hydraulic jump with friction in a rectangular channel.

In the aforementioned test cases, two principles are made for the discretisation of the baseline mesh for the computational domains of both adaptive schemes. These principles are based on the main scope of test cases :

1. when the test case seeks to show the quantitative comparisons, the mesh resolutions is taken to have the same number of cells. This principle will be used in test cases that are given in Sections 5.4 and 5.6.
2. when the qualitative comparisons are sought in the test cases, it is interesting to have the same degrees of freedom for both adaptive schemes at the highest resolution level. This principle will be used in test cases that are given in Sections 5.5, 5.7 and 5.9.

5.4 Idealized Dam-break

The purpose of this case is to test the capability of the HWFV and the MWDG2 schemes to efficiently and accurately solve the homogeneous shallow water equations with discontinuities including a shock. The solutions are compared with the exact solution, using the *RMSE* and maximum error metrics. A 1D channel with a horizontal frictionless bed is considered. The length of the channel is $2000\,m$ and an imaginary dam is located at $1000\,m$ from the upstream end. Initially, the upstream water level is $20\,m$ whereas at the downstream end, the water level is $5.0\,m$. The initial discharge is set to zero in every cell. Boundary conditions, although set to be numerically transmissive, are effectively irrelevant in this test case as the propagating wave does not reach the boundary.

At the instant of dam failure, water is released producing a shock wave travelling downstream meanwhile a rarefaction wave is formed and propagating upstream. The computational domain of the adaptive schemes at the coarse level ($n\ =0$) is discretised to 71 uniform cells and the computational models are run up to 40 s after the dam break. Since both adaptive schemes allow for up to three levels of mesh refinement (i.e., $n\ =\ 3$), the size of the reference fine uniform mesh is 568 cells.

5.4.1 HWFV solution

The numerical results are illustrated in Figure 5.7 and they give a satisfactory solution compared with the analytical solution. The highest level of resolution is noted to be reached at the shock wave and the kink at the tail of rarefaction wave. The other zones of the rarefaction wave are achieved with the intermediate resolution of level 1 and level 2. Meanwhile, the rest of the domain, where the solution is smooth, the HWFV scheme has retained the baseline coarse level. In this test case, the adaptive solution required a maximum of 124 cells (22 % of the reference mesh).

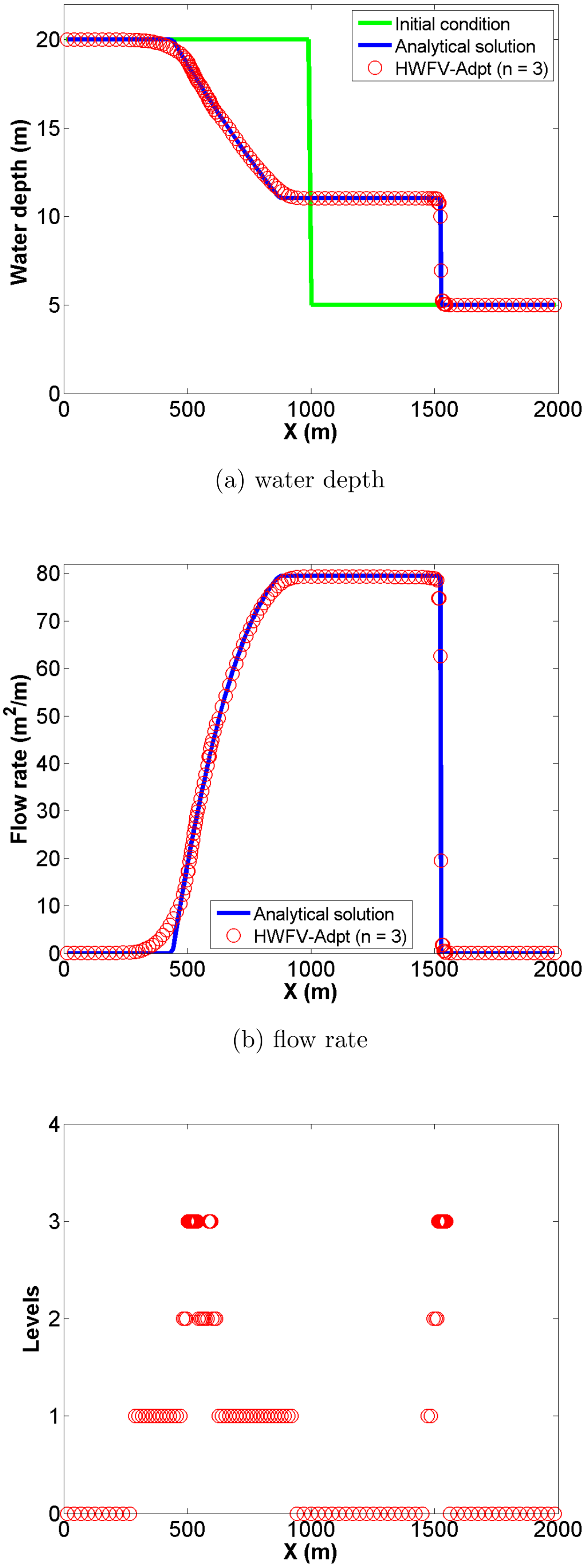
Further quantitative analysis is performed via calculating the $RMSE$ for both the water surface and the flow rate,i.e.,

$$RMSE_h = \sqrt{\left(\frac{\sum_{i=1}^{N_0}\sum_{n=0}^{n_l}\sum_{j=0}^{2^n-1}\left(h_{i,j}^n - \hat{h}_{i,j}^n\right)^2}{N}\right)} \tag{5.8}$$

$$RMSE_q = \sqrt{\left(\frac{\sum_{i=1}^{N_0}\sum_{n=0}^{n_l}\sum_{j=0}^{2^n-1}\left(q_{i,j}^n - \hat{q}_{i,j}^n\right)^2}{N}\right)} \tag{5.9}$$

In which N is the total number of active cells forming the adaptive mesh, $h_{i,j}^n$ and $q_{i,j}^n$ are the numerical results and $\hat{h}_{i,j}^n$ and $\hat{q}_{i,j}^n$ are the reference data (from the analytical solution).

Figure 5.8 compares the $RMSE$ of the water depth and the flow rate for different resolution levels throughout the simulation time. The $RMSEs$ for each resolution show high fluctuation which decreases in error magnitude with an increasing



(a) water depth

(b) flow rate

(c) levels

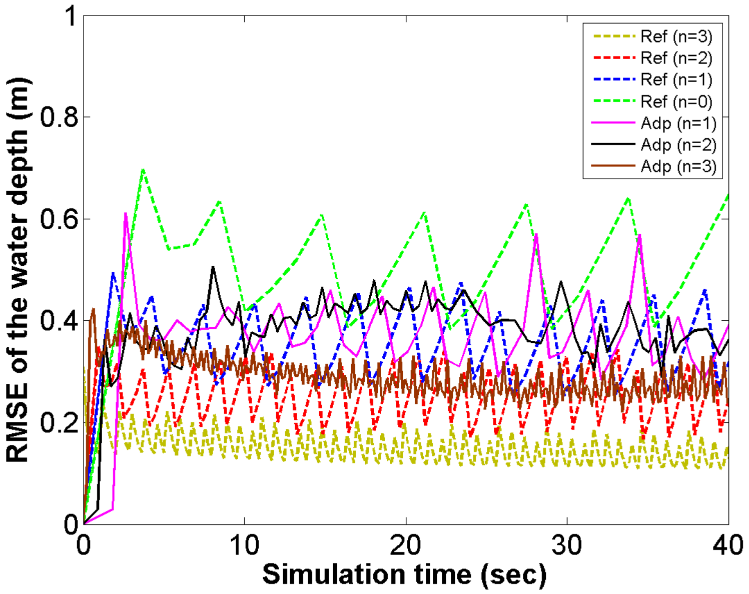
Figure 5.7: HWFV adaptive numerical solution for the idealized dam-break flow.

in baseline mesh resolution. They also show a periodic pattern. The $RMSE$ profiles resulting from the HWFV scheme are bounded between the coarse and fine uniform solution which means that the model is not introducing additional errors to the numerical solution during the adaptivity process. Furthermore, the fluctuation of the $RMSE$ profile of the adaptive level $n = 3$ is less fluctuating compared with the others, which is expected due to grid convergence proprieties. To confirm that the obtained $RMSEs$ are meaningful in terms of comparison, another quantitative analysis is considered via calculating the maximum depth error which is governed by the highest resolution level (due to the presence of shock). These error trends (see Figure 5.11) are more logical in this sense, as it can be seen that: **i)** any of the adaptive models at resolution $n = 1$ and 2 show comparable error range to the reference uniform counterpart and **ii**) the variation of the adaptive errors are consistent level-wise.

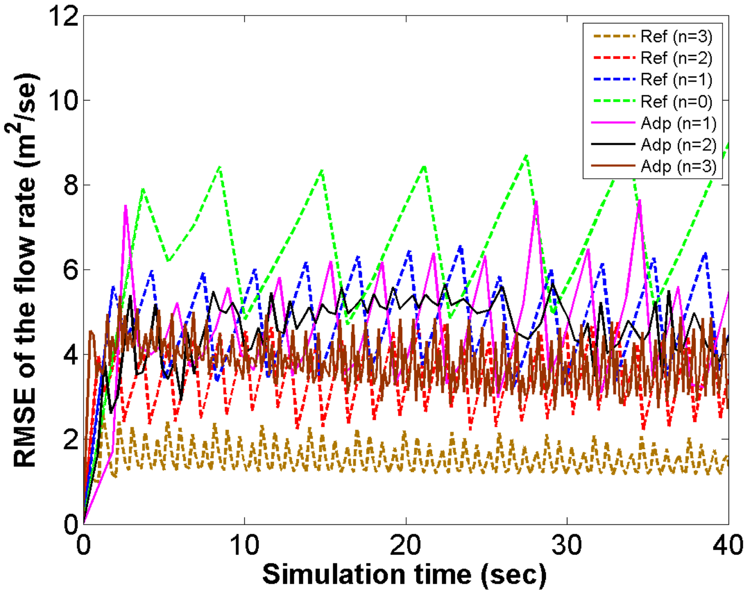
5.4.2 MWDG2 solution

The setting used for the HWFV scheme is also considered in MWDG2 scheme. This is for the sake of the comparisons between the quantitative results which are computed for both adaptive schemes. It can be seen in Figure 5.9 that the most of the domain is retaining the intermediate resolution of level 1 and level 2. Whereas the highest level of resolution $n = 3$ is noted to be reached at the shock wave and the kink at the tail of rarefaction wave. This peak in resolution is noted to track the evolution of shock wave and relatively coarse level of resolution elsewhere. Which is expected due to the slow dynamics of the flow rate. In Figure 5.9a, there are some variations trailing the shock. These variations are not generated by the adaptive scheme because they had been observed in the computations of non-adaptive schemes. These are due to chosen limiting strategy (see Subsection 4.3.3) that allows very small variations around the shock to avoid unnecessary limiting in smooth regions. The adaptive solution required a maximum of 119 cells which represents only 20.9 % of the highest resolution uniform mesh.

