High-order Adaptive Discontinuous Galerkin Inundation Modeling

Dissertation with the aim of achieving
a doctoral degree at the
Faculty of Mathematics, Informatics and
Natural Sciences
Department of Mathematics of Universität Hamburg

submitted by Nicole Beisiegel

2014 in Hamburg

Day of oral defense: September 17, 2014

The following evaluators recommend the admission of the dissertation

Prof. Dr. Jörn Behrens
CEN, Universität Hamburg
Hamburg, Germany
email: joern.behrens@zmaw.de

and Prof. Dr. Jens Struckmeier
Department of Mathematics, Universität Hamburg
Hamburg, Germany
email: struckmeier@math.uni-hamburg.de

and Prof. Dr. Gregor Gassner
Department of Mathematics, University of Cologne
Cologne, Germany
email: gassner@iag.uni-stuttgart.de

Declaration on Oath

I hereby declare on oath, that I have written the present dissertation on my own and have not used other than the acknowledged resources and aids.

Place, date	signature

Contents

List of Figures

List of Tables

Acknowledgements

This doctoral thesis would not have been possible and such an unforgettable experience without the support of a lot of kind people around me, of whom I am only able to mention some here.

First of all, I acknowledge, with gratitude, my debt of thanks to my advisor Jörn Behrens for his constant encouragement, advice and patience throughout the four years. I appreciate the freedom, he gave me, to conduct the research I was most interested in and his generous support to attend a large number of international conferences.

Furthermore, I am indebted to my co-advisor Cristóbal E. Castro for his dedicated support, scientific advice and the insightful discussions and suggestions.

Likewise, Jochem Marotzke, who chaired my regular advisory panel meetings, has been essential for my progress with his constructive and motivating suggestions of improvement during the panel meetings.

I also thank Jens Struckmeier for his willingness to write the second review for this thesis.

Moreover, I would also like to express my gratitude to my graduate school SICSS, particularly Berit Hachfeld, Ingo Harms and Oliver Dilly, who helped me a lot with organizational matters and supported my travels to a great extent.

Finally, I am grateful for the current and former members of Jörn Behrens' working group for the open and productive working atmosphere. Our regular coffee breaks were and are always a welcome distraction and a boost of creativity. I especially want to express my thanks to Werner Bauer, Oliver Kunst, Andreas Müller and Stefan Vater, not only for partially proof reading this thesis, but also for their constant, encouraging and open-minded support throughout the four years.

Abstract

Starting from the nodal Discontinuous Galerkin (DG) model presented in Giraldo et al. (2002), we developed and validated a quasi-nodal high-order and well-balanced DG model on an adaptive mesh to solve the two-dimensional one-layer, non-linear shallow water equations for the simulation of quasi-realistic inundation events.

The important scales of inundation problems range between several orders of magnitude. However, fine-scale features often occur heavily localized, so that the simulations usually do not need to be finely resolved everywhere in the spatial domain. Therefore, we introduce a dynamically adaptive triangular mesh (see Behrens et al. (2005)), that allows for temporal varying spatial resolution, which reduces the computational complexity. Idealized numerical tests of flooding and drying show a reduction of computational complexity compared to simulations on a uniformly fine mesh.

Spurious oscillations destabilize the DG model and affect the accuracy of the numerical solution. To reduce them, slope limiters are employed in an explicit time-stepping scheme. Moreover, the use of the strong form of the equations leads to a well-balanced model with a computational efficient modification that we suggest. A theoretical study on limiters reveals desirable properties and a comparison highlights the strengths and weaknesses of commonly used limiters. Numerical tests are performed to show the accuracy of the DG method in combination with different limiters.

In order to improve local convergence rates and conservation properties, we study the viability of higher than second order basis functions in the DG model. The usage of higher-order functions is especially desirable because the majority of problems of interest is smooth almost everywhere. The nodal Lagrange basis of the model, however, leads to stability problems. Therefore, we introduce higher-order Bernstein basis functions and show an improvement with idealized numerical tests.

A suitable model for wind stress is essential for storm surge modeling, because the wind is the major driving force in storm surge events. We show a discretization that was first introduced by Holland (1980) and allows to predict wind drag from a given cyclone track using information on the radius of maximum winds as well as the central pressure of the storm. We study idealized storm scenarios in order to detect suitable refinement indicators for the use of the adaptive mesh. First simulation results suggest the practicability of inundation simulations with wind forcings.

Kurzfassung

Ausgehend von dem in Giraldo et al. (2002) vorgestellten nodalen unstetigen Galerkin (DG) Modell entwickelten und validierten wir ein quasi-nodales wohlbalanciertes DG Modell höherer Ordnung auf einem adaptiven Gitter zur Lösung der zwei-dimensionalen einschichtigen, nichtlinearen Flachwassergleichungen zur Simulation von Überflutungsereignissen.

Die wichtigsten Skalen von Überflutungsproblemen erstrecken sich über mehrere Größenordnungen. Jedoch treten feinskalige Merkmale häufig stark lokalisiert auf, so dass die Simulation nicht überall fein aufgelöst sein muss. Daher führen wir ein dynamisch adaptives Dreiecksgitter (siehe Behrens et al. (2005)) ein, das eine zeitliche Variation der räumlichen Auflösung ermöglicht, die die Berechnungskomplexität reduziert. Idealisierte numerische Tests von Überflutungen zeigen eine deutliche Reduktion des Rechenaufwands im Vergleich zu Simulationen auf einem uniformen feinen Gitter.

Numerische Oszillationen destabilisieren das Modell und beeinflussen die Genauigkeit. Um sie zu reduzieren, werden Slope Limiter in einem expliziten Zeitschrittverfahren verwendet. Die Verwendung der starken Form der Gleichungen in Kombination mit einer Modifikation, die wir präsentieren, stellt zudem sicher, dass das Modell wohlbalanciert ist. Eine theoretische Studie von Slope Limiter verdeutlicht wünschenswerte Eigenschaften und ein Vergleich zeigt die Stärken und Schwächen der am häufigsten verwendeten Limiter auf. Wir testen numerisch die Genauigkeit des DG Verfahrens in Kombination mit verschiedenen Limitern.

Um lokale Konvergenzraten und Erhaltungseigenschaften zu verbessern, untersuchen wir die Funktionsfähigkeit von Basisfunktionen höherer als zweiter Ordnung in dem DG Modell. Die Nutzung von Funktion höherer Ordnung ist besonders wünschenswert, weil, die Mehrzahl der Anwendungsprobleme fast überall glatt ist. Der nodale Lagrange Ansatz des Modells führt jedoch zu Stabilitätsproblemen. Daher schlagen wir Bernstein Polynome als Basisfunktionen vor und zeigen mit idealisierten numerischen Tests von Überflutungen eine Verbesserung der Simulation.

Ein geeignetes Modell für den Windschub als Hauptantriebskraft ist essentiell für die Sturmflut-Modellierung. Wir zeigen eine Diskretisierung, die zuerst in Holland (1980) eingeführt wurde und studieren idealisierte Sturmszenarien, um geeignete Indikatoren für die Verfeinerung der adaptiven Gitter zu bestimmen. Erste Simulationsergebnisse deuten darauf hin, dass das Modell für die Anwendung von Sturmflut Simulationen praktikabel ist.

Notation and Abbreviations

α_{ik}	Runge-Kutta coefficients
$\boldsymbol{\alpha}$	multiindex for derivatives on the triangle
β_{ik}	Runge-Kutta coefficients
b	bathymetry in [m]
B_λ^n	λth Bernstein polynomial of degree n
c	wave speed
$\frac{D}{D\cdot}, \frac{\partial}{\partial\cdot}$	material, partial derivative
η_{Ω_e}	error estimator for element Ω_e
e	index for triangle numbering
$\mathcal{E}$	index set for element numbers
$\mathcal{E}_2, \mathcal{E}_\infty$	error measures for non-analytic functions
$E_{rel}^2, E_{abs}^2, E_{rel}^\infty, E_{abs}^\infty$	error measures for analytic functions
$\mathbf{F}, \mathbf{F}^*$	flux tensor, its numerical approximation
$\mathbf{f}$	x-split flux vector
$\mathbf{F}_{cn}$	vector of conservative forces
$\mathbf{f}_c, f$	Coriolis forcing, scale of Coriolis forcing
Fr	Froude number
∇	gradient operator
$\gamma_{\boldsymbol{\tau}}$	wind friction
g	gravitational acceleration of the earth ($\approx 9.81 ms^{-2}$)
h	water height in $[m]$
$H = h + b$	total height
$\mathbf{H}$	right-hand side of ordinary differential equation
$\widetilde{\mathbf{H}}$	limiter
$\mathbf{I}_2$	identity matrix in $\mathbb{R}^{2\times 2}$
$\mathcal{J}, \mathcal{J}_e$	arbitrary index set, arbitrary index set for element Ω_e
Δ	Laplace operator
$\boldsymbol{\lambda}$	multiindex
$\boldsymbol{\lambda}$	eigenvalues of shallow water equations
$L^\infty(\Omega)$	space of bounded functions over domain Ω
μ	modal basis function
$\boldsymbol{\mu}$	also multiindex for Bernstein derivatives
M	number of triangles in domain decomposition
m	number of degrees of freedom per triangle $m = (n+1)(n+2)/2$
m_s	order of Runge Kutta scheme
n	order of polynomial
n_M	Manning's n (dimensionless parameter)
$\mathbf{n}, \mathbf{n}_k$	normal vector, normal vector w.r.t kth edge
N	number of degrees of freedom
ν	physical viscosity coefficient