In Figure 5.10, the $RMSEs$ for both the water surface and the flow rate are presented to identify that the extent to which the adaptive process are preserving the accuracy of the associated uniform mesh model. Note that the t-axis is limited



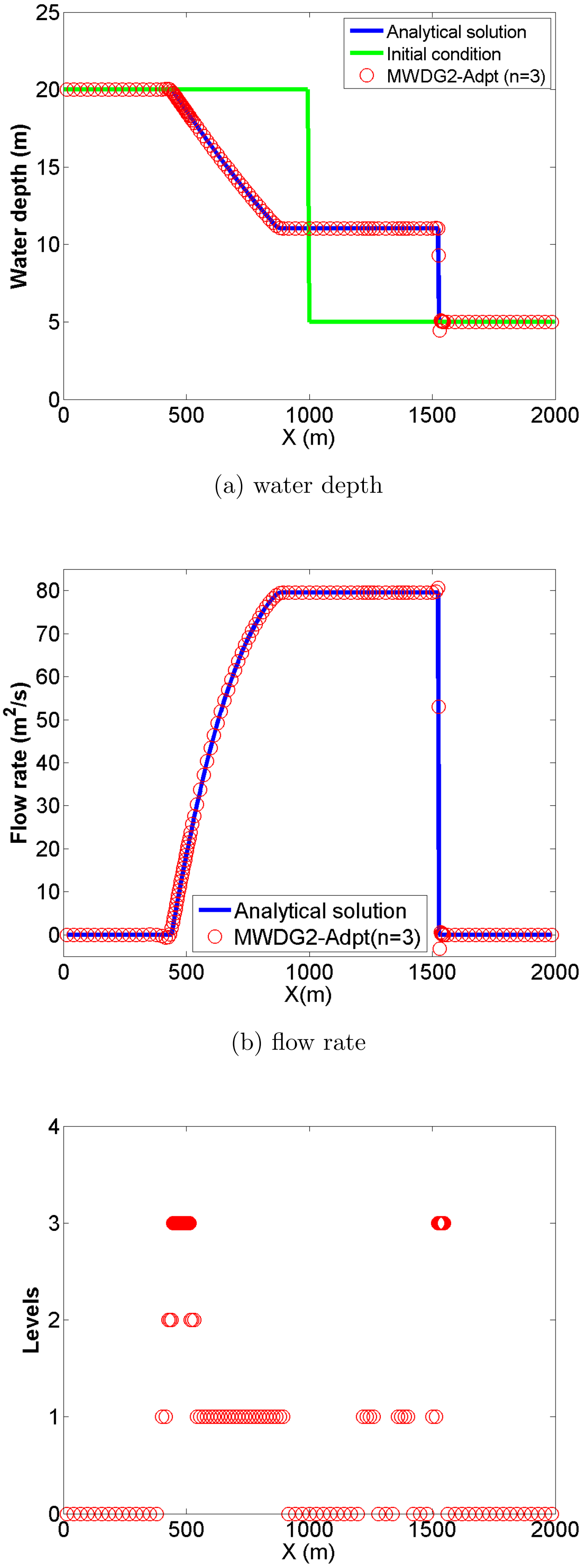
(a) water depth



(b) flow rate

Figure 5.8: RMSE evolution for idealized dam-break test case using HWFV scheme.

between $0\,s$ to $15\,s$ for clarity purpose. Generally speaking, the behavior of the $RMSEs$ have the same fluctuation and the adaptive $RMSE$ profiles are bounded between the coarse and fine uniform solution which confirms that the adaptive scheme is not introducing any error into the numerical solution. It is obvious that the maximum magnitude of the error is obtained at the shock. Therefore, it is necessary to consider also the maximum error of water depth (infinity norm) to understand more the logical sense of this errors. In Figure 5.12, the maximum water depth error shows that the variation of any adaptive errors n are consistent with the reference uniform counterpart. This is reasonable, since the adaptive technique is deliberately designed to more accurately track shocks.

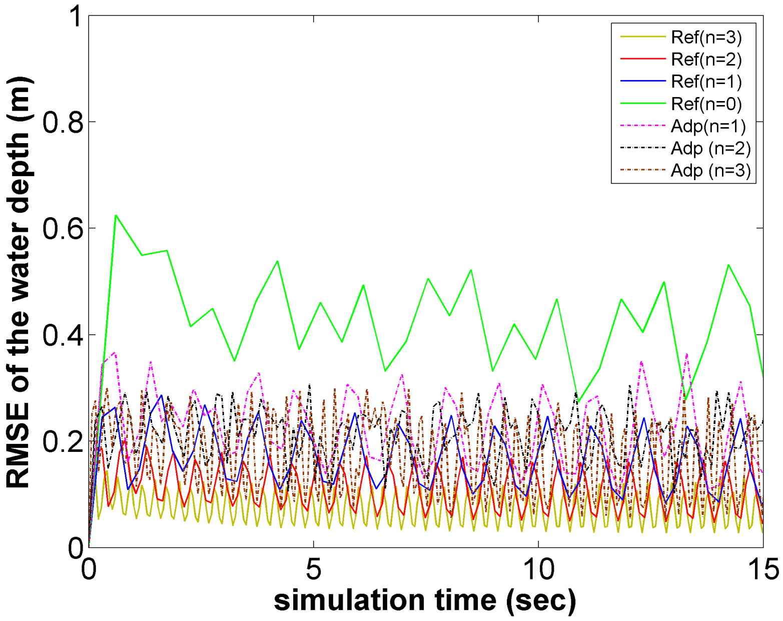


(a) water depth

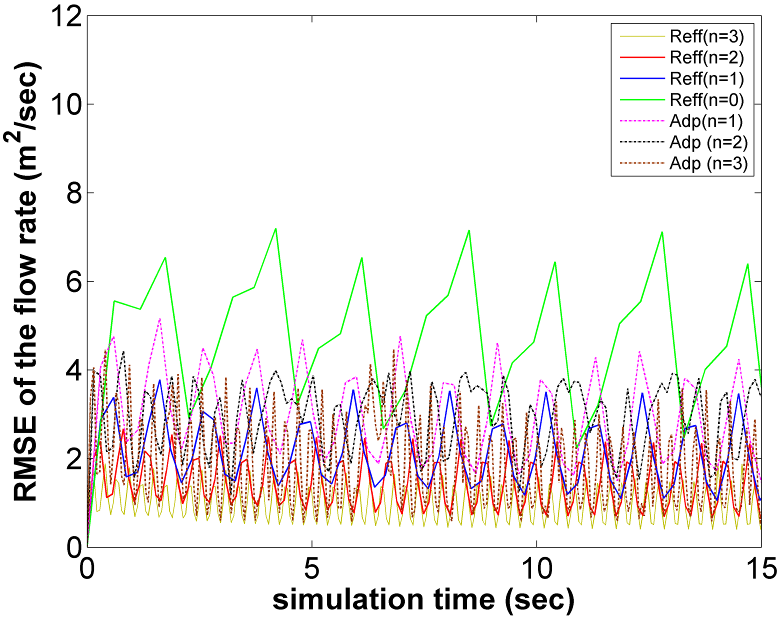
(b) flow rate

(c) levels

Figure 5.9: MWDG2 adaptive numerical solution for the idealized dam-break flow.



(a) water depth



(b) flow rate

Figure 5.10: *RMSE* evolution for idealized dam-break test case using MWDG2 scheme.

.

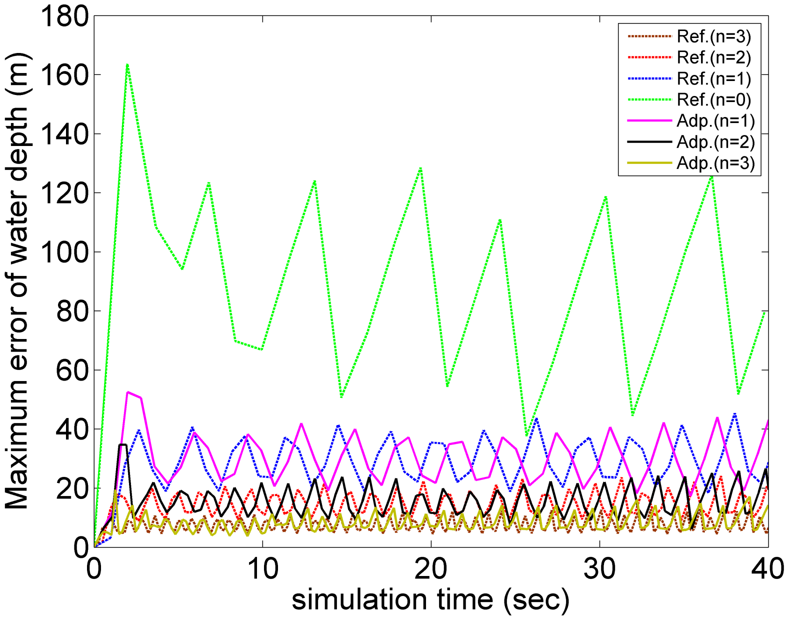


Figure 5.11: HWFV max water depth error evolution for the dam-break case.

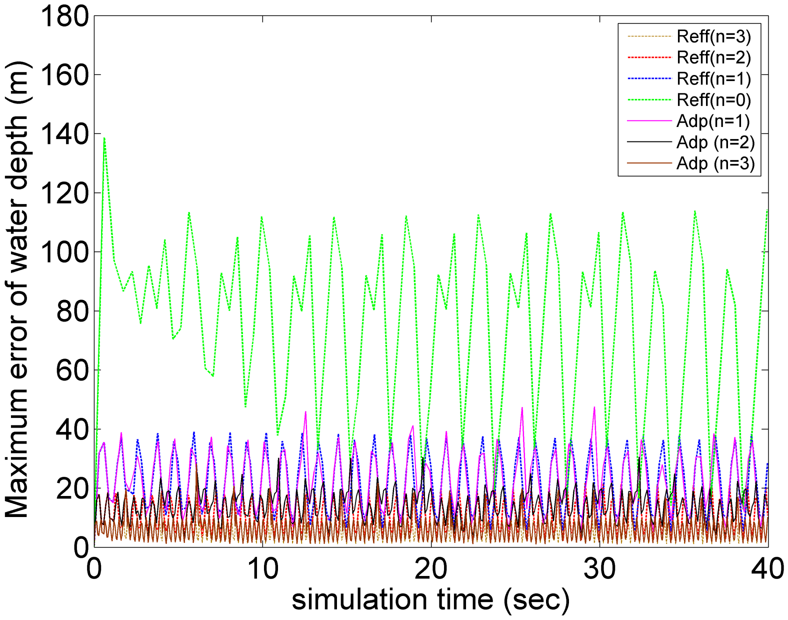


Figure 5.12: MWDG2 max water depth error evolution for the dam-break case.

5.4.3 Comparisons

The numerical results in both schemes generally show a good agreement with the analytical solution. In particular, in the case of MWDG2 scheme which the shock and the tail of the flow rate are well captured compared to the HWFV scheme (see Figures 5.7 and 5.9). Moreover, the new schemes are able to obtain the level of resolution according to the dynamic of the flow while significantly reducing the number of cells and therefore the computational effort is reduced. As for the calculated error (i.e. $RMSE$ and maximum depth error), they are much smaller for MWDG2 scheme than the error of HWFV scheme. However, the adaptive *RMSE* profiles in both schemes have the same patterns in terms of their variations between the coarse and fine uniform solutions. But such variations are obtained from the adaptive $RMSE$ profiles which is expected because an adaptive schemes at level n is set to further allow lower resolution down to the baseline resolution $n = 0$. To figure out more the behavior of the computed errors, another quantitative analysis is considered via tabulating the mean and standard deviation of the $RMSEs$ and the maximum depth error (see tables 5.2 , 5.3 and 5.4). These results show decreasing trend for the means accompanied by a faster reduction in their SDs with more profound refinement levels in adaptive schemes. Importantly, this reduction is consistent in which the mean values of the adaptive schemes at level (n) are somewhere in between the mean values of the uniform schemes at level $(n-1)$ and level (n).

Based on the results, the MWDG2 scheme shows better performance in this test case in terms of reducing the magnitude of errors. The both adaptive frameworks perform the adapivity process without introducing any additional errors to the numerical solutions and they are able to sensibly decide resolution level for highly dynamic break flow.

Table 5.2: $RMSE$ for Dam-break test case - water depth.

	FV		DG2	
	mean	SD	mean	SD
Uni. mesh ($n = 0$)	0.497	0.141	0.441	0.135
Uni. mesh ($n = 1$)	0.381	0.096	0.205	0.061
Uni. mesh ($n = 2$)	0.249	0.058	0.152	0.054
Uni. mesh ($n = 3$)	0.157	0.032	0.111	0.033
Adpt. mesh ($n = 1$)	0.389	0.100	0.209	0.066
Adpt. mesh ($n = 2$)	0.362	0.074	0.173	0.058
Adpt. mesh ($n = 3$)	0.251	0.043	0.158	0.055

Table 5.3: $RMSE$ for Dam-break test case-flow rate.

	FV		DG2	
	mean	SD	mean	SD
Uni. mesh ($n = 0$)	6.588	1.952	4.899	1.992
Uni. mesh ($n = 1$)	4.731	1.326	2.191	0.739
Uni. mesh ($n = 2$)	3.524	0.822	1.773	0.587
Uni. mesh ($n = 3$)	1.617	0.351	0.945	0.424
Adpt. mesh ($n = 1$)	4.539	1.434	2.770	1.040
Adpt. mesh ($n = 2$)	4.531	0.902	2.159	0.569
Adpt. mesh ($n = 3$)	3.770	0.659	1.538	0.577

Table 5.4: Maximum error for Dam-break test case-water depth.

	FV		DG2	
	mean	SD	mean	SD
Uni. mesh ($n = 0$)	87.874	35.612	76.366	32.261
Uni. mesh ($n = 1$)	29.621	9.176	22.678	10.741
Uni. mesh ($n = 2$)	14.480	4.366	11.826	5.569
Uni. mesh ($n = 3$)	6.552	2.333	5.364	2.818
Adpt. mesh ($n = 1$)	29.87	9.74	22.300	10.341
Adpt. mesh ($n = 2$)	15.901	5.698	14.840	5.199
Adpt. mesh ($n = 3$)	5.752	3.022	5.491	3.875

5.5 Quiescent flow over an irregular bed

The quiescent flow (well-balancing) property was introduced first by Bermudez and Vazquez (1994b) and has been broadly used by numerical scheme developers (see, among others, (Rogers et al., 2003; Aureli et al., 2008) and (Caleffi and Valiani, 2012)). Therefore a test case which takes into consideration a differentiable and non-differentiable topography, defined by the equation 5.10 and shown in Figure 5.13a, is introduced to achieve the well-balancing property for both the adaptive schemes.