p	pressure
$\phi = gh, L$	geopotential height and its length scale
$\mathbf{\Phi}$	vector of non-conservative forces
$\mathcal{P}_n$	polynomial space of degree n
$\mathbb{R}, \mathbb{R}^2, \mathbb{R}^3$	space of real numbers in 1, 2 and 3 dimensions
$\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$	shallow water eigenvectors
Δr_e	radius of largest inscribed circle of element Ω_e
ρ, ρ_0	density function, constant density
Re	Reynolds number
Ro	Rossby number
$\boldsymbol{\sigma}$	barycentric coordinates
s	number of stages in Runge-Kutta scheme
S	source term
S_L, S_R, S_*	speeds of characteristics
$\otimes$	tensor product
$\boldsymbol{\tau}$	two-dimensional wind vector field
$\boldsymbol{\tau}_b$	bottom friction forcing
$\boldsymbol{\tau}_s$	wind forcing
t, t_0	time variable, fixed point of time
$T = [0, t_0]$	time interval
Δt	time step
$\theta_{crs}, \theta_{ref}$	parameters for adaptive mesh refinement
$\mathbf{u} = (u, v)^\top, U$	two-dimensional velocity vector and its scale
$\mathbf{U} = (\phi, \phi\mathbf{u})^\top$	balanced quantities
$\mathbf{U}_h$	numerical solution for balanced quantities
$\mathbf{U}_h^{(k)}$	intermediate Runge-Kutta numerical solution at stage k
$\mathbf{U}_L, \mathbf{U}_R, \mathbf{U}_{L*}, \mathbf{U}_{R*}$	state values
ψ_k	kth nodal basis functions
ψ_L	latitude in degree
$\mathbf{x} = (x, y)^\top$	spatial two-dimensional coordinates
$\Delta\mathbf{x}_{min}, \Delta\mathbf{x}_{max}$	shortest and longest edge of adaptive mesh
$\mathbf{x}_i$	Lagrange interpolation points
$\boldsymbol{\xi}_k$	Gauss quadrature points
χ	indicator function
$\mathbf{V}$	Vandermonde matrix
Ω	domain
$\partial\Omega$	boundary of domain Ω
Ω_e	element (triangle) number e
Ω_{rot}	rotation rate of the earth
z	vertical coordinate

1. Introduction

1.1. Motivation and Background

With the recent advances in computer technology and the development of mathematical algorithms, it is possible to compute reasonable solutions to evermore complex problems within acceptable time. Of special interest are geophysical problems, as computer simulations can contribute to improve our understanding of earth system processes, which are, for example, an important part of the simulation of natural hazards. One of them, and the focus of this thesis, are storm surges.

In Holthuijsen (2007) storm surges are defined as a large-scale elevation in a severe storm generated by a (low) atmospheric pressure and high wind speeds with temporal and spatial scales equal to the ones of the generating storm. These strong winds advect water from the ocean towards the coast and flood coastal areas as is depicted in sketch 1.1. In that case, the role of other geophysical forces, such as Coriolis force and bottom friction, strongly depend on local conditions such as the linearity of the waves or the consistency of the bottom topography. Usually it is not the wind, but the water that causes the main damage, which can be devastating to the population and local infrastructure in coastal regions. Not uncommonly, there are up to hundreds of casualties and loss of property to the amount of several hundreds of thousands of euros recorded that are caused by severe storms.

Regularly, reports on current storms and floods can be found in the media, constantly reminding us of the vulnerability of coastal region to be damaged by major floods. An example for a severe storm surge is the surge caused by Hurricane Ike–a Category 4 storm (on the Saffir-Simpson hurricane scale, see appendix chapter A), that traveled from the west coast of Africa on August 28, 2008 over the Atlantic Ocean, hit the Caribbean and Cuba, and finally made landfall as a Category 2 storm at Galveston Bay, Texas on September 12, 2008 (see Berg (2010)). It caused 103 direct deaths and further 94 deaths can be attributed to it, most of them at the U.S. Golf Coast. The total damage amounted to 27.2 billion euros which made it the third costliest hurricane in the United States.

Flood watches and warnings, and the implementation of measures to mitigate effects of major storm surges are therefore essential for the protection of the population. Mostly, they are the responsibility of national governments. For example in Germany, according to the law (Seeaufgabengesetz, or short, SeeAufG) it is the responsibility of the federal government to reside the nautical and hydrographic services which includes

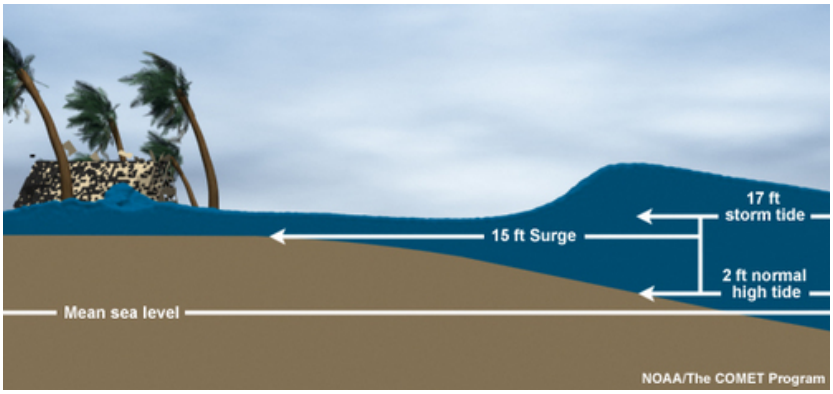

Figure 1.1.: Sketch of a storm surge, taken from NOAA (2013).

the prediction of water level, tides and storm surges. For the performance, a network of local hydrographic service centers is appointed, that employ numerical models for simulations and have a responsibility to take local measurements to improve the forecast if necessary. In the USA the respective hydrographic service is the National Oceanic and Atmospheric Administration (NOAA) in Washington, DC. They also have a range of numerical models, both deterministic and stochastic, for long- and short-term predictions of the water level. Hence, the ability to predict the run-up of a surge is crucial information for a large number of institutions, because it is needed for hazard prediction tools which are fundamental for decision making. These decisions concern emergency warning and evacuation of the local population as well as modification of means of mitigation such as dike or levee systems.

The computer models, that are employed for the simulation and prediction, are required to be robust, efficient and accurate. For operational purpose, usually a combination of deterministic and stochastic models is used.

The latter take storm-specific uncertainties into account and are mostly computationally inexpensive relative to deterministic models. However, they strongly depend on the availability and quality of measurement data, which is often hard to acquire. Examples for stochastic models are the model developed in Wahl et al. (2012a,b) for the German Bight or NOAA's P-Surge (Tropical Cyclone Storm Surge Probabilities). P-Surge computes cyclone tracks by statistically evaluating ensemble runs of an analytical model called SLOSH (Sea, Lake, and Overland Surges from Hurricanes) and takes historical errors for track and intensity into account. Mostly, operational stochastic models are employed in a time window that is close (between 24 to 48 hours) to the landfall of the storm.

For longer-term prediction, usually deterministic models are employed. They are advanced models that compute inundation scenarios from initial measurement data using mathematical equations that are suitable to describe the present physical processes. Examples are NOAA's MOM (Maximum of the Maximum) and MEOW (Maximum Envelop of Water Height), that compute worst case scenarios for the water height un-

der perfect storm conditions as well as the finite element model developed in Westerink et al. (2008) which has most prominently been applied to the Louisiana coast.

During the past decades of model development huge progress has been made with respect to practical applicability. But even with the resources that are availably nowadays, the results are still improvable. For example, there are a lot of multi-scale processes that are not taken into account in the equations that are solved, the models are not fast enough to provide sufficient resolution on small scales within acceptable time, and numerical models need further development to be able to yield robust results with a reduced application of numerical stabilization techniques.

The overall goal of our study is to develop a numerical method to accurately and efficiently simulate inundation events in order to gain knowledge for future improvement of storm surge simulations and predictions.

1.2. Objective

The main goal of this thesis is the methodological improvement of discontinuous Galerkin (DG) methods for the simulation of storm surges. This comprises aspects of computational complexity as well as numerical accuracy. We are concerned with reducing the computational cost of the simulations without significantly affecting the overall accuracy. The latter is strongly linked to the application of slope limiters, which are used to reduce numerical oscillations in order to stabilize the model and obtain physically meaningful solutions. In all, our goals can be formulated as three mostly independent research questions:

- To which extent does varying spatial resolution in a DG model reduce the computational complexity and, in turn, how does it affect the overall accuracy and robustness of the method, especially for flooding and drying scenarios?
- Does a theoretical formulation of general slope limiters for the reduction of numerical oscillations exist; how does it help to determine their strengths and weaknesses in handling flooding and drying and can the results be confirmed numerically?
- Are third-order basis functions viable for the simulation of wetting and drying?

The objective is accomplished by the development of a two-dimensional single-layer and high-order adaptive nodal DG shallow water model. A three-dimensional or multi-layer model could achieve a higher accuracy. However, these approaches are known to be computationally expensive compared to a two-dimensional model and due to a strong limitation of computational resources, we do not pursue them. We implemented the model within the existing software framework `amatos` (see Behrens et al. (2005)), that provides routines for the use of adaptive meshes. Following the framework's naming conventions, it is named `StormFlash2d`. It has been tested with a set of idealized test cases, most of them which admit an analytical solution for comparison, to demonstrate the major functionalities: accurate modeling of flooding and drying, robustness on a

dynamically adaptive mesh, preservation of steady states, and correct representation of wind forcing.

1.3. Overview

This thesis is organized as follows. Chapter 2 provides a mathematical derivation of the shallow water equations, which we will use throughout this monograph. We review the general characteristics of the system and a set of dimensionless numbers, that classify the strength of certain effects. In the end, we discuss the physics of storm surges, how they relate to the equations and which effects are not taken into account in our model.

In chapter 3 we introduce our nodal discontinuous Galerkin model with its basic functionalities: an adaptive triangular mesh and slope limiters to simulate wetting and drying. Numerical testcases demonstrate the ability to model flooding and drying and to preserve steady state solutions with linear basis functions. An investigation of computational complexity for the combination of our nodal DG model with an adaptive triangular mesh reveals a considerable decrease of computing time without any significant loss of accuracy for all testcases.

An accuracy analysis is performed in chapter 4, i.e. we investigate the impact of different numerical approximations of our model on the accuracy. This comprises the theoretical and numerical comparison of different limiting strategies that are employed for the stable computation of wetting and drying, the investigation of the influence of Riemann solvers that are used to compute fluxes over cell boundaries as well as our method to preserve steady state solutions.

Furthermore, in chapter 5, we explore strategies for the usage of higher than second order basis functions which requires a modification of the used basis functions and leads at least locally to an improvement of convergence.

Chapter 6 presents simulation results for storm-based inundation. This comprises the discretization of source terms in our model and the study of suitable refinement indicators for the adaptive mesh. On idealized storm surge testcases we show the validity of our storm implementation and the capability of our model to adaptively simulate storm surge events.

Finally, in the last part of this thesis, chapter 7, we summarize our findings and discuss to which extent higher than second order modeling of inundation scenarios is advantageous and the performance of our model applied to storm surges. The chapter closes with a list of open research questions in this context, that are left for future investigation.

2. Mathematical Description of Storm Surges

The deterministic computational modeling of storm surges requires a precise mathematical description of the involved geophysical processes. But this description can only be accurate to a certain degree, as we have to make assumptions to derive general laws of physical mechanisms. These assumptions are then present in the underlying mathematical equations. They will lead to drawbacks of the model that we will develop and validate for their solution in chapters 3–6. In the following, we will mainly focus on two aspects: a description of the one-layer two-dimensional shallow water equations, that we will use throughout this monograph, and their deficiencies with respect to the modeling of flooding and drying, as well as the inherent wave patterns of the corresponding homogeneous equations.

2.1. Shallow Water Physics

Starting point of our derivation are the Navier Stokes equations as in McWilliams (2006), which are a set of partial differential equations, defined on a domain $\Omega \subset \mathbb{R}^3$, that are used to describe the general motion of a fluid by considering the temporal change of its density $\rho[kg/m^3] \in L^\infty(\Omega)$ and velocity $\mathbf{u}[m/s] = (u, v, w)^\top \in \left(L^\infty(\Omega)\right)^3$. The expressions in square brackets indicate the corresponding units for each component. The approach in McWilliams (2006) is of Eulerian kind, i.e. we fix one specific point $\mathbf{x}$ and observe the general motion of the fluid in contrast to a Lagrangian perspective, where individual fluid parcels are taken into account. We will refer to both space- and time-dependent quantities, $\rho = \rho(\mathbf{x}, t)$ and $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$, as balanced quantities, where $\mathbf{x} = (x, y, z)^\top$ are the spatial variables and t is the time variable. The equations themselves are balance equations of the form

$$\frac{D\rho}{Dt} = 0 \tag{2.1a}$$

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \nabla \cdot \mathbf{\Phi} + \mathbf{F}_{cn} \tag{2.1b}$$

where $\frac{D}{Dt}$ is the material derivative defined as the sum of the partial temporal derivative and the inner product of velocity times spatial gradient with respect to $\mathbf{x}$, which can be written as $\frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla$. The other effects, that are taken into account, are due to pressure $p[Pa]$, a vector $\mathbf{F}_{cn}[m^2 s^{-1}]$ that includes for example the conservative Coriolis forcing, which will be described later, and a term $\mathbf{\Phi}[X m^2 s^{-2}]$ that comprises all non-

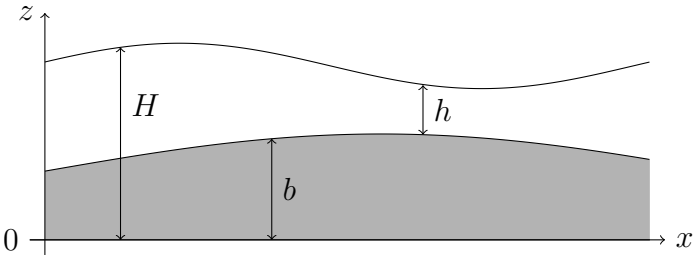

Figure 2.1.: Shallow Water Equations. Sketch of bathymetry b, total height H and fluid height h.

conservative forces such as viscosity, wind stress and bottom friction. Equation (2.1a) is known as continuity equation and the second to fourth equations in (2.1b) are the momentum balances.

For computational purposes, the equations (2.1) are too complex and, as we will see later, a reduced set of equations, the shallow water equations, will also lead to satisfying results. In the following, we will list the shallow water assumptions and approximations and refer to the appendix, section B.1, for a detailed derivation. Furthermore, we introduce the notations h for the fluid height, b for the bathymetry and $H = h + b$ for the total height, which are also illustrated in figure 2.1:

- We assume that the vertical extent of the waves of the system is small compared to the depth of the water, so that the vertical velocities w are assumed to be negligible, which leads to a hydrostatic pressure p which only depends linearly on the vertical position z.

- With a slight abuse of notation, we denote the now two-dimensional domain of interest with $\Omega \subset \mathbb{R}^2 \times [0, t]$ with $t \in \mathbb{R}$, the spatial variable with $\mathbf{x} = (x, y)^\top$ and the now only two-dimensional horizontal velocity with $\mathbf{u}[m/s] = (u, v)^\top$.

- The density of the fluid is assumed to be constant, $\rho(\mathbf{x}, t) \equiv \rho_0 \in \mathbb{R}$. A depth integration yields the fluid height h, over which the horizontal velocities u, v are then depth-averaged.

- We assume no-slip boundary conditions, i.e. $u = v = 0$, at the lower vertical boundary with $z = b$ and no-normal-flow conditions at the free surface, i.e. $\frac{D}{Dt} = 0$, at $z = H$.

- The source terms are

$$\begin{aligned} \mathbf{F}_{cn} &= \mathbf{f}_c \\ \nabla \cdot \mathbf{\Phi} &= -\boldsymbol{\tau}_s + \boldsymbol{\tau}_b - \nu \Delta \mathbf{u} \end{aligned}$$

  where Δ is a Laplace operator with respect to $\mathbf{x}$, ν is a physical viscosity parameter, $\mathbf{f}_c$ the Coriolis forcing, $\boldsymbol{\tau}_s$ a wind stress and $\boldsymbol{\tau}_b$ bottom friction forcing.

With all these assumptions, we obtain equations for the new balanced quantities $\mathbf{U} := (\phi, \phi\mathbf{u})^\top$, where now $\mathbf{u}$ is a two-dimensional depth-averaged velocity and $\phi = gh$ is the geopotential fluid height with $g \approx 9.81[ms^{-1}]$ the gravitational constant.

$$\frac{\partial \phi}{\partial t} + \frac{\partial \phi u}{\partial x} + \frac{\partial \phi v}{\partial y} = 0 \tag{2.2a}$$

$$\frac{\partial \phi \mathbf{u}}{\partial t} + \nu \Delta \mathbf{u} + \nabla \cdot ((\phi \mathbf{u}) \otimes \mathbf{u}) + \phi \nabla(\phi + b) + \mathbf{f}_c - \boldsymbol{\tau}_s + \boldsymbol{\tau}_b = 0 \tag{2.2b}$$

where Δ is a Laplace operator and ∇ is the gradient with respect to $\mathbf{x}$, $\otimes$ is a tensor product on $\mathbb{R}^2 \times \mathbb{R}^2$. All vector-valued quantities are indicated with a bold print. With the choice of appropriate boundary conditions, this is a well-posed problem, i.e. a unique solution exists that depends continuously on the initial data. A detailed description of the discretization of the terms from (2.2) can be found in chapters 3 and 6.

In summary, we consider the major effects, that are important for the simulation of storm surges: viscosity, advection, pressure gradients, Coriolis forcing, wind, and bottom friction. The effect of the different components depends on external factors, such as the spatial location and the presence of winds. Some insight can be obtained by introducing dimensionless numbers in the following subsection.

Characterization by Dimensionless Numbers

The characterization of oceanic flow and the severity of external effects can partly be estimated with dimensionless numbers. Important for their assessment are typical scales of the inhered quantities. We will denote the characteristic length scale of the system with L and the characteristic flow velocity with U. The quantity ν will be the physical viscosity parameter as described above, f represents the Coriolis forcing and c the wave speed of the system. For shallow water equations, this is equivalent to the speed of the gravity waves. A detailed description of phase speeds and their derivation will be given in the next subsection 2.1.1.

The numbers, that we are interested in, are defined as

$$Re \equiv \frac{U\ L}{\nu}, \qquad Fr \equiv \frac{U}{c}, \qquad Ro \equiv \frac{U}{L\ f}. \tag{2.3}$$

We will call Re Reynolds number, Fr Froud number and Ro Rossby number.

The Reynolds number Re is defined as the quotient of momentum divided by viscosity and characterizes the impact of viscosity on momentum. For advection dominated flows this number is usually large, i.e. of the order 10^6 (see McWilliams (2006)).

The Froude number Fr gives an estimate for the relation between the characteristic speed of gravity waves c and the flow velocity U. It allows to characterize the flow as critical ($Fr \approx 1$), subcritical ($Fr < 1$), or supercritical ($Fr > 1$), which means, that the propagation of perturbations remains stationary, is directed in the opposite direction of the flow velocity, or the same direction of the flow velocity respectively.