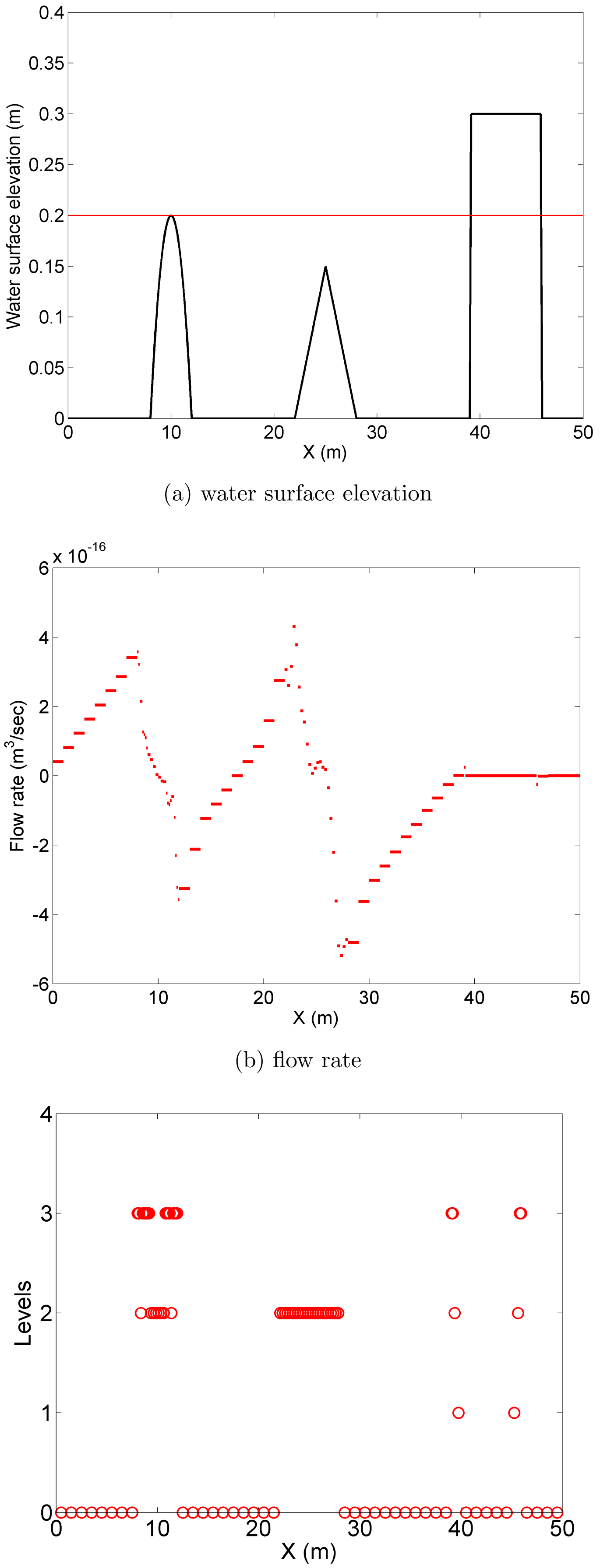
In addition, the adaptive schemes are devoted to resolve complicated flows over topography through considering three different cases: (**i**) wet/dry fronts case (i.e. at both sides of the piecewise constant elevation); (**ii**) critical wet case (i.e. $h = 0\,m$ at the peak of a bump) and (**iii**) fully wet (i.e. the computational domain is wet apart from regions (**i**) and (**ii**)).

$$z(x) = \begin{cases} 0.2\left(0.05\,(\,x-10)^2\right) & \text{if } \; 8 < x < 12 \\ 0.05\,x - 1.1 & \text{if } \; 22 \le x < 25 \\ -0.05\,x + 1.4 & \text{if } \; 25 \le x \le 28 \\ 0.3 & \text{if } \; 39 < x < 46 \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

The computational domains at the coarse level for the HWFV and MWDG2 schemes is comprised of 50 cells and 25 cells, receptively. The computational models are run up to $1000\,s$ with $n = 3$. An upstream boundary condition is imposed by zeroing flow rate whereas a downstream boundary condition is set to $0.2\,m$. These imposed conditions are also used to initialize the simulation.

Figures 5.13 and 5.14 show the numerical results of the adaptive scheme respectively, considering the full solution of the FV and DG. The resolution level throughout the domain in both adaptive schemes is almost the same in the wet/dry fronts case, see Figures 5.13c and 5.14c. Whereas at the triangular hump, the HWFV scheme retains the resolution level at 2, but the resolution level is varied between $n = 2$ and $n = 3$ in the MWDG2 scheme. This is expected to be caused by the extra slope coefficient values that are more sensitive and are therefore activated by the adaptive scheme, see Figure 5.14b. The adaptive schemes succeed in capturing, the critical wet region at the peak of the bump, but the MWDG2 scheme refines more cells to the highest resolution level. This is attributed to the involvement of the slope details coefficients in the adaptive solutions.

The full adaptive solutions of the free surface elevation preserve the motionless steady state throughout the simulations (see Figures 5.13a and 5.14a) and, the zero flow rate in both adaptive computations is accurately replicated within the range of machine precision (1×10^{-16}) (see Figures 5.13b and 5.14b).

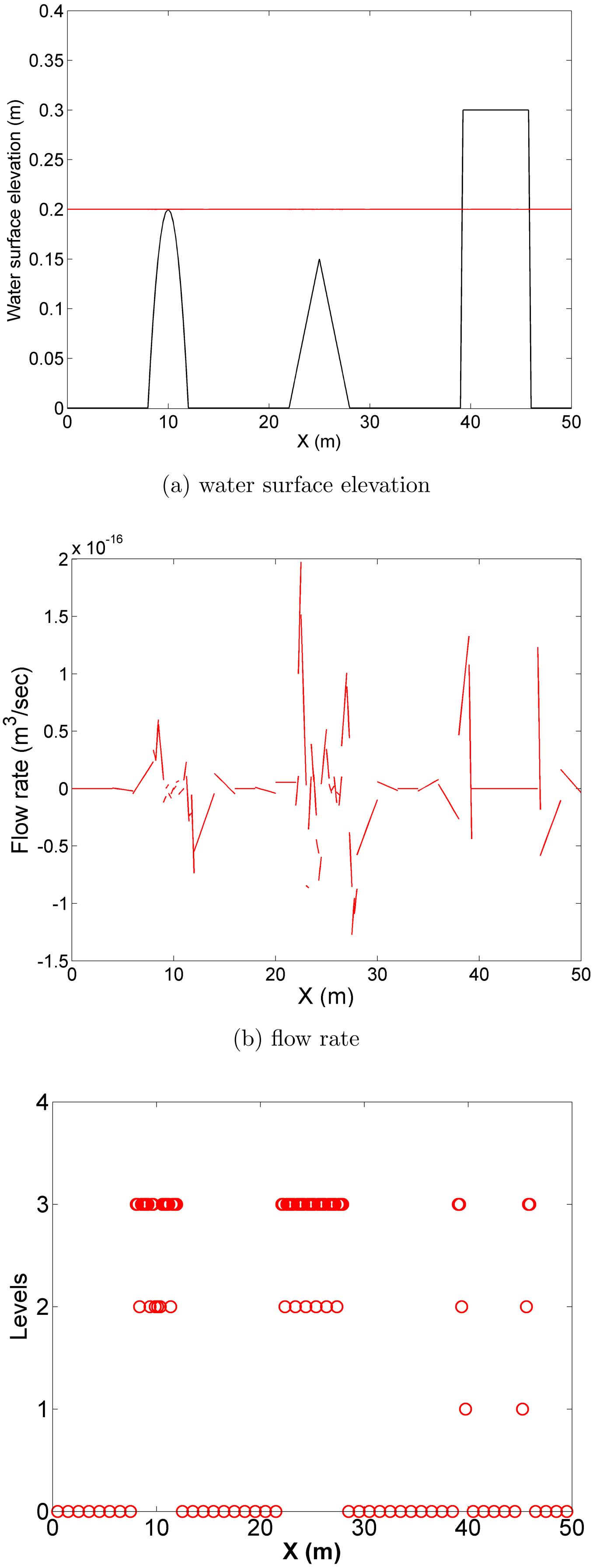


(a) water surface elevation

(b) flow rate

(c) level

Figure 5.13: Quiescent flow with wet/dry fronts for HWFV scheme.



(a) water surface elevation

(b) flow rate

(c) level

Figure 5.14: Quiescent flow with wet/dry fronts for MWDG2 scheme.

5.6 Dam-break over a triangular hump

This test case is employed to verify the capability of both adaptive schemes in retaining numerically the total mass in the system. This justifies that the adaptivity process does not introduce or lose the numerical mass even in the presence of moving shocks and wet/dry fronts. The description of the test case, including the initial and boundary conditions was given in section 5.3. For adaptive schemes, the mesh resolution at the coarse level is consisted of 13 cells. The adaptive computations are performed with $n = 3$ and, the simulation time is set to $1000\,s$.

Since the system is closed, the initial mass water should be conserved. During the simulation, the total mass of water at time t (M_t) is computed and compared with the total initial discrete mass (M_0) according to the RME below:

$$RME = \frac{\mid M_t - M_0 \mid}{M_0} \tag{5.11}$$

The results in Figure 5.15 show that both adaptive schemes conserve the initial total mass in the closed system and the magnitude of the $RMEs$ are within the range of machine precision (10^{-15}) throughout the entire computational time. However, since in both adaptive schemes the mesh resolutions are not fixed, the real physical mass (M_R) is also used, as a reference similar to the test 5.3, to show the ability of these schemes to preserve the same amount of mass as compared to the equivalent uniform high resolution level (i.e., $n = 3$); this is performed via invoking the equation 5.7. In Figure 5.16, the results show that the RME profiles for both the adaptive mesh levels and their counterparts uniform levels are the same. This implies that both schemes are not introducing any additional mass error beyond the capability of the discretization relative to the fine reference uniform scheme.

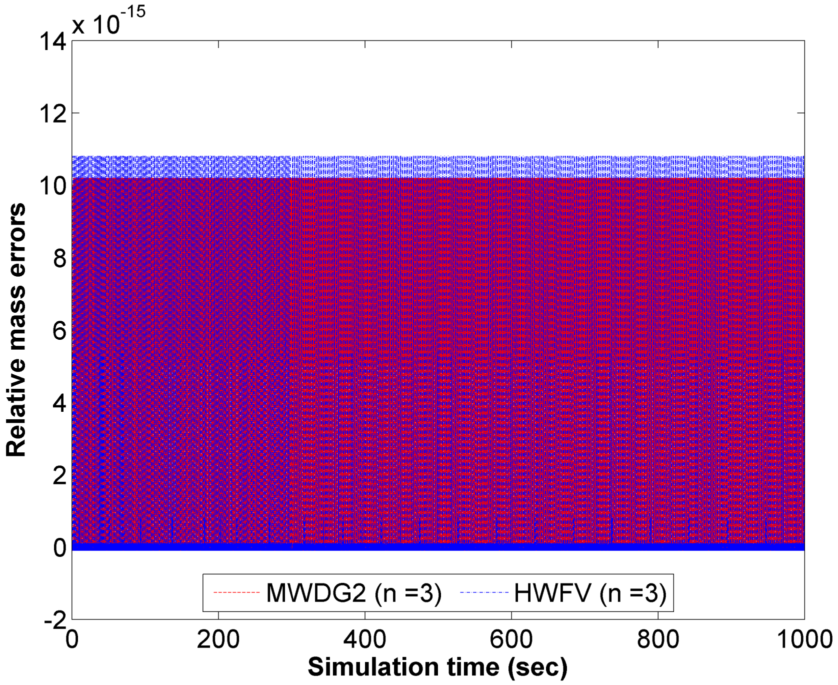


Figure 5.15: RME evolution for dam-break over a triangular hump (compared with the projected mass $t = 0\,s$)

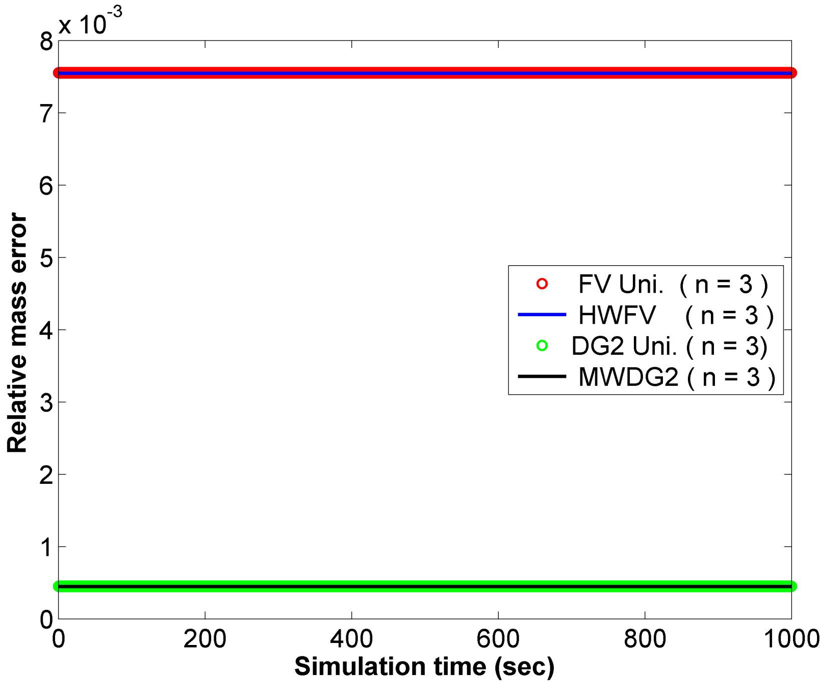


Figure 5.16: RME evolution for dam-break over a triangular hump (compared with the physical real mass).

5.7 Transcritical steady flow over a hump

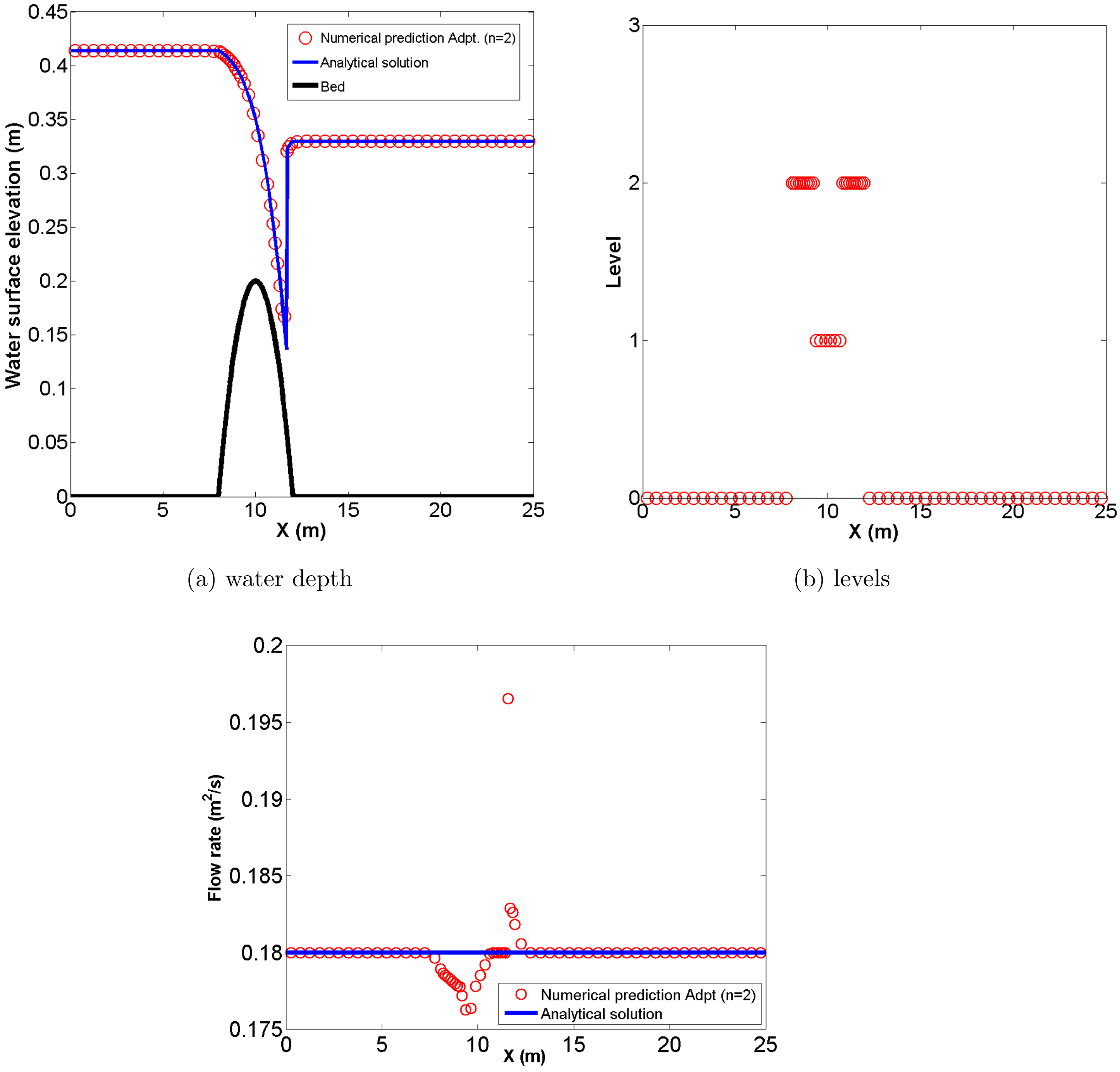
This case is employed to test the performance of both adaptive schemes in reproducing a steady transcritical flow over non-uniform topography in a frictionless rectangular channel 1 m wide and a 25 m long. The bed is defined by

$$z(x) = \begin{cases} 0.2 - 0.05(x-10)^2 & \text{if } 8 \leq x \leq 12 \\ 0 & \text{otherwise} \end{cases} \tag{5.12}$$

The analytical solution was supplied by SWASHES (2014). The initial water surface and flow rate per unit width were set to $h+z = 0.33\,m$ and $q = 0.18\,m^3/(s.m)$, respectively. Physical boundary conditions consisted of the steady discharge at inflow and the initial water level at outflow. The adaptive computations for both schemes were performed with $n = 2$ levels.

5.7.1 HWFV solution

The computational domain at the coarse level comprised of 50 uniform cells. The solution converges to the steady state of the transcritical flow at $t = 170\,s$. Figure 5.17 presents the numerical results of the adaptive scheme compared with the analytical solution. The numerical water surface profile shows a good agreement with the analytical solution. Furthermore, the flow rate stays constant apart from small variations caused by the presence of the hydraulic jump. Arguably, this is a known deficiency in the standard finite volume schemes which was reported by Garcia-Navarro and Vazquez-Cendon (2000); Toro and Garcia-Navarro (2007) and thus these variations are not related to the proposed wavelet-based adaptivity . In this test case, the majority of the domain features required the coarse resolution level mesh except at the hump, which dictated local level 1 of refinement from the onset and at discontinuities (i.e., starting kink of the transcritical transition and shock) where level is refined to highest resolution.



(a) water depth

(b) levels

(c) flow rate

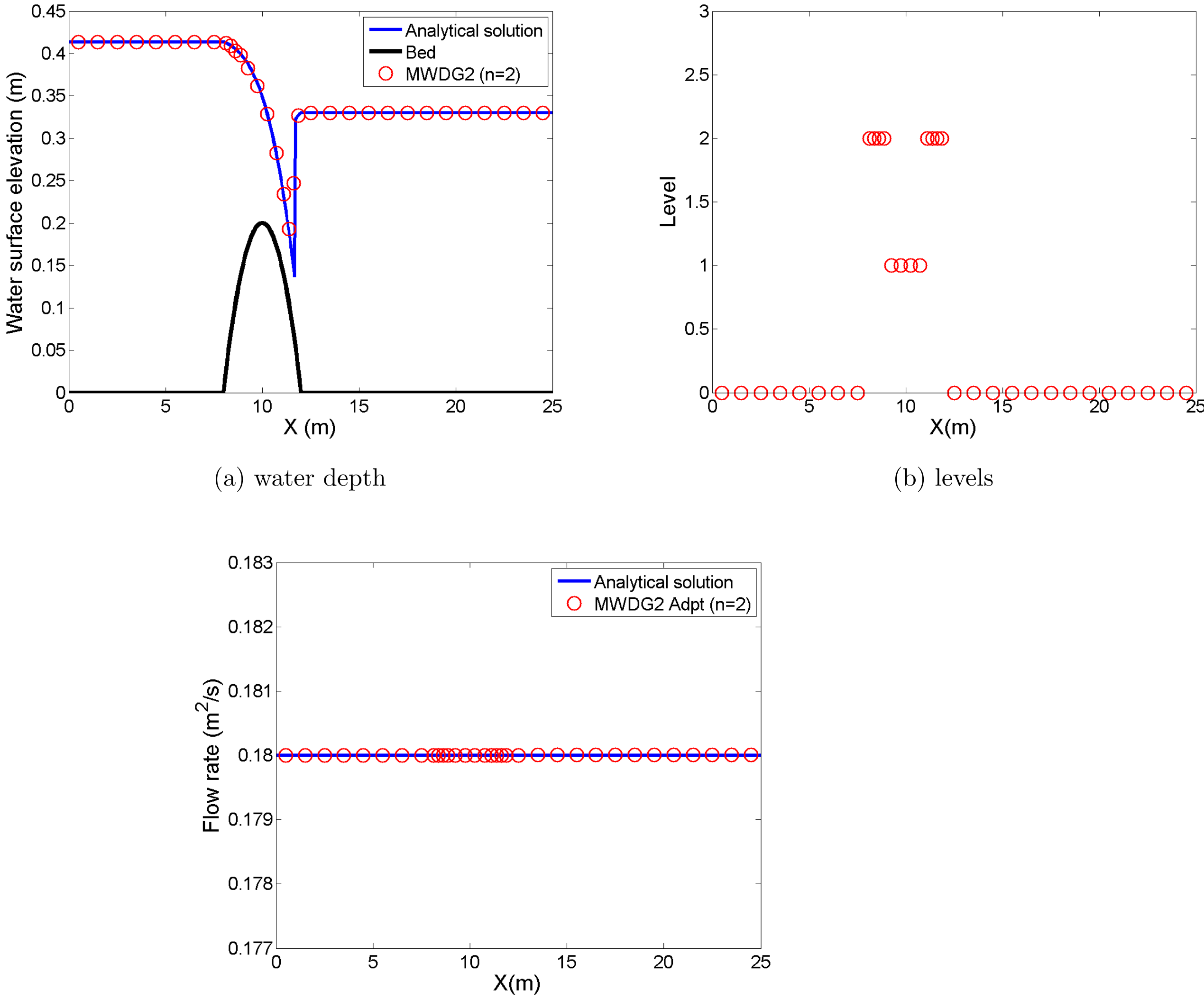
Figure 5.17: HWFV adaptive numerical solution for the steady transcritical flow over a hump.

5.7.2 MWDG2 solution

Since the DG2 formulation provides two stencils (degree of freedoms) in every cell and for the comparison purpose with HWFV simulation, the computational domain at the coarse level is discretised to 25 cells, which results 100 cells in finest resolution. The solution converges to the steady state at $t = 200\,s$. Figure 5.18 presents the numerical results of the adaptive scheme as compared to the analytical solutions. The numerical water surface profile shows a very good agreement with the analytical solution. Furthermore, the flow rate stays constant as it featured by the DG2 scheme, and as previously reported by Kesserwani and Liang (2012b). Most of the domain features require the coarse resolution level except the region where the presence of a hump which the peak of the resolution mesh is noted to track the sub-super critical flow and near the hydraulic jump as required.

5.7.3 Comparisons

The two schemes show generally a good performance in reproducing transcritical water depth over hump. But in terms of discharge, the MWDG2 scheme can transfer and preserve better than HWFV scheme and this is featured as an advantage of the DG2 method over FV method especially in prediction of steady discharge when the mesh is coarse. In both test cases, the maximum resolution is obtained mainly in the region where the hump exists in order to track and capture the varied steady flow. Furthermore, irrespective to the baseline mesh size, the adaptive solutions required almost the same proportion of the associated fine reference meshes. This is due to fixing the initial degrees of freedom for both adaptive schemes. These proportions are 34% and 33% for the HWFV and MWDG2 schemes, respectively which results a reduction in computation efforts compared to associated uniform schemes.