Finally, the Rossby number *Ro* states the relation between the influence of Coriolis forcing and characteristic velocities. Especially for quasi-realistic applications it can be seen as a good indicator of whether Coriolis forcing has to be taken into account.

All these dimensionless numbers are of help when it comes to estimating the effect of the different terms in our equations (2.2) onto the numerical solution in different regimes. More detail on their values for a specific case can be found in chapter 6.

2.1.1. Characteristics of the Homogeneous System

If we neglect the source terms, we get the so called homogeneous shallow water equations. The study of a simplified homogeneous system is essential for understanding the physics of the full system. We are interested in analyzing an x-split system of equations of the form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{U})}{\partial x} = 0, \tag{2.4}$$

where $\mathbf{U} = (\phi, \phi\mathbf{u})^\top$ is the three-dimensional vector of conserved quantities, and $\mathbf{f}(\mathbf{U}) = (\phi u, \phi u^2 + 0.5\phi^2, \phi uv)^\top$ is the flux. The system (2.4) can be characterized as hyperbolic, which by definition means, that the waves of the system travel with the speed of real and distinct eigenvalues and are separable.

For our investigations, we will consider a simple initial value problem, that is closely related to the modeling of fluxes over element boundaries in DG methods, which we will describe later in this section. We assume constant initial values at each side of the boundary, which is known as Riemann problem (see the right display of figure 2.2). We will denote the side-wise constant values with $\mathbf{U}_L = (\phi_L, \phi_L\mathbf{u}_L)^\top$ and $\mathbf{U}_R = (\phi_R, \phi_R\mathbf{u}_R)^\top$ for the left element and the right respectively and study the general characteristics and the evolving waves.

An eigen-structure analysis using the Jacobian matrix defined as $\frac{D\mathbf{f}}{D\mathbf{U}} := \left(\frac{\partial \mathbf{f}_i}{\partial \mathbf{U}_j}\right)_{1\leq i,j\leq 3}$ reveals, that the three eigenvalues λ_k and eigenvectors $\mathbf{r}_k, k = 1, 2, 3$ have the following structure (as can be found for example in LeVeque (2002)):

$$\begin{aligned}
\lambda_k &= u + \mathbf{i}_k\sqrt{\phi}, \quad \text{with } \mathbf{i} = (-1, 0, 1)^\top \text{ and } k \in \{1, 2, 3\},\\
\mathbf{r}_1 &= (1, u - \sqrt{\phi}, v)^\top,\\
\mathbf{r}_2 &= (0, 0, 1)^\top,\\
\mathbf{r}_3 &= (1, u + \sqrt{\phi}, v)^\top,
\end{aligned} \tag{2.5}$$

Studying the elementary wave solution of the problem, we find a wave corresponding to every λ_k characteristic field. The wave pattern of shallow water waves can mainly be characterized as in figure 2.2. The solid lines refer to non-linear waves (shocks or rarefaction waves) that correspond to the λ_1- and λ_3-fields and the dotted line indicates a contact discontinuity corresponding to the λ_2-field, i.e. a wave that only allows a discontinuity in the zonal velocity v. We will denote the speed of the waves

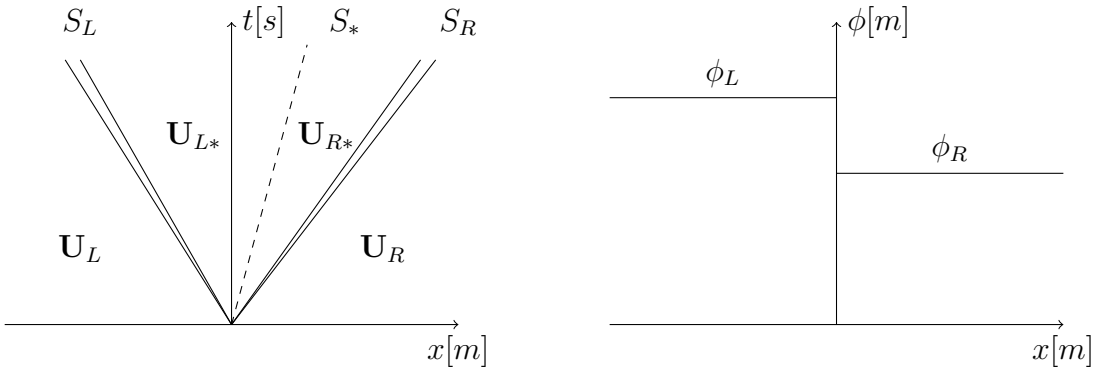

Figure 2.2.: Riemann problem for shallow water equations. Depicted are the elementary wave solution over time (left) and the initial condition of the geopotential height ϕ (right).

with S_L, S_* and S_R. They partition the domain into four regions in which the prognostic variables can be characterized as $\mathbf{U}_L, \mathbf{U}_{L*}, \mathbf{U}_{R*}$ and $\mathbf{U}_R$.

As was shown for example in Toro (2001), the presence of a wet/dry interface, i.e. $\mathbf{U}_R = \mathbf{0}$ and $\mathbf{U}_L \neq \mathbf{0}$, prohibits the development of shock waves, so that all non-linear waves will be smooth rarefactions. Another scenario, which prevents the forming of shock waves, is a linearization of the underlying equations.

The reason why the study of Riemann solvers, i.e. approximate or exact solutions for the Riemann problem, of the x-split system is meaningful for our development of a two-dimensional DG model is the following. One part of DG models is the computation of fluxes over element boundaries. This comprises the pointwise solution of one-dimensional Riemann problems, because only the normal component with respect to the corresponding edge of vector-valued quantities is of interest. Therefore, the question of finding a representation of the edge fluxes reduces to finding an appropriate solver for the Riemann problem. In section 4.4 (or Toro (2001) for more detail) we find solvers for the one-dimensional case. We can easily apply these solvers to the two-dimensional case, because only the normal-direction of vector-valued quantities is of interest and the shallow water equations are rotation-invariant. This means that any one-dimensional solver can be used if we rotate the system of eigenvalues into the normal direction, apply the solver and then rotate the result back.

2.1.2. Drawbacks of the Shallow Water Approximation

As we have seen, shallow water equations provide a reasonable model for the major processes, that are important for inundation scenarios, although certain limitations have to be recognized.

The shallow water assumption is violated for waves approaching the coast, because the horizontal extent of the waves is then large compared to the depth of the water. Furthermore, assuming a two-dimensional model for the ocean will only give good

results in regimes where the vertical velocity of the fluid is negligible. A hydrostatic correction, as has been developed for example in Fuchs (2013) for a continuous Galerkin model, can improve the results in that case at the cost of solving a huge linear system.

Some further effects such as turbulence are completely neglected and are in general a delicate task that would have to take into account scales that are much smaller than the ones we are studying in our current work.

Further limitations are given by assumptions on the fluid itself such as wave breaking not being modeled. There are approaches though, that solve larger and therewith more expensive systems with shallow water equations using multiple phases that allow for wave breaking as is shown for example in Dumbser (2011) for a finite volume model. The assumption of constant density implies that, non-Newtonian fluids are not respected in the model. This limitation plays an essential role for flooding scenarios where severe water masses run on shore and transport infrastructure into land, which causes severe damages.

Finally, we assume that the bathymetry is constant over time and no sedimentation is taking place. In the case of storm surges, though, this will prevent levees and dikes from being damaged by water soaking in, so that events, that worsen through dike and levee breaking, are not considered. An approximation on whether levees and dikes will remain steady is, however, in principle possible with a post-processing estimate.

3. Development of an Adaptive Discontinuous Galerkin Inundation Model

3.1. Abstract

A nodal Discontinuous Galerkin (DG) model in combination with a conforming and adaptive triangular mesh solves the two-dimensional non-linear shallow water equations. A dynamically adapted mesh reduces the computational complexity – a major drawback of DG methods compared to commonly used methods such as finite volume and continuous finite elements – and shows the potential to correctly represent complex domain geometries. Numerical tests are performed to show the accuracy and efficiency of the method. With a limiter that is total-variation-bounding and positivity-preserving, the model handles wetting and drying.

3.2. Introduction

The main purpose of this chapter is to show that dynamically adaptive triangular meshes in combination with a well-balanced Discontinuous Galerkin (DG) method, applied to two-dimensional shallow water equations (SWE), lead to a robust and computationally efficient scheme for the simulation of simplified inundation scenarios.

Besides DG methods, other advanced numerical methods for partial differential equations can be found in current literature. Continuous Galerkin (CG) methods have been developed in Westerink et al. (2008), Kärnä et al. (2011), Heniche et al. (2000) and Kerr et al. (2013). They assume an at least globally continuous approximation of the prognostic variables. Reasons for their success are the (high-order) accuracy and the geometric flexibility of the finite elements. Another common approach is the application of finite volume (FV) schemes. These take advantage of the inherent conservation properties of the shallow water equations. Examples can be found in Chen et al. (2008); Zhang et al. (2004). For operational purposes, the aforementioned models are usually run with local approximation order of less than two. However, it is shown in Hesthaven and Warburton (2008) that low-order basis functions theoretically lead to low-order numerical errors. Consequently, a high-order model would reduce numerical errors as recent developments in Zhang and Baptista (2008) and Giraldo and Restelli (2009) show for FV and DG methods respectively.

We will concentrate on DG methods in this monograph since the physics-based way of computing fluxes between elements leads to local conservation of mass and momentum

as shown in Hesthaven and Warburton (2008) –usually a typical feature of FV methods. Moreover, they allow for a high-order representation, thus taking advantage of the above mentioned error minimization. Dealing with the computational expense caused by the relatively large number of degrees of freedom compared to other numerical methods will thus be one main aspect of this chapter.

In the following we will use dynamically adaptive grids, i.e. grids with a temporal locally varying spatial resolution, that is high in regions of interest and relatively low beyond. This modification increases the efficiency, does not significantly affect the accuracy and leads to a robust scheme in combination with our DG method. The grid itself consists of conforming triangles which are suitable for representing coastlines accurately up to polygonal representation. Its adaptation will be determined via a predefined refinement indicator that takes for example geometrical features of the domain or physically relevant quantities into account. For inundation problems, the representation of a large variety of different scales—ranging from hundreds and tens of kilometers in the open sea to just a few meters or smaller at the inundated boundary—is a further asset.

One challenge of inundation modeling is to accurately compute interfaces between wet and dry cells, because non-linear numerical models tend to produce oscillations in the vicinity of steep gradients and discontinuities, which can result in unphysical negative fluid heights, which are usually avoided. Though, a scheme, that is able to deal with negative water heights is shown in Heniche et al. (2000). In this respect, the concept of total variation bounded (TVB) methods that minimize unphysical oscillations and ensure boundedness of a measure for the oscillatory behavior over time, is of interest. The development of wetting and drying schemes is an active field of research. Particularly in operational models, estimators for the inundation run-up are used (see for example the one-dimensional Boussinesq model in Kennedy et al. (2012) for the model presented in Westerink et al. (2008)). Computing a boundary of zero water height separating wet from dry areas in every time step—also known as moving boundaries—is another approach presented in Leclerc et al. (1990). Yet another possibility is the use of limiters to ensure positivity of the fluid height as well as the TVB property. This has been carried out for example in Kuzmin (2010) for a vertex-based limiter which is based on an idea first presented in Bell et al. (1988) and will be discussed in section 4.3.1 in more detail. The main principle behind this limiting is a local Taylor expansion including element mean values and scaling factors in case of oscillations. An alternative is performed for the models in Chen et al. (2008) and Gopalakrishnan and Giraldo (2014): Artificially wetting of physically dry domains in order to overcome stability problems with non-positive fluid heights. In addition thereto in the model presented in Gopalakrishnan and Giraldo (2014) a minmod-based limiter is applied to remove oscillations. However, the artificial wetting lead to a different representation of the waves of the system.

Moreover, an approach has been developed in Kärnä et al. (2011) that includes a transformation of the bathymetry such that with the modified bathymetry, the whole domain is wet and no special treatment has to be developed for dry cells. For our studies, we will not pursuit this approach. We employ limiters to our model that are strong enough to suppress most of the artificial oscillations while neither effecting the accuracy nor the non-negativity of the water height.

Now, building in parts on the model of Giraldo et al. (2002) we basically introduce three extensions:

- Spurious oscillations that violate the physics of the problem (positivity of the fluid height) are reduced with slope limiters,
- A flux correction of the strong form of the equations is introduced in order to correctly approximate steady state solutions and
- The underlying triangular mesh is dynamically adapted by bisection according to heuristic refinement indicators.

We show numerically, that the above mentioned extensions lead to a scheme that is both, positivity preserving and wellbalanced. The numerical test cases also demonstrate, that the model handles wetting and drying accurately and that the adaptive mesh reduces computing time essentially compared to a uniform simulation on a fine grid.

3.3. Numerical Method

We employ the two-dimensional shallow water equations, that were derived in section 2.1, for modeling inundation. The hyperbolic part of the equations is known to be able to develop shocks in finite time even for smooth initial data. These phenomena are captured with a method that solves the equations in a distributional sense and is capable of representing discontinuities in the numerical solution.

Building in parts on the well-established nodal DG model presented in Giraldo et al. (2002), we introduce a dynamically adaptive triangular mesh that is generated using the library `amatos` which is introduced in Behrens et al. (2005). The explicit time integration of the resulting semi-discrete equations will be complemented with limiting techniques to model wetting and drying.

3.3.1. Discontinuous Galerkin Method

Our numerical model is based on the generic balance equations in flux form

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = S(\mathbf{U}) \qquad \text{in } \Omega \times T \tag{3.1}$$

where $\mathbf{U} = (\phi, \phi\mathbf{u})^\top$ are the balanced quantities, $T \subset \mathbb{R}$ is a compact subset of the real numbers and $\Omega \subset \mathbb{R}^2$ is a two-dimensional spatial model domain. The flux function $\mathbf{F}$ and the source term S are given by

$$\mathbf{F}(\mathbf{U}) = \begin{pmatrix} \phi\mathbf{u} \\ \phi\mathbf{u} \otimes \mathbf{u} + \frac{1}{2}\phi^2\mathbf{I}_2 \end{pmatrix}, \qquad S(\mathbf{U}) = -\begin{pmatrix} 0 \\ \mathbf{f}_c + \phi\nabla b - \boldsymbol{\tau}_s + \boldsymbol{\tau}_b \end{pmatrix}, \tag{3.2}$$

and are known as the one-layer two-dimensional shallow water equations.

If not indicated otherwise, we will always assume that the spatial coordinates are $\mathbf{x} = (x, y)^\top$. We further use the following notation: $\phi = gh$ is the geopotential height,

$\mathbf{u} = (u, v)^\top$ the velocity, g the gravity constant, h the fluid height and $\mathbf{I}_2$ indicates the 2×2 identity matrix. The source term consists of bottom forcings including a temporal constant bathymetry $b = b(\mathbf{x})$, bottom friction forcing $\boldsymbol{\tau}_b$, Coriolis forcing $\mathbf{f}_c$ and wind stress $\boldsymbol{\tau}_s$. Throughout this thesis vector-valued quantities in $\mathbb{R}^2$ will be indicated with bold print, other quantities are assumed to be scalar.

The system of equations only has a well-defined solution in combination with compatible boundary conditions. They are problem-dependent and can be specified as e.g. reflecting or transmissive boundaries. Imposing additional forcings such as tidal forcing is also done for some simulations. If not stated differently, we will work with reflecting boundary conditions defined as

$$\mathbf{u} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega,$$

where $\mathbf{n}$ is the outward pointing normal with respect to the domain boundary $\partial\Omega$.

We obtain a set of semi-discrete equations by rewriting the system of equations (3.1) as

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) - S(\mathbf{U}) = 0 \qquad \text{in } \Omega \times T.$$

The standard Galerkin assumption implies that the residual is orthogonal to the space of test functions. The latter is spanned by the element-wise nodal Lagrange polynomial basis of degree n, $\mathcal{P}_n = \text{span}\left\{\bigcup_{j\in\mathcal{J}} \psi_j\right\}$, to be further specified later on. For the moment, we assume that $\mathcal{J}$ is an arbitrary index set. Employing the standard $L_2(\Omega)$ inner product for $\Omega \subset \mathbb{R}^2$ leads to

$$\int_\Omega \left(\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) - S(\mathbf{U}) \right) \psi_j(\mathbf{x}) d\mathbf{x} = 0$$

for all $t \in T$ and all indices $j \in \mathcal{J}$. A division into local problems can be obtained by domain decomposition. Here we decompose Ω into M non-overlapping and conforming triangles of the form

$$\Omega = \bigcup_{e=1}^{M} \Omega_e, \tag{3.3}$$

which will allow us to perform all the subsequent operations on a reference element as long as metric terms are respected in the computations. Note that the domain decomposition also led to a decomposition of the index set $\mathcal{J} = \bigcup_e \mathcal{J}_e$ with $\mathcal{J}_e$ denoting the indices of the local Lagrange polynomial basis on the triangle Ω_e.

Integration by parts of the flux tensor and replacing the edge boundary flux with $\mathbf{F}^*$—a numerical approximation of it—leads to the weak form of the equations:

$$\int_{\Omega_e} \left(\frac{\partial \mathbf{U}}{\partial t} - S(\mathbf{U}) \right) \psi_j(\mathbf{x}) \, d\mathbf{x} - \int_{\Omega_e} \mathbf{F}(\mathbf{U}) \cdot \nabla \psi_j(\mathbf{x}) d\mathbf{x} = - \int_{\partial\Omega_e} \mathbf{F}^* \, \psi_j(\mathbf{x}) \cdot \mathbf{n} \, dS.$$

for all $t \in T$, and indices $j \in \mathcal{J}_e$. Repeating this procedure leads to the strong form as in Hesthaven and Warburton (2008) that we will exclusively use throughout this paper for well-balancing reasons, that we will describe in detail in section 4.5:

$$\int_{\Omega_e} \left(\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) - S(\mathbf{U}) \right) \psi_j(\mathbf{x}) \, d\mathbf{x} = - \int_{\partial\Omega_e} \left(\mathbf{F}^* - \mathbf{F}(\mathbf{U}) \right) \psi_j(\mathbf{x}) \cdot \mathbf{n} \, dS. \quad (3.4)$$

for all $t \in T$, and $j \in \mathcal{J}_e$. Rearranging the equations leads to

$$\begin{aligned} \int_{\Omega_e} \frac{\partial \mathbf{U}}{\partial t} \, \psi_j(\mathbf{x}) \, d\mathbf{x} = & \int_{\Omega_e} \left(-\nabla \cdot \mathbf{F}(\mathbf{U}) + S(\mathbf{U}) \right) \, \psi_j(\mathbf{x}) \, d\mathbf{x} \\ & - \int_{\partial\Omega_e} \left(\mathbf{F}^* - \mathbf{F}(\mathbf{U}) \right) \, \psi_j(\mathbf{x}) \cdot \mathbf{n} \, dS. \end{aligned} \quad (3.5)$$

for all indices $j \in \mathcal{J}_e$ with $\mathbf{F}^*$ the numerical flux obtained from solving the Riemann problem (see section 2.1.1) with any standard Riemann solver at cell interfaces $\partial\Omega_e \cap \partial\Omega_f \neq \emptyset$ for $e \neq f$. A detailed presentation and a numerical comparison of Riemann solvers is given in section 4.4. The discretization leads to a time step restriction of the form

$$\Delta t \leq \max_{\Omega_e} \frac{\Delta r_e}{\max_{\partial\Omega_{e,k}} \langle \mathbf{u}, \mathbf{n}_k \rangle + \sqrt{\phi}} \quad (3.6)$$

with Δr_e being the radius of the inscribed circle of Ω_e and $\partial\Omega_e = \sum_k \Omega_{e,k}$, is the boundary of Ω_e, where $\Omega_{e,k}$ is the edge of index k, $k \in \mathbb{N}$, of element Ω_e, $\mathbf{n}_k$ the corresponding outward pointing normal, and $\langle \cdot, \cdot \rangle$ the Euclidean inner product.