(a) water depth

(b) levels

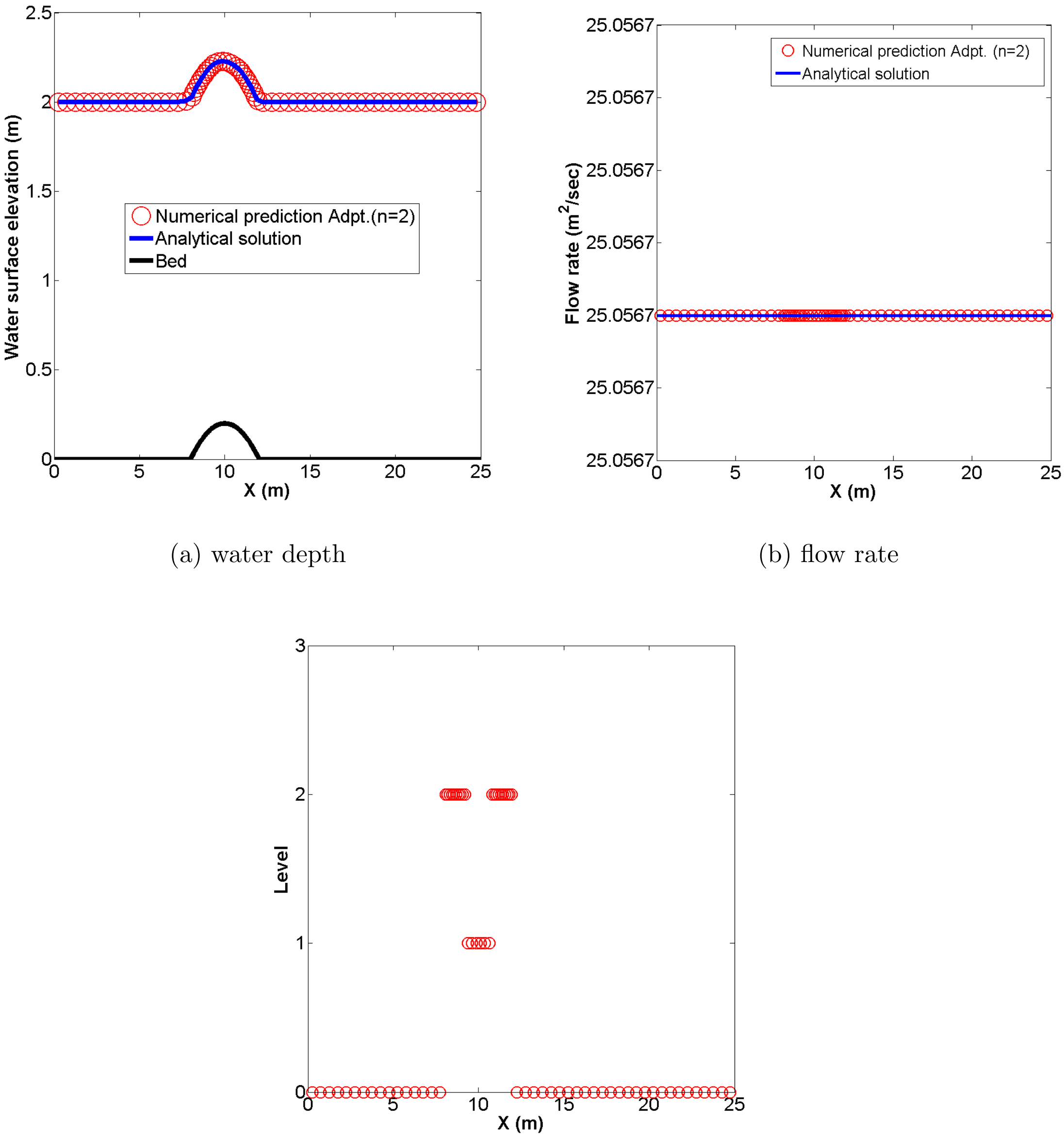
(c) flow rate

Figure 5.18: MWDG2 adaptive numerical solution for the steady transcritical flow over a hump.

5.8 Supercritical flow over a hump

The supercritical flow over hump is considered using only the HWFV scheme to ensure that the disturbances observed in the discharge in the transcritical flow test case (see Figure 5.17c) are not induced by the adaptivity over the local cells. Channel geometry and bed topography are identical to the previous test case, but the unit inflow rate and the elevation water surface at the upstream of the channel are set to $q = 25.0567\, m^3/(s.m)$ and $h + z = 2.0\, m$, respectively. Herein, both of these physical values are used as steady state inflow boundary conditions; whereas a free outlet is numerically set. The simulation time is set to $t = 20\, s$ and 50 cells are chosen for the baseline coarse mesh.

The results are shown in Figure 5.19. The constant flow rate and surface water profile compared with the analytical solution, from SWASHES (2014), are well captured. Again, the mesh refinement is obtained only in the regions where both the bed elevation and the flow are varying. However, no artifact are noted in the prediction of the discharge discreization (see Figure 5.19b). Consequently, the HWFV scheme demonstrates a good performance of the adaptive shallow water flow model in reproducing well-balance steady flows over topography, and in performing a selection of resolution levels relevance with the flow and topographic region.



(a) water depth

(b) flow rate

(c) levels

Figure 5.19: HWFV adaptive numerical solution for the steady supercritical flow over a hump.

5.9 Steady hydraulic jump with friction in a rectangular channel

The purpose of this test case is to illustrate the behavior of both schemes in performing the adaptivity process considering the bed-friction with the source term. This test case is based on the setup proposed by MacDonald (1996). A hydraulic jump is assumed to be formed in a prismatic rectangular channel which has $1000\,m$ long and $20\,m$ width. The Manning's roughness coefficient is set to 0.02 and the bed slope is spatially varied. The initial flow rate and the water depth are set to $20\,m^3/s$ and $1.334899\,m$ in every cell respectively. Boundary conditions at the upstream are set to $q = Q/B = 2\,m^2/s$ and $h = 0.543853\,m$ to obtain a supercritical flow and the flow condition changes via a hydraulic jump to be a subcritical flow at the mid of the channel. The flow remains subcritical until the end of the channel, therefore the boundary condition at the down stream requires only one physical condition to be specified. Here, the water depth is set to $1.334899\,m$ and for the flow rate, it is proceed by a numerical boundary condition.

5.9.1 HWFV solution

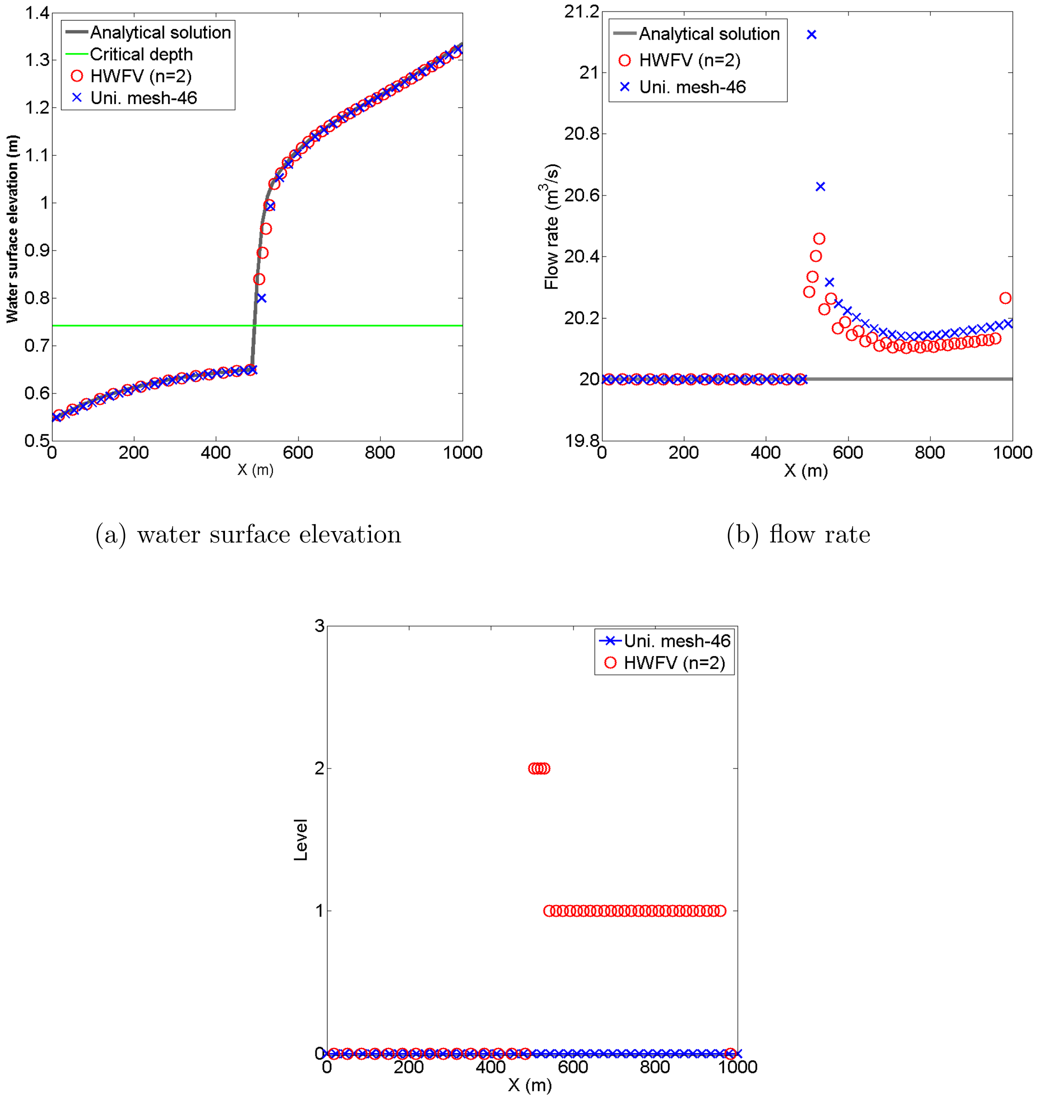
The baseline mesh has 30 cells, which results 120 cells at the highest level (i.e. $n = 2$). The non-adaptive computation is performed using the same number of cells that the adaptive solution is needed to converge the steady hydraulic jump. the convergence of the adaptive solution is achieved with 46 cells. This is performed for the purpose of comparison. The simulations are noted to converge at around $T = 600\,s$. Figure 5.20 shows the results of the adaptive and uniform conditions, together with the exact solution. Generally, the numerical solutions match together with the analytical solution and they overlap almost fully. However, some differences exist, particularly in the flow rate solutions, see Figure 5.20b, and that due to the chosen discretisation of the friction source term, as previously reported by Burguete Tolosa et al. (2008). Clearly, the adaptive solution has more cells for capturing the hydraulic jump compared to the uniform schemes solution and, at the subcritical region, the uniform solution is slightly less accurate, see Figures 5.20a. Based on this, the adaptive scheme can refine more cells through promoting the local solution ($n = 2$ at the hydraulic jump region and $n = 1$ at the subcritical region, see Figure 5.20c), to minimize errors at the same computational cost.

5.9.2 MWDG2 solution

The domain of the computation at $n = 0$ is divided into 15 cells and the adaptive computation is performed with $n = 2$ levels, which results in 60 cells. This discretization of the domain provides 120 degrees of freedom. Thus, the performance of the comparisons between the adaptive results obtained from both schemes (i.e., HWFV and MWDG) become reasonable.

The adaptive solution converges with a maximum of 25 cells and this number of cells is used to run the non-adaptive computation. This is performed to compare its results with the adaptive solution results. The convergence of the solutions are achieved at around $T = 530\,s$ which is less compared with the HWFV scheme and this is due to the advantage of the DG2 over FV schemes. Figure 5.21 presents the results for both the adaptive and uniform conditions, together with the analytical solution. The numerical solutions of both schemes have a good agreement with the analytical solution and they generally overlap throughout the domain with some difference in terms of the distribution of cells in the domain, see Figure 5.21a. At the hydraulic jump and subcritical regions the adaptive scheme has refined more cells in contrast with the supercritical region, in which the adaptive scheme retains coarse level. This implies that the MWDG2 scheme responses to the flow conditions in an optimal way compared to DG2 scheme.

It is clear that the adaptive MWDG2 scheme has transferred successfully the feature of the standard DG2 of preserving the constant discharge. In addition, the adaptive scheme also offers more cells where the hydraulic jump occurs which leads to capturing it better when compared to the uniform numerical solution results. Furthermore, at the smooth region, particularly in the supercritical flow region, the MWDG2 scheme allows for coarser cells which result in a reduction of computational cost when compared to the uniform computation.

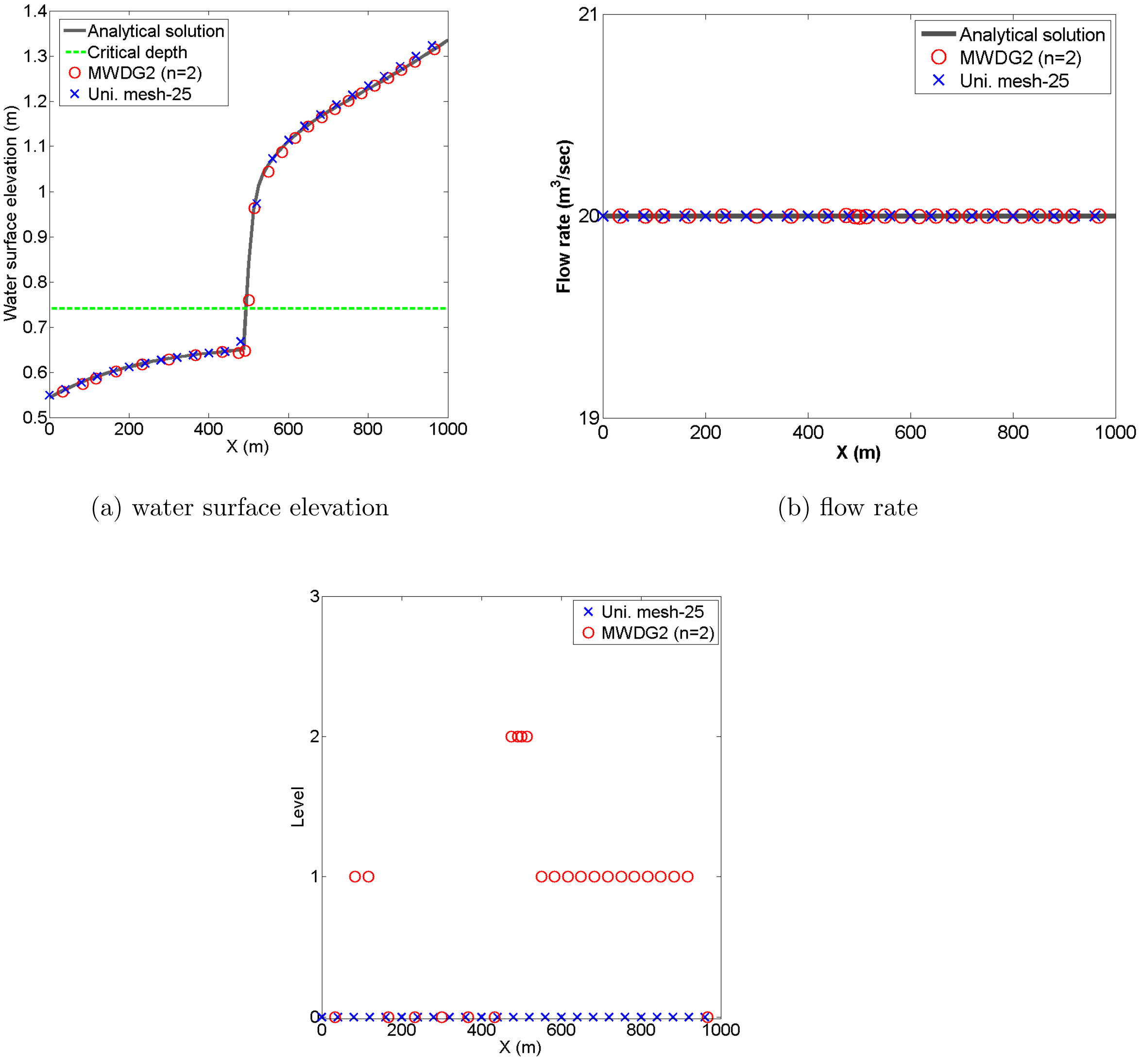


(a) water surface elevation

(b) flow rate

(c) level

Figure 5.20: The results of HWFV scheme for Steady hydraulic jump in a prismatic rectangular channel.



(a) water surface elevation

(b) flow rate

(c) level

Figure 5.21: The results of MWDG2 scheme for Steady hydraulic jump in a prismatic rectangular channel.

5.9.3 Comparisons

In comparing of the two schemes, the MWDG2 scheme is generally better in capturing the steady discharge solution throughout the domain. The both schemes have an advantage over the uniform scheme (i.e., FV and DG2) in terms of responding to the flow condition feature, which they refine more cells where the hydraulic jump occurs. Irrespective of the baseline mesh size, the HWFV and the MWDG2 schemes require around 38 % and 42 % respectively, less in computational efforts, than the simulation on the associated fine uniform mesh. These percentages are reasonable because their degrees of freedom are set to be same at $t = 0\,s$. However, these percentages change with time. In Figure 5.22, which shows the evolution of the number of active cells and their variance with N_2. At around $t = 80\,s$, too many cells are being activated by the MWDG2 scheme which is around 75 % of the equivalent uniform scheme. While in the HWFV scheme, the adaptive solution require a maximum of 51 % of the counterpart uniform mesh. The magnitude of N/N_n for both adaptive schemes is relatively the same when $200\,s \leq t \leq 300\,s$ and it becomes almost constant with $t \geq 400\,s$.

It is clear that an efficient adaptive processes are obtained with the selected baseline meshes in both adaptive schemes and also the active cell profiles of both schemes have almost the same response to the flow conditions regardless how many cells are being activated during the simulation. Furthermore, the possibility of coarsening baseline mesh for both adaptive schemes is feasible, since both adaptive schemes are not activating all accessible cells during the simulation.

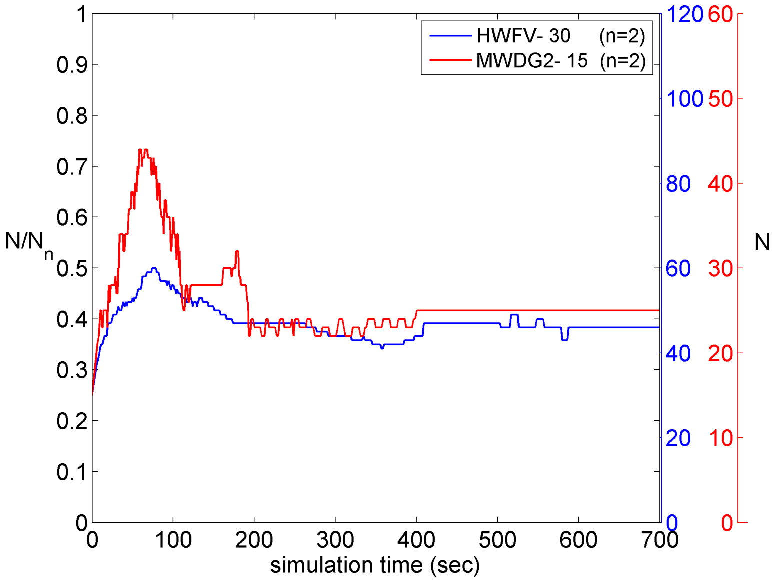


Figure 5.22: The active cells evolution of the adaptive schemes for the steady hydraulic jump with friction bed in the rectangular channel.

Chapter 6

Conclusions and recommendations

6.1 Conclusions

In this work, by combining wavelet-based scaling functions with the design of both standard shallow water flow modelling schemes (i.e. FV and DG), two new adaptive schemes (i.e. HWFV and MWDG2) have been obtained. In these new schemes, the spatial resolution adaptivity is achieved from the local solution itself and it is based on a single user-prescribed parameter. From the analysis of the performance of the test cases in Chapter 5, the following conclusions can be drawn:

1. The precision digits of the filter coefficients were noted to affect the conservation of adaptivity in terms of truncation error of precision digits (Section 5.3). Their effect became greater as more filter coefficients were involved in the computations (i.e. increasing the order of accuracy). It can be, therefore, concluded that the filter coefficients, regardless of the accuracy of the adaptive solution, must be set with a minimum of double precision digits to reduce the loss of conservation due to the truncation error. Based on the behaviour of the identified errors, it is important to consider the trade-off between the baseline mesh resolution and the accuracy of the adaptive solution. This means when the adaptive scheme seeks for high order-accurate solution, the baseline mesh should be coarsened as much as possible in an attempt to reduce the calls of filter coefficients in computations which results

to preserve the precision digits of the conserved variables.

2. The selected threshold value (i.e. $\xi = 0.01$ obtained from Section 5.2 and used for both adaptive schemes) has clearly shown its capability to control a solution-driven mesh adaptation process, in which the quality of the numerical solution is conserved and the adaptive solution errors do not exceed the discretisation-errors of the counterpart uniform schemes. Furthermore, by using this threshold value, the adaptive schemes offered a significant reduction in the computational efforts compared to the associated uniform schemes on the highest resolution level. For instance, the reduction of the HWFV scheme varied between 62 % to 78 % while for the MWDG2 scheme, the reduction varied between 40 % to 79 %.

3. The quantitative analysis considering three levels of resolutions showed that the adaptive schemes can conserve sharp shock wave which is known as one of the main sources of error. This was demonstrated by the maximum error (infinity norm) in which these errors were bounded by the uniform meshes' counterparts at different resolutions (Section 5.4) and the trend errors of both the adaptive and counterpart uniform schemes at resolution n have shown the asymptotic pattern. In addition, the magnitude of mean and standard deviation values for the adaptive error at resolution n are somewhere between the values of uniform counterpart schemes at the resolution level n and $n-1$. These findings signal that for shock-free shallow water flow simulations, the adaptive schemes predictions trend to asymptotically converge to the quality of the highest resolution available.

4. The adaptive schemes was able to retain all properties from the reference uniform counterpart schemes such as the mass conservation, shock capturing, well-balancing, moving wet/dry fronts and slope limiter properties (Sections 5.5, 5.7 and 5.8). This means if any property is valid or improved for the reference uniform schemes, it will be transferable into the adaptive counterparts schemes.

5. Response of the accuracy to the adaptivity process via fixing the degrees of freedom (DOF) in both adaptive schemes (when $t = 0\,s$) was shown

a good performance in terms of providing almost the same computational efforts when the adaptive solutions converged (Section 5.9). In addition, the adaptive schemes delivered better-resolved capture of the flow conditions than the counterpart uniform schemes. In particular, the MWDG2 scheme offered coarser-mesh predictions at supercritical region compared to the HWFV results and counterpart uniform results. This means at the same level of DOF, the adaptive MWDG2 scheme offers better accuracy, conservativity and coarse (efficiency) performance than the HWFV scheme. This clearly lays out to benefit in further from the mutiwavelets scalability for increasing the accuracy of the adaptive solution along with coarsening the baseline meshes.

Altogether, the results in Chapter 5 demonstrated that the multiwavelets adaptive technique has successfully merged into the FV and DG2 frameworks. The results are also encouraging and suggesting that the adaptive schemes can provide a rigorous, robust and efficient solution to the multiscale problems arise in shallow water modelling.

6.2 Recommendations for future work

The following future work will consist of considering the different aspects of hydraulic engineering problems to bring these adaptive schemes into the mainstream as the approaches of demanding shallow water flow simulations:

1. Conducting more test cases to show the capability of these schemes in reproducing experimental reference data.

2. Increasing the accuracy of the adaptive solution. This will encourage to perform *hp*-adaptation.

3. Increasing the refinement levels of both schemes in such a way that the local solution can promote to any resolution level. This will offer a significant gain in reducing the computational efforts.

4. Investigating the trade-off between coarsening the baseline mesh and increasing the number of resolution levels. This will give insight as the best choice

of resolution level that can be used with simulating the realistic applications of shallow water flow.

5. Extending these schemes to simulate 2D shallow flows considering all quantitative and qualitative assessments that are performed for 1D to ensure the possibility of using these adaptive schemes in modelling the real shallow water flow problems.

.

Bibliography

Abgrall, R. (1998). Multiresolution representation in unstructured meshes. *Siam Journal on numerical analysis*, 35(6):2128–2146.

Alcrudo, F. and Garcia-Navarro, P. (1993). A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations. *International Journal for Numerical Methods in Fluids*, 16(6):489–505.

Alpert, B., Beylkin, G., Gines, D., and Vozovoi, L. (2002). Adaptive solution of partial differential equations in multiwavelet bases. *Journal of Computational Physics*, 182(1):149–190.

Alpert, B. K. (1993). A class of bases in L^2 for the sparse representation of integral operators. *SIAM journal on Mathematical Analysis*, 24(1):246–262.

Alpert, B. K., Mohamed, J., Methods, C., and Equations, I. (1992). Wavelet and other bases for fast numerical linear algebra.

Audusse, E., Bouchut, F., Bristeau, M. O., Klein, R., and Perthame, B. (2004). A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *Siam Journal on Scientific Computing*, 25(6):2050–2065.

Aureli, F., Maranzoni, A., Mignosa, P., and Ziveri, C. (2008). A weighted surface-depth gradient method for the numerical integration of the 2D shallow water equations with topography. *Advances in Water Resources*, 31(7):962 – 974.

Bader, M., Böck, C., Schwaiger, J., and Vigh, C. (2010). Dynamically adaptive simulations with minimal memory requirement-solving the shallow water equations using sierpinski curves. *Siam Journal on Scientific Computing*, 32(1):212–228.

Bermudez, A. and Vazquez, M. E. (1994a). Upwind methods for hyperbolic conservation laws with source terms. *Computers and Fluids*, 23(8):1049–1071.

Bermudez, A. and Vazquez, M. E. (1994b). Upwind methods for hyperbolic conservation laws with source terms. *Computers and Fluids*, 23(8):1049 – 1071.

Bihari, B. L. and Harten, A. (1995). Application of generalized wavelets: an adaptive multiresolution scheme. *Journal of computational and applied mathematics*, 61(3):275–321.

Bokhove, O. (2005). Flooding and drying in discontinuous Galerkin finite-element discretizations of shallow-water equations. part 1: one dimension. *Journal of scientific computing*, 22(1-3):47–82.

Brufau, P., García-Navarro, P., and Vázquez-Cendón, M. (2004). Zero mass error using unsteady wetting–drying conditions in shallow flows over dry irregular topography. *International Journal for Numerical Methods in Fluids*, 45(10):1047–1082.

Brufau, P., Vazquez-Cendpon, M., and Garcia-Navarro, P. (2002). A numerical model for the flooding and drying of irregular domains.

Bunya, S., Kubatko, E. J., Westerink, J. J., and Dawson, C. (2009). 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):1548–1562.