Assuming that the numerical approximations $\mathbf{U}_h \approx \mathbf{U}, \mathbf{F}(\mathbf{U}_h) \approx \mathbf{F}(\mathbf{U})$ and $S(\mathbf{U}_h) \approx S(\mathbf{U})$ are elements of $\mathcal{P}_n$ with time-dependent coefficients of the form

$$\mathbf{U}_h(\mathbf{x}, t) = \sum_{k \in \mathcal{J}_e} \mathbf{U}_k(t) \psi_k(\mathbf{x}),$$

one of the remaining tasks now is to find an appropriate quadrature formula to solve the integrals in (3.5). Therefore, we follow the implementations from Hesthaven (1998); Hesthaven and Warburton (2008) to construct an interpolatory Gauss Lobatto quadrature (see Blyth and Pozrikidis (2006)) for the inner element quadrature, that is known to minimize the Lebesgue constant of the underlying interpolation problem. The boundary integrals are approximated with a Legendre-Gauss quadrature. For polynomials of order $n \in \mathbb{N}$ the resulting quadrature is of order $2n$, i.e. they give the exact result for polynomials of at most order $2n$.

Bringing all terms that do not contain a temporal derivative to the right-hand side and inverting the so-called mass matrix $\mathbf{M}$ with entries $\mathbf{M}_{kj} := \int_{\Omega_e} \psi_k \psi_j d\mathbf{x}$, (3.5) can be written as a semi-discrete system of (ordinary differential) equations (ODEs) of the

form

$$\frac{\partial \mathbf{U}_h}{\partial t} = \mathbf{H}(\mathbf{U}_h), \tag{3.7}$$

where a detailed presentation of the now discrete operator $\mathbf{H}$, which contains the flux and source terms multiplied by $\mathbf{M}^{-1}$, can be found in section 4.5. The system (3.7) can in principle be solved using any time-integrator known for ODEs.

For reasons of computational efficiency, we will solve (3.7) explicitly with an s-stage strong stability preserving Runge Kutta scheme of order m_s (RK$m_s s$). Denoting the numerical solution with $\mathbf{U}_h$ the scheme for computing an update of step $\mathbf{U}_h^{n+1}$ from known values $\mathbf{U}_h^n$ takes the form:

$$\begin{aligned} \mathbf{U}_h^{(0)} &= \mathbf{U}_h^n \\ \mathbf{U}_h^{(i)} &= \sum_{l=0}^{i-1} \widetilde{\mathbf{H}} \left(\alpha_{il} \mathbf{U}_h^{(l)} + \beta_{il} \Delta t \mathbf{H}\big(\mathbf{U}_h^{(l)}\big) \right) \qquad \text{for } i = 1, \dots s \\ \mathbf{U}_h^{n+1} &= \mathbf{U}_h^{(s)} \end{aligned} \tag{3.8}$$

with α_{il} and β_{il} the corresponding Runge Kutta coefficients, where the resulting time step restriction can be measured with an effective cfl number as defined in Gottlieb (2005):

$$\mathrm{cfl}_{\mathrm{eff}} = \min_{i,l} \frac{\alpha_{il}}{|\beta_{il}| \cdot s}.$$

In (3.8), $\widetilde{\mathbf{H}}$ is the slope-limiter of the prognostic variables, that is used to eliminate unphysical oscillations in the numerical solution while not affecting the accuracy. We will shortly comment on the choice of slope limiters in section 3.3.3. In sections 4.3.1 and 4.3.3 we carried out an extensive study on general characterizations of limiters and a numerical convergence study.

3.3.2. Adaptive Triangular Mesh

The localized multi-scale nature of the underlying problem is taken into account through the use of a dynamically adaptive mesh, i.e. the mesh, that we introduced in the last paragraph, will allow temporal changing spatial resolution in the computational domain. Hence, our model will be able to perform more accurate computations in domains where fine-scale features are present while it computes relatively coarse numerical solutions outside of these regions.

The meshes we use will solely consist of triangles in order to obtain a better representation of bathymetric data. Finer triangles are obtained through bisecting a coarse triangle as is suggested in Bänsch (1991). For convenience, the mesh will be kept conforming, or free of hanging nodes, by the use of patch-wise refinement and coarsening during the computations. We stress that this is not a requirement of the method itself. Hanging nodes would lead to necessary modification of the boundary integral since they

impose the need to combine two or more Riemann solutions over one (coarse) edge as is for example recently done in Kopera and Giraldo (2013) and Hermann et al. (2011).

The dynamic adaptation of the mesh involves problem-dependent refinement indicators $\eta_{\Omega_e} \in \mathbb{R}$ for each element Ω_e as in Behrens (2006), that control the refinement and coarsening as well as user-defined tolerances $0 \leq \theta_{crs} < \theta_{ref} \leq 1$. One heuristic example that will also be used for the later computations is the divergence of the geopotential height

$$\eta_{\Omega_e} = \max_{\mathbf{x} \in \Omega_e} \nabla \cdot \phi(\mathbf{x}, t).$$

The refinement will then depend on the tolerances as well as on the maximum value of the refinement indicator $\eta_{max} = \max_{\Omega_e \subset \Omega} \eta_{\Omega_e}$ in the following way

$$\begin{array}{lll} \text{if} & \eta_{\Omega_e} \leq \theta_{crs}\eta_{max} & \rightarrow \text{coarsen element } \Omega_e \\ \text{if} & \eta_{\Omega_e} \geq \theta_{ref}\eta_{max} & \rightarrow \text{refine element } \Omega_e \end{array} \tag{3.9}$$

The values of the modified nodes are then interpolated or restricted after refinement or coarsening using the known Lagrange basis functions for each element. The practical implementation of mesh creation and manipulation is done with the library `amatos` as introduced in Behrens et al. (2005). It has the advantage that it uses an efficient space-filling curve-ordering of elements (see Behrens and Bader (2009)), which allows fast access of neighboring elements and thus reduction of cache misses.

3.3.3. Wetting and Drying

The modeling of wetting and drying with a nodal Lagrange model requires imposing additional conditions on the numerical solution such as minimizing artificial oscillations in order to retain non-negativity of the fluid height as well as preservation of steady states.

The cells communicate through boundary integrals as in (3.4). The numerical flux, $\mathbf{F}^*$, will be computed using a consistent, monotone, Lipschitz-continuous and conservative Riemann solver such as the ones described in Toro (2001). Our observation is that both, the approximate and exact solvers for shallow water equations lead to comparable results (see also section 4.4.

In general, oscillations can be minimized through the use of filters, slope limiters or artificial viscosity. Numerous limiting techniques are available in present literature. Examples are the edge-based limiter developed in Cockburn and Shu (1998), Xing et al. (2010) and Xing and Zhang (2013) that mainly relies on linear scaling around cell averages and can be shown to formally preserve both, accuracy and positivity of the fluid height. Another vertex-based limiter has been established in Kuzmin (2010, 2013b). The latter depends on inner-element Taylor expansions and in Aizinger (2011) it could be shown that the limited solution is minimizing a local optimization problem, which, in an L_2-sense makes the modification optimal. The choice of a limiter is critical and in combination with our model the vertex-based limiter showed the best performance

among the stated ones (for detail, please consult sections 4.3.1 and 4.3.3) with respect to robustness and accuracy. We will use this limiter throughout the chapter.

Preservation of steady states, or well-balancing, is another important and non-trivial aspect of a nodal DG model. Our approach uses the strong form (3.4), which, on a discrete level, is balanced and takes into account that on dry cells with absent momentum the balance of pressure gradients has automatically to be fulfilled. So, we only compute the gradients for the wet part of the element and for the interpolation, we assume that in case of small velocities, there is a perfect horizontal water surface approaching the coast. A detailed presentation of this approach is given in section 4.5.

Finally, we shortly comment on the source term from (3.2). In our one-layer model, the wind stress will be modeled as a movement of water columns and for bottom friction a Manning-type law is assumed. So, the source term entries look as follows.

$$\boldsymbol{\tau}_s = \frac{\gamma_\tau \boldsymbol{\tau}}{\phi\rho}, \qquad \boldsymbol{\tau}_b = g\ n_M^2\ \phi\mathbf{u}\frac{\sqrt{\|\mathbf{u}\|_2}}{\phi^{4/3}}, \qquad \mathbf{f}_c = 2\Omega_{rot}\sin(\psi_L)\begin{pmatrix}\phi v\\ -\phi u\end{pmatrix} \tag{3.10}$$

with $\boldsymbol{\tau}$ a prescribed wind vector field, ρ the density of the fluid, $\gamma_\tau \in \mathbb{R}$ the wind friction coefficient, n_M a dimensionless number (Manning's n), Ω_{rot} the rotation rate of the earth and ψ_L the latitude. Details on the discretization of the source terms and an extension for the computation of $\boldsymbol{\tau}_s$ from a given storm can be found in section 6.3.