Burguete Tolosa, J., García-Navarro, P., Murillo, J., et al. (2008). Friction term discretization and limitation to preserve stability and conservation in the 1d shallow-water model: Application to unsteady irrigation and river flow. *International Journal for Numerical Methods in Fluids*, 58(4):403–425.

Caleffi, V. and Valiani, A. (2009). Well-balanced bottom discontinuities treatment for high-order shallow water equations WENO scheme. *Journal of Engineering Mechanics*, 135(7):684–696.

Caleffi, V. and Valiani, A. (2012). A well-balanced, third-order-accurate RKDG scheme for SWE on curved boundary domains. *Advances in Water Resources*, 46:31 – 45.

Caleffi, V., Valiani, A., and Bernini, A. (2006). Fourth-order balanced source term treatment in central WENO schemes for shallow water equations. *Journal of Computational Physics*, 218(1):228 – 245.

Caviedes-Voullième, D., García-Navarro, P., and Murillo, J. (2012). Influence of mesh structure on 2D full shallow water equations and SCS curve number simulation of rainfall/runoff events. *Journal of Hydrology*, 448-449(0):39–59.

Chavent, G. and Cockburn, B. (1987). The local projection P0P1-discontinuous-Galerkin-finite element method for scalar conservation laws.

Chavent, G. and Salzano, G. (1982). A finite-element method for the 1-D water flooding problem with gravity. *Journal of Computational Physics*, 45(3):307–344.

Cockburn, B. (1998). An introduction to the discontinuous Galerkin method for convection-dominated problems. In *Advanced numerical approximation of nonlinear hyperbolic equations*, pages 151–268. Springer.

Cockburn, B. (1999). Discontinuous Galerkin methods for convection-dominated problems. In *High-order methods for computational physics*, pages 69–224. Springer.

Cockburn, B. and Shu, C.-W. (1989). TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. ii. general framework. *Mathematics of computation*, 52(186):411–435.

Cockburn, B. and Shu, C.-W. (1998). The local P0P1 discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463.

Cockburn, B. and Shu, C.-W. (2001). Runge Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of scientific computing*, 16(3):173–261.

Cockburn, Bernardo; Shu, C.-W. (1987). The Runge-Kutta local projection P1-discontinuous-galerkin finite element method for scalar conservation laws. Retrieved from the University of Minnesota Digital Conservancy.

Cohen, A., Kaber, S., Müller, S., and Postel, M. (2003). Fully adaptive multiresolution finite volume schemes for conservation laws. *Mathematics of Computation*, 72(241):183–225.

Cunge, J. A., Holly, F. M., and Verwey, A. (1980). Practical aspects of computational river hydraulics.

Dahmen, W., Gottschlich-Müller, B., and Müller, S. (2001). Multiresolution schemes for conservation laws. *Numerische Mathematik*, 88(3):399–443.

Daubechies, I. et al. (1988). Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 41(7):909–996.

Delis, A. and Kampanis, N. (2009). Numerical flood simulation by depth averaged free surface flow models. *Environmental Systems–Encyclopedia of Life Support Systems (EOLSS)*.

Farge, M. and Schneider, K. (2001). Coherent vortex simulation (CVS), a semi-deterministic turbulence model using wavelets. *Flow, Turbulence and Combustion*, 66(4):393–426.

Garcia-Navarro, P. and Vazquez-Cendon, M. E. (2000). On numerical treatment of the source terms in the shallow water equations. *Computers and Fluids*, 29(8):951–979.

Gargour, C., Gabrea, M., Ramachandran, V., and Lina, J.-M. (2009). A short introduction to wavelets and their applications. *IEEE circuits and systems magazine*, 9(2):57–68.

Gerhard, N., Iacono, F., May, G., Mller, S., and Schfer, R. (2015). A high-order discontinuous Galerkin discretization with multiwavelet-based grid adaptation for compressible flows. *Journal of Scientific Computing*, 62(1):25–52.

Glaister, P. (1988). Approximate riemann solutions of the shallow water equations. *Journal of Hydraulic Research*, 26(3):293–306.

Goldberg, D. (1991). What every computer scientist should know about floating point arithmetic. *ACM Computing Surveys*, 23(1):5–48.

Grossmann, A. and Morlet, J. (1984). Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM journal on mathematical analysis*, 15(4):723–736.

Guinot, V. (2003). *Godunov-type schemes: an introduction for engineers*. Elsevier.

Haleem, D. A., Kesserwani, G., and Caviedes-Voullime, D. (2015). Haar wavelet-based adaptive finite volume shallow water solver. *Journal of Hydroinformatics In Press, IWA Publishing*.

Harten, A. (1995). Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Communications on Pure and Applied Mathematics*, 48(12):1305–1342.

Henderson, F. (1966). Open channel flow macmillan series in civil engineering.

Hirsch, C. (2002). *Numerical Computation Of Internal And External Flows: Volume 2: Computational Methods For Inviscid And Viscous Flows*. Chichester [etc.]: John Wiley & Sons.

Hou, J., Simons, F., Mahgoub, M., and Hinkelmann, R. (2013). A robust well-balanced model on unstructured grids for shallow water flows with wetting and drying over complex topography. *Computer Methods in Applied Mechanics and Engineering*, 257:126–149.

Hovhannisyan, N., Müller, S., and Schäfer, R. (2014). Adaptive multiresolution discontinuous Galerkin schemes for conservation laws. *Mathematics of computation*, 83(285):113–151.

Johnson, C. and Pitkäranta, J. (1986). An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Mathematics of computation*, 46(173):1–26.

Keinert, F. (2003). *Wavelets and Multiwavelets*. CRC Press.

Kesserwani, G. (2013). Topography discretization techniques for Godunov-type shallow water numerical models: a comparative study. *Journal of Hydraulic Research*, 51(4):351–367.

Kesserwani, G., Caviedes-Voulliía, D., Gerhard, N., and Muller, S. (2015). Multiwavelet discontinuous Galerkin -adaptive shallow water model. *Computer Methods in Applied Mechanics and Engineering*, 294:56 – 71.

Kesserwani, G., Gerhard, N., Caviedes-Voullime, D., Haleem, D. A., and Mller, S. (2014). A multi-resolution discontinuous Galerkin method for one dimensional shallow water flow modelling. the 3rd IAHR Europe Congress with the theme Water Engineering and Research. Porto, Portugal.

Kesserwani, G. and Liang, Q. (2010a). A discontinuous Galerkin algorithm for the two-dimensional shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 199(4952):3356 – 3368.

Kesserwani, G. and Liang, Q. (2010b). Well-balanced RKDG2 solutions to the shallow water equations over irregular domains with wetting and drying. *Computers and Fluids*, 39(10):2040 – 2050.

Kesserwani, G. and Liang, Q. (2011). A conservative high-order discontinuous Galerkin method for the shallow water equations with arbitrary topography. *International journal for numerical methods in engineering*, 86(1):47–69.

Kesserwani, G. and Liang, Q. (2012a). Dynamically adaptive grid based discontinuous galerkin shallow water model. *Advances in Water Resources*, 37:23–39.

Kesserwani, G. and Liang, Q. (2012b). Locally limited and fully conserved RKDG2 shallow water solutions with wetting and drying. *Journal of scientific computing*, 50(1):120–144.

Kesserwani, G. and Wang, Y. (2014). Discontinuous Galerkin flood model formulation: Luxury or necessity? *Water Resources Research*, 50(8):6522–6541.

Khan, A. A. and Lai, W. (2014). *Modeling Shallow Water Flows Using the Discontinuous Galerkin Method*. CRC Press.

Krivodonova, L., Xin, J., Remacle, J., Chevaugeon, N., and Flaherty, J. E. (2004). Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. *Applied Numerical Mathematics*, 48(3):323–338.

Kubatko, E. J., Bunya, S., Dawson, C., and Westerink, J. J. (2009). Dynamic p-adaptive Runge–Kutta discontinuous Galerkin methods for the shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 198(21):1766–1774.

Kubatko, E. J., Westerink, J. J., and Dawson, C. (2006). hp discontinuous Galerkin methods for advection dominated problems in shallow water flow. *Computer Methods in Applied Mechanics and Engineering*, 196(13):437 – 451.

Lesaint, P. and Raviart, P.-A. (1974). On a finite element method for solving the neutron transport equation. *Mathematical aspects of finite elements in partial differential equations*, (33):89–123.

Liang, Q. (2010). Flood simulation using a well-balanced shallow flow model. *Journal of Hydraulic Engineering*, 136(9):669–675.

Liang, Q. and Borthwick, A. G. (2009). Adaptive quadtree simulation of shallow flows with wet dry fronts over complex topography. *Computers and Fluids*, 38(2):221 – 234.

Liu, X.-D., Osher, S., and Chan, T. (1994). Weighted Essentially Non-oscillatory Schemes. *Journal of Computational Physics*, 115(1):200 – 212.

MacDonald, I. (1996). *Analysis and computation of steady open channel flow. PhD. Thesis*. University of Reading.

Mallat, S. G. (1989). Multiresolution approximations and wavelet orthonormal bases of L^2(R). *Transactions of the American Mathematical Society*, 315(1):69–87.

Mehra, M. and Kevlahan, N. K.-R. (2008). An adaptive wavelet collocation method for the solution of partial differential equations on the sphere. *Journal of Computational Physics*, 227(11):5610–5632.

Moler, C. B. (2008). *Numerical Computing with MATLAB: Revised Reprint*. Siam.

Müller, S. (2003). *Adaptive multiscale schemes for conservation laws*, volume 27. Springer.

Müller, S. and Stiriba, Y. (2007). Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping. *Journal of scientific computing*, 30(3):493–531.

Murillo, J. and Garca-Navarro, P. (2010). Weak solutions for partial differential equations with source terms: Application to the shallow water equations. *Journal of Computational Physics*, 229(11):4327 – 4368.

Murillo, J., García-Navarro, P., Burguete, J., and Brufau, P. (2007). The influence of source terms on stability, accuracy and conservation in two-dimensional shallow flow simulation using triangular finite volumes. *International journal for numerical methods in fluids*, 54(5):543–590.

Nemec, M. and Aftosmis, M. J. (2007). Adjoint error estimation and adaptive refinement for embedded-boundary cartesian meshes. *AIAA Paper*, 4187:2007.

Nikolos, I. and Delis, A. (2009). An unstructured node-centered finite volume scheme for shallow water flows with wet/dry fronts over complex topography. *Computer Methods in Applied Mechanics and Engineering*, 198(47):3723–3750.

Osher, S. (1984). Riemann solvers, the entropy condition, and difference. *SIAM Journal on Numerical Analysis*, 21(2):217–235.

Peterson, T. E. (1991). A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM Journal on Numerical Analysis*, 28(1):133–140.

Qiu, J. and Shu, C.-W. (2005). A comparison of troubled-cell indicators for Runge–Kutta discontinuous galerkin methods using weighted essentially nonoscillatory limiters. *SIAM Journal on Scientific Computing*, 27(3):995–1013.

Quirk, J. J. (1994). *Godunov-type schemes applied to detonation flows*. Springer.

Reed, W. and Hill, T. (1973). *Triangular mesh methods for the neutron transport equation.*

Remacle, J.-F., Frazao, S. S., Li, X., and Shephard, M. S. (2006). An adaptive discretization of shallow-water equations based on discontinuous Galerkin methods. *International journal for numerical methods in fluids*, 52(8):903–923.

Roe, P. L. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372.

Roe, P. L. and Pike, J. (1985). Efficient construction and utilisation of approximate riemann solutions. In *Proc. Of the Sixth Int'L. Symposium on Computing Methods in Applied Sciences and Engineering, VI*, pages 499–518, Amsterdam, The Netherlands, The Netherlands. North-Holland Publishing Co.

Rogers, B. D., Borthwick, A. G., and Taylor, P. H. (2003). Mathematical balancing of flux gradient and source terms prior to using roe approximate riemann solver. *Journal of Computational Physics*, 192(2):422 – 451.

Schneider, K., Kevlahan, N.-R., and Farge, M. (1997). Comparison of an adaptive wavelet method and nonlinearly filtered pseudospectral methods for two-dimensional turbulence. *Theoretical and computational fluid dynamics*, 9(3-4):191–206.

Schneider, K. and Vasilyev, O. V. (2010). Wavelet methods in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 42:473–503.

Schwanenberg, D. and Harms, M. (2004). Discontinuous Galerkin finite-element method for transcritical two-dimensional shallow water flows. *Journal of Hydraulic Engineering*, 130(5):412–421.

Schwanenberg, D. and Köngeter, J. (2000). A discontinuous Galerkin method for the shallow water equations with source terms. In *Discontinuous Galerkin Methods*, pages 419–424. Springer.

Shelton, A. B. (2009). *A multi-resolution discontinuous Galerkin method for unsteady compressible flows.* PhD thesis, Georgia Institute of Technology.

Shu, C.-W. (1987). TVB uniformly high-order schemes for conservation laws. *Mathematics of Computation*, 49(179):105–121.

Skoula, Z., Borthwick, A., and Moutzouris, C. (2006). Godunov-type solution of the shallow water equations on adaptive unstructured triangular grids. *International Journal of Computational Fluid Dynamics*, 20(9):621–636.

Soman, K. et al. (2010). *Insight into wavelets: From theory to practice.* PHI Learning Pvt. Ltd.

SWASHES (2014). Shallow Water Analytic Solutions for Hydraulic and Environmental Studies. http://www.univ-orleans.fr/mapmo/soft/SWASHES/. [Online; accessed 19-July-2014].

Tan, W.-Y. (1992). *Shallow water hydrodynamics: Mathematical theory and numerical solution for a two-dimensional system of shallow-water equations.* Elsevier.

Tassi, P. A. and Vionnet, Carlos A, B. (2003). Discontinuous Galerkin method for the one dimensional simulation of shallow water flows. *Mecánica Computacional*, 22:1474–1488.

Te Chow, V. (1959). Open channel hydraulics. *McGraw-Hill Book Company, Inc; New York.*

Thacker, W. C. (1981). Some exact solutions to the nonlinear shallow-water wave equations. *Journal of Fluid Mechanics*, 107:499–508.

Toro, E. F. (2001). *Shock-capturing methods for free-surface shallow flows.* Wiley.

Toro, E. F. (2009). *Riemann solvers and numerical methods for fluid dynamics: a practical introduction.* Springer Science & Business Media.

Toro, E. F. (2012). *Godunov methods: theory and applications.* Springer Science & Business Media.

Toro, E. F. and Garcia-Navarro, P. (2007). Godunov-type methods for free-surface shallow flows: A review. *Journal of Hydraulic Research*, 45(6):736–751.

Vázquez-Cendón, M. E. (1999). Improved treatment of source terms in upwind schemes for the shallow water equations in channels with irregular geometry. *Journal of Computational Physics*, 148(2):497–526.

Xing, Y., Zhang, X., and Shu, C.-W. (2010). Positivity-preserving high order well-balanced discontinuous Galerkin methods for the shallow water equations. *Advances in Water Resources*, 33(12):1476 – 1493.

Yoshimatsu, K., Okamoto, N., Kawahara, Y., Schneider, K., and Farge, M. (2013). Coherent vorticity and current density simulation of three-dimensional magnetohydrodynamic turbulence using orthogonal wavelets. *Geophysical & Astrophysical Fluid Dynamics*, 107(1-2):73–92.

Zhou, T., Li, Y., and Shu, C.-W. (2001). Numerical comparison of WENO finite volume and runge kutta discontinuous galerkin methods. *Journal of Scientific Computing*, 16(2):145–171.

Appendix A

One-dimensional Gauss integration

Gauss integration is explained for the one-dimensional case without resorting to the concept of Legendre polynomials. The trade-off is that symmetry of the integration point positions must be assumed. In what follows, an integration domain of the interval $x \in [-\frac{1}{2}L, \frac{1}{2}L]$ is assumed.

linear function: A constant function is written as $y = a + bx$ and its integral can be elaborated as aL. For a one-point Gauss integration the weight is necessarily unity, and the position x_1 of the integration point is found via

$$(a + bx_1)\,L = aL \tag{A.1}$$

which leads to $x_1 = 0$, i.e. the integration point is positioned in the centre of the interval.

quadratic function: A quadratic function is written as $y = a + bx + cx^2$ and its integral equals $aL + \frac{1}{12}cL^3$. A single integration point would be placed in the centre, and is not sufficient to account for arbitrary a and c. Therefore, two integration points are needed. It should then hold that

$$\sum_i y_i w_i L = aL + \frac{1}{12}cL^3 \qquad i = 1, 2 \tag{A.2}$$

Comparing the coefficients of a, b and c separately yields

$$aL\,(w_1 + w_2) = aL \tag{A.3a}$$

$$bL\left(x_1w_2 + x_2w_2\right) = 0 \tag{A.3b}$$

$$cL\left(x_1^2w_1 + x_2^2w_2\right) = \frac{1}{12}cL^3 \tag{A.3c}$$

These are three equations with four unknowns. Additional equations are generated by requiring that the integration rule is objective with respect to the orientation of the x-axis, i.e. $x_1 = -x_2$ and $w_1 = w_2$. Equation (A.3a) then yields $w_1 = w_2 = \frac{1}{2}$ and Equation (A.3b) is satisfied automatically. Equation (A.3c) results in $x_1 = -\frac{1}{6}L\sqrt{3}$ and $x_2 = \frac{1}{6}L\sqrt{3}$.

cubic function: A cubic function is written as $y = a + bx + cx^2 + dx^3$ and its integral is $aL + \frac{1}{12}cL^3$. The form of integral is identical to that of a quadratic function, therefore the same integration rule can be applied.

quartic function: A quartic function is written as $y = a + bx + cx^2 + dx^3 + ex^4$ and its integral equals $aL + \frac{1}{12}cL^3 + \frac{1}{80}eL^5$. Two integration points are not sufficient for arbitrary a, c and e, therefore three integration points are needed. It is required that

$$\sum_i y_iw_iL = aL + \frac{1}{12}cL^3 + \frac{1}{80}eL^5 \qquad i = 1,2,3 \tag{A.4}$$

Comparing the coefficients of a, b, c, d and e separately yields

$$aL\left(w_1 + w_2 + w_3\right) = aL \tag{A.5a}$$

$$bL\left(x_1w_1 + x_2w_2 + x_3w_3\right) = 0 \tag{A.5b}$$

$$cL\left(x_1^2w_1 + x_2^2w_2 + x_3^2w_3\right) = \frac{1}{12}cL^3 \tag{A.5c}$$

$$dL\left(x_1^3w_1 + x_2^3w_2 + x_3^3w_3\right) = 0 \tag{A.5d}$$

$$eL\left(x_1^4w_1 + x_2^4w_2 + x_3^4w_3\right) = \frac{1}{80}eL^5 \tag{A.5e}$$

These are six equations with five unknowns. Additionally, symmetry of the integration point locations is required, that is $x_1 = -x_3$ and $x_2 = 0$ and for the weights it holds $w_1 = w_3$. Equations (A.5b) and (A.5d) are then fulfilled automatically.

Substituting these values into Equations (A.5c) and (A.5e) leads to

$$2x_1^2w_1 = \frac{1}{12}L^2 \qquad \Rightarrow x_1^4 = \frac{1}{576w_1^2}L^4 \tag{A.6a}$$

$$2x_1^4 w_1 = \frac{1}{80}L^4 \qquad \Rightarrow x_1^4 = \frac{1}{160 w_1}L^4 \tag{A.6b}$$

which yields $w_1 = \frac{5}{18}$. Equation (A.5a) then results in $w_2 = \frac{4}{9}$. Finally, it is found that $x_1 = -\frac{1}{10}L\sqrt{15}$ and $x_3 = \frac{1}{10}L\sqrt{15}$.

quintic function: A quintic function is written as $y = a+bx+cx^2+dx^3+ex^4+fx^5$ and its integral equals $aL + \frac{1}{12}cL^3 + \frac{1}{80}eL^5$. Therefore the same integration rule applies as for the quartic function.

Appendix B

The algorithm of the dry bed treatment HWFV

Algorithm 5 Dry bed treatment based on the work of Liang (2010)

```
Construct the water surface elevation ($\eta_i = h_i + z_i$) and ($\eta_{i+1} = h_{i+1} + z_{i+1}$).
Record the velocities at cells $I_i$ and $I_{i+1}$, i.e.,$u_i = q_i/h_i\,; u_{i+1} = q_{i+1}/h_{i+1}$. (Audusse et al., 2004)
Impose the continuity of the topography at interfaces, such that $z^*_{i+1/2} = max\,(z_i, z_{i+1})$. This step facilitates the wetting and drying front, $\Delta\eta_{i+1/2} = max\big[0, -(\eta_i - z^*_{i+1/2})\big]$.
Reconstruct depths involved in the hydrostatic term $h^*_i = \eta_i - z^*_{i+1/2}$ and $h^*_{i+1} = \eta_{i+1} - z^*_{i+1/2}$.
Compute the discharge terms, $q^*_i = h^*_i\,u_i\,; q^*_{i+1} = h^*_{i+1}\,u_{i+1}$.
The bed topography discretization at interfaces with wet/dry front needs further steps
```

```
for each dry cell do
    a. Construct the new surface gradients $\eta_i^* = h_i^* + z_{i+1/2}^*.$ ; $\eta_{i+1}^* = h_{i+1}^* + z_{i+1/2}^*.$
    b. Ensure that the actual water level of wet/dry interface is preserved i.e, when $\Delta\eta_{i+1/2} > 0$ an extra modification is performed to revise variables; $\eta_{i,i+1}^* \longleftarrow \eta_{i,i+1}^* - \Delta\eta_{i+1/2}$ ; $z_{i+1/2}^* \longleftarrow z_{i+1/2}^* - \Delta\eta_{i+1/2}.$
    c. Bring back the water depth $h_{i,i+1}^* = \eta_{i,i+1}^* - z_{i+1/2}^*$ and then use $\mathbf{U}_{i+1/2}^L = [h_i^*, q_i^*]^T$ and $\mathbf{U}_{i+1/2}^R = [h_{i+1}^*, q_{i+1}^*]^T$ in equation 4.18 to evaluate the flux.
end for

7: Likewise for evaluating flux $\tilde{\mathbf{F}}_{i-1/2}$ at interface $x_{i-1/2}$, repeat the steps (1) to (6).
8: Set $\tilde{h} = (h_{i+1/2}^{L,*} + h_{i-1/2}^{R,*})/2$ and $\partial_x \tilde{z} = (z_{i+1/2}^{L,*} + z_{i-1/2}^{R,*})/\Delta x$ and use them to evaluate $\mathbf{S}_i^n$ in expression 4.25.
```

Appendix C

The algorithm of the dry bed treatment MWDG

Algorithm 6 Dry bed treatment based on Kesserwani and Liang (2012b) work

```
1: Construct the water surface elevation ($\eta = h + z$) at cell boundary $x_{i+1/2}$.
2: Record velocities at $x_{i+1/2}$ obtained from the original topography
discretization, i.e. $z^L_{i+1/2} = z_i = z^R_{i+1/2} = z_{i+1}$.
3: Perform $h^K_{i+1/2} = \eta^K_{i+1/2} - z^K_{i+1/2}$ where $^{K=R\,or\,L.}$
4: The bed topography discretization at interface $x_{i+1/2}$ has wet/dry front will
be
    for each dry cell do
        a. Re-define numerically $z^{K,*}_{i+1/2} = \eta^K_{i+1/2} - h^K_{i+1/2}$.
        b. Check if ($h^K_{i+1/2} < \epsilon_{dry}$) then $u^K_{i+1/2} = 0$ end if
        c. Set a single $z$-value defined by the maximum value,i.e.
        $z^{R,L,*}_{i+1/2} = max(z^{L,*}_{i+1/2}, z^{R,*}_{i+1/2})$. Only to ensure nothing
        omitted in step (a.).
        d. Preserve positivity of water depth, i.e. $h^{K,*}_{i+1/2} =$
        $max(0, \eta^K_{i+1/2} - z^{L,R,*}_{i+1/2})$.
        e. Compute the discharge incorporating from the original velocities,i.e.
        $q^{K,*}_{i+1/2} = h^{K,*}_{i+1/2} u^K_{i+1/2}$ and the free-surface elevation, $\eta^{K,*}_{i+1/2}$
        $= h^{K,*}_{i+1/2} + z^{L,R,*}_{i+1/2}$, correlated with the positivity-preserving water depth
        and single value of topography bed.
        f. Ensure that the step (4) is preserving the actual water level at
        wet/dry front, this achieves by imposing the extra steps of Liang (2010):
        i. Find $\Delta\eta_{i+1/2} = max\Big[(0, -(\eta^{k}_{i+1/2} - z^{L,R,*}_{i+1/2})\Big]$.
        ii. Adjust $\eta^{K,*}_{i+1/2} \longleftarrow \eta^{K,*}_{i+1/2} - \Delta\eta_{i+1/2}$ and $z^{L,R,*}_{i+1/2} \longleftarrow z^{L,R,*}_{i+1/2} - \Delta\eta_{i+1/2}$.
    end for
    5: Find the flux $\tilde{\mathbf{F}}_{i+1/2}$ using $h^{K,*}_{i+1/2} = \eta^{K,*}_{i+1/2} - z^{L,R,*}_{i+1/2}$ and $u^{K}_{i+1/2} = q^{K,*}_{i+1/2}/h^{K,*}_{i+1/2}$.
    6: Likewise for evaluating flux $\tilde{\mathbf{F}}_{i-1/2}$ at interface $x_{i-1/2}$, repeat the steps
    (1) to (5).
    7: Set $\tilde{h} = (h^{L,*}_{i+1/2} + h^{R,*}_{i-1/2})/2$ and $\partial_x z = (z^{L,*}_{i+1/2} + z^{R,*}_{i-1/2})/\Delta x$ and use them to
    evaluate the integral term $\mathcal{S}_{i,k}$ in expression 4.34.
```