3.4. Numerical Tests

In this section we show results of test cases that demonstrate major functionalities of our model: handling wetting and drying, wind stress and wellbalancing. The test suite comprises both, tests with and without analytical solutions. We use the test cases introduced in Balzano (1998) and Thacker (1981) to demonstrate the capability of modeling wetting and drying accurately. The wellbalancing test is a classical partially dry lake at rest and the implementation of the wind stress will be tested with a circulating wind set up. Throughout this section, we employ second order, i.e. linear, basis functions for the simulation as higher than second order basis functions are not straightforwardly usable. We developed techniques, however, that enable the use of higher order basis function which can be found in section 5.2.

The tests are performed using different refinement criteria which will be stated in each case. The results of the tests are supposed to show how dynamic adaptivity performs for cases in which wetting and drying and wind stress is present. If no analytical solution is available, we will compare the results of adaptive simulations with uniform numerical approximations on a very fine mesh. Therefore, we define the error measures

$$\mathcal{E}_2(f_h) = \frac{\|f_{fine} - f_{crs}\|_2}{\|f_{fine}\|_2}, \qquad \mathcal{E}_\infty(f_h) = \frac{\|f_{fine} - f_{crs}\|_\infty}{\|f_{fine}\|_\infty} \qquad \text{for } f_h \in \mathcal{P}_n \tag{3.11}$$

to assess the loss of accuracy that occurs due to the usage of the adaptive mesh. With $\|\cdot\|_2$ we denote the standard L_2-Norm and with $\|\cdot\|_\infty = \max_\Omega|\cdot|$. Here, f_{fine} is to be understood as the discrete numerical solution f_h, that was computed on a fine uniform

mesh and f_{crs} is the coarser numerical solution from a non-uniform, adaptive test run that has been interpolated onto the uniform points of the fine mesh using the local basis functions.

For the testcases with analytical solutions, we use

$$\begin{aligned} E^{\infty}_{abs}(f_h) &= \|f_h - f\|_{\infty}, \qquad & E^{2}_{abs}(f_h) &= \|f_h - f\|_{2}, \\ E^{\infty}_{rel}(f_h) &= \frac{\|f_h - f\|_{\infty}}{\|f\|_{\infty}}, \qquad & E^{2}_{rel}(f_h) &= \frac{\|f_h - f\|_{2}}{\|f\|_{2}} \end{aligned} \tag{3.12}$$

where f is the analytical and f_h the numerical solution.

The adaptive simulations will involve different spatial resolutions. With $\Delta\mathbf{x}_{min}$ we will denote the shortest and with $\Delta\mathbf{x}_{max}$ the longest edge within the domain. Through the tolerances θ_{ref} and θ_{crs} we control the amount of cells to be refined and/or coarsened in each timestep as proposed in (3.9). The larger θ_{ref}, the relatively fewer cells will be refined and the larger θ_{crs}, the more cells will be coarsened and vice versa. For comparison of how much computational time is saved, we state a CPU time normalized to the CPU time of the uniformly fine run. We remark that all the non-normalized quantities are dimensioned and, if not stated otherwise, we assume SI units.

One of the major aspects of this paper is the reduction of computational complexity, which we understand as the amount of computing resources that are needed to perform a particular task. Our particular task here is the computation of a numerical solution of a certain precision and the computing resources are measured with CPU time. This is significantly different from stating that reducing the number of degrees of freedom leads to an equal reduction in computing time, because the term computational complexity also depends on a fixed goal that is identified with a fixed accuracy of the numerical solution in our case.

3.4.1. Testcases with Analytical Solutions

Testcase 1. *Partially dry lake at rest*
We assume a square domain $\Omega = [0, L]^2$ for $L = 30$, $\alpha = 0.25$, $\beta = 0.5$, bathymetry

$$b(\mathbf{x}) = \max\left(0, \alpha - 5\,\left((x-\beta)^2 + (y-\beta)^2\right)\right)$$

and initial conditions:

$$\begin{aligned} h(\mathbf{x}, t) &= \max\left(0, 0.2 - \max\left(0, \alpha - 5\,\left((x-\beta)^2 + (y-\beta)^2\right)\right)\right) \\ u(\mathbf{x}, t) &= \mathbf{0} \qquad \text{for all } t \in [0, T]. \end{aligned}$$

This test-case is chosen in order to test the conservation of mass and momentum in a steady state. Here, we are mainly concerned to check the compatibility of the discretization of the flux tensor and source term. We ran all the tests with a Rusanov Riemann solver and the explicit Runge Kutta scheme RK22 with a time-step of

			E^2_{rel}	E^∞_{rel}	E^2_{abs}	E^∞_{abs}	CPU
$\Delta\mathbf{x}_{min}$	$\Delta\mathbf{x}_{max}$	# elmt	ϕ		$\phi\mathbf{u}$		time
1.326	1.326	1024	5.7477E-13	3.2250E-12	3.2164E-15	1.3055E-14	1.0000
1.326	1.875	528	5.8306E-13	1.0336E-12	1.2348E-14	4.2208E-14	0.5034
1.326	2.652	296	5.4989E-13	9.6542E-13	1.1106E-14	3.6360E-14	0.2722
1.326	7.500	152	5.3813E-13	9.3359E-13	1.1932E-14	2.7532E-14	0.1873

Table 3.1.: Partially dry lake at rest. Errors at t=40s (RK22, $\Delta t = 0.001$) - with adaptation parameters $\theta_{crs} = 0.1, \theta_{ref} = 0.9$ for different resolutions.

$\Delta t \in \{0.001, 0.002\}$ depending on the spatial resolution. For limiting purposes, we used the nodal limiter. We observed that neither a higher order Runge Kutta scheme nor a more sophisticated Riemann solver led to significant improvement of the errors for this problem which might be due to the small overall magnitude of the errors for this test in general. The results are depicted in table 3.1 and show the relative errors for the fluid height and the absolute errors for the momentum that were defined in (3.12) at time $t = 40s$. The reason why we use two different error measures is, that the analytical solution for the velocity is zero. In all, the deviations from the steady state solution are nearly within the range of machine precision. As a refinement criterion for the adaptive test runs, we used the divergence of the fluid height: $\eta_{\Omega_e} = \nabla \cdot \phi$ and observe that, while the error measures are not affected by the use of an adaptive mesh, the computational costs were significantly reduced. Moreover, the modification of the flux tensor that we introduced for wet/dry interfaces does not affect the accuracy or robustness of the computations. However, probably due to damping effects and numerical diffusion, there seems to be a slight improvement in the errors for some of the adaptive test runs which contradicts our convergence study that is presented in chapter 4 for partially dry problems.

Testcase 2. Paraboloid 2D basin
We consider a domain: $\Omega = [-L, L]^2$ with parameters: $D_0 = 50, L = 430620, \eta_0 = 2$, reflecting boundaries, Coriolis forcing and

$$
\begin{aligned}
A &= \frac{(D_0+\eta_0)^2 - D_0^2}{(D_0+\eta_0)^2 + D_0^2}, \qquad D(\mathbf{x}) = D_0\left(1 - \frac{x^2+y^2}{L^2}\right) \\
\eta(\mathbf{x}) &= D_0\left(\frac{\sqrt{1-A}}{1-A} - 1 - \frac{x^2+y^2}{L^2}\left(\frac{1-A^2}{(1-A)^2} - 1\right)\right) \\
h(\mathbf{x},0) &= \begin{cases} D(\mathbf{x}) + \eta(\mathbf{x}) & \text{if } D(\mathbf{x}) + \eta(\mathbf{x}) \geq 0 \\ 0 & \text{otherwise.} \end{cases} \\
\mathbf{u}(\mathbf{x},0) &= \mathbf{0} \\
b(\mathbf{x}) &= D_0 \frac{x^2+y^2}{L^2}
\end{aligned}
$$

The analytical solution for this problem can be found in Thacker (1981).

$\Delta\mathbf{x}_{min}$	$\Delta\mathbf{x}_{max}$	θ_{ref}	θ_{crs}	# elmt	$E^2_{rel}(\phi)$	$E^\infty_{rel}(\phi)$	CPU time
35988.95	50896.06	0.0	0.0	2048	4.0348E-11	0.0312E+00	1.0000
71977.90	101792.13	0.0	0.0	512	1.1656E-10	0.0559E+00	0.0525
35988.95	143955,81	0.9	0.1	936	8.5180E-11	0.0435E+00	0.4846
50896.06	50896.06	0.0	0.0	1024	5.8590E-11	0.0396E+00	0.1994
35988.95	101792.13	0.9	0.1	1040	5.0650E-11	0.0375E+00	0.5458

Table 3.2.: Paraboloid 2D basin. Relative errors at $t = 30\ 000s$ (RK35, $\Delta t \in [0.01, 0.05]$) for different resolutions.

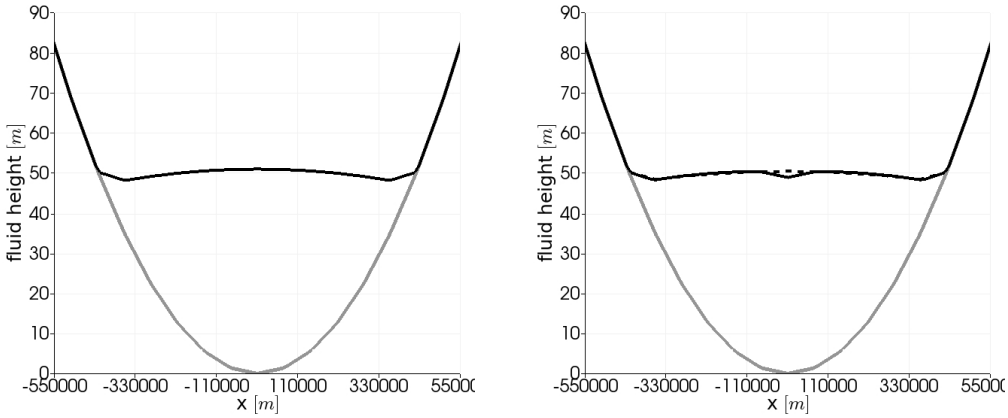

Figure 3.1.: Paraboloid 2D basin. Plot of linear cross section $x = y$ after $t = 0s$ (left) and $t = 30\ 000s$ (right). Depicted are the numerical (solid line) and the analytical solution (dotted line).

Figure 3.1 shows the analytical solution and our numerical results on a uniform mesh for a linear cross-section of the domain with $x = y$ at time $t = 0s$ and $t = 30\ 000s$. With the employed limiter from Kuzmin (2010) the deviations are small. This test has been carried out using a third-order Runge-Kutta scheme (RK35) and a time-step of $\Delta t \in [0.01, 0.05]$ depending on the spatial resolution. The nodal limiter led to satisfying results at the wet/dry interface; even after 3 million time-steps no significant effect of diffusion was to be seen as is shown in figure 3.1. The apparent error in the center of the domain is due to the slope limiter. As we used linear basis functions local extrema are not approximated well, so that in the center the maximum of the fluid height is cut off. As a refinement indicator we used

$$\eta_{\Omega_e} = \begin{cases} \frac{1}{\overline{\phi_{e,h}}} & \text{for } \overline{\phi_{e,h}} \geq 10^{-8} \\ 0 & \text{otherwise,} \end{cases}, \tag{3.13}$$

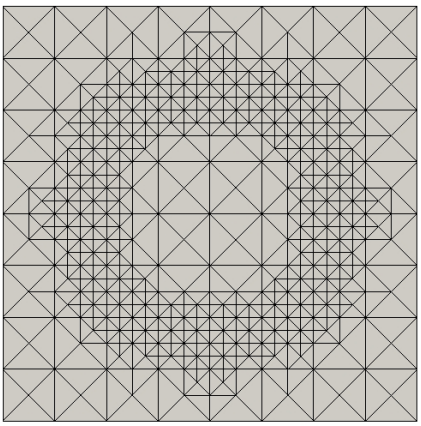

Figure 3.2.: Paraboloid 2D basin. Example of an adaptive mesh.

where $\phi_{e,h}$ is the numerical solution of the fluid height in element Ω_e and the over-lined quantity the element mean value. The indicator leads to a refinement in areas that are close to dry or wet. A picture of an adaptively refined mesh can be found in figure 3.2. The regularity of the mesh originates from the user-defined initial triangulation. As an initial triangulation we chose a quadrilateral consisting of two right triangles. The errors obtained from the uniform and adaptive runs can be acquired from table 3.2. The last two lines show that on the one hand, a uniformly refined coarser mesh leads to a reduction of computational time since a larger time-step can be chosen. But on the other hand if you take the approximately same number of elements that are non-uniformly refined, the relative errors can be reduced.

3.4.2. Testcases without Analytical Solutions

The tests presented in this subsection do not permit an analytical solution. For comparisons, we always used a numerical solution on a uniform fine mesh and the error measures defined in (3.11).

Testcase 3. *Tidal beach*
The model domain $\Omega = [0, L] \times [0, W]$ is of length $L = 13800$ and width $W = 3450$, Manning's $n_M = 0.02$, $\mathbf{f}_c = \boldsymbol{\tau}_s = \mathbf{0}$ and we have parameters $\alpha = \frac{5}{13800}$, $c_1 = \frac{60}{23}, c_2 = \frac{100}{23}$,

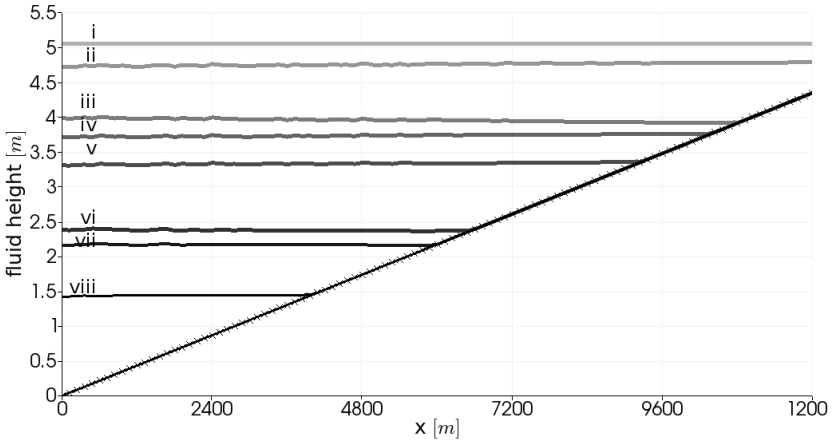

Figure 3.3.: Tidal beach. Uniform simulation: Uniformly sloping bathymetry b_1. Bathymetry (crossed line) and total water height (solid lines) every $2500s$ over the linear crosssection with $y = 1725$. The larger the number of time-steps, the darker and thinner the line and the higher the roman numbering

and initial conditions

$$
\begin{aligned}
h(\mathbf{x}, 0) &= 5 - \alpha x \\
\mathbf{u}(\mathbf{x}, 0) &= 0 \\
b_1(\mathbf{x}) &= \alpha x \\
b_2(\mathbf{x}) &= \begin{cases} \alpha x & \text{for } x \leq 3600 \text{ or } x \geq 6000 \\ -\alpha x + c_1 & \text{for } 3600 \leq x \leq 4800 \\ \frac{x}{920} - c_2 & \text{for } 4800 \leq x \leq 6000 \end{cases}
\end{aligned}
$$

The boundary conditions are reflecting on the two opposite walls at the top and bottom of the domain as well as on the right wall. On the remaining left wall we prescribe periodic tidal boundary conditions of the kind

$$h(\widetilde{\mathbf{x}}, t) = h_{bou}(t) := 5 - 2\sin\left(\frac{2\pi t}{43200}\right)^2$$

with $\widetilde{\mathbf{x}} \in \{\mathbf{x} : x = 0 \wedge y \in [0, W]\}$ a point on the tidal boundary.

Figures 3.3 and 3.4 show our results on a uniform mesh for the linear cross-section of the domain with $y = 1725$ over time for both bathymetries b_1 and b_2. Depicted are the point-values every $2500s$. To improve the readability of the plots, the lines are numerated using roman numbers (k) to refer to the corresponding simulation time $t = (k-1)\cdot 2500s$, i.e. the line indicated with (i) is the initial condition. We observe that the tidal forcing leads to retreating water on the right of the domain and formerly wet areas run dry

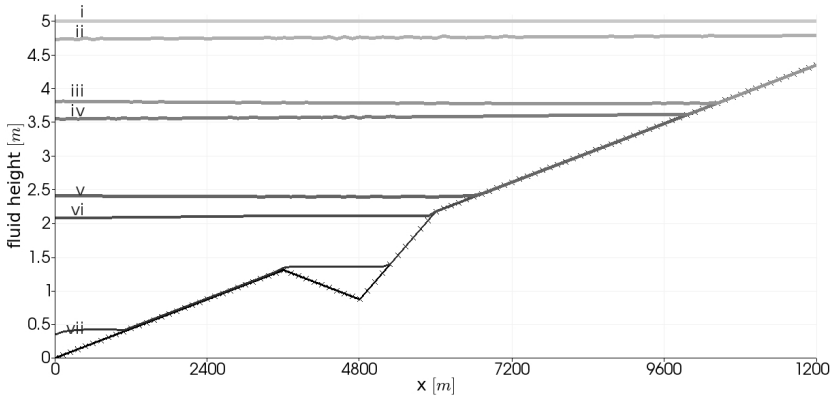

Figure 3.4.: Tidal beach. Uniform simulation: Sloping beach with reservoir and bathymetry b_2. Bathymetry (crossed line) and total water height (solid lines) every $2500s$ over the linear crosssection with $y = 1725$. The larger the number of time-steps, the darker and thinner the line and the higher the roman numbering.

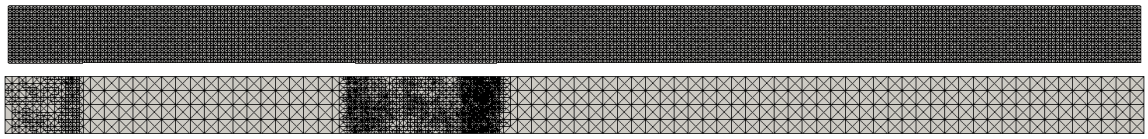

Figure 3.5.: Tidal beach. Meshes corresponding to the right figure 3.4 for the uniform simulation (top) and an adaptive simulation (bottom) at time $t = 30\,000s$.

$\Delta\mathbf{x}_{min}$	$\Delta\mathbf{x}_{max}$	θ_{ref}	θ_{crs}	#elmt	$\mathcal{E}_2(\phi)$	$\mathcal{E}_\infty(\phi)$	CPU time
26.517	53.033	0.9	0.1	60288	0.1550E-02	0.7397E-01	0.8150
53.033	150.000	0.9	0.1	12576	0.3153E-02	0.9194E-01	0.2202
75.000	106.066	0.0	0.0	10240	0.3660E-01	0.5284E-01	0.1196
53.033	212.132	0.8	0.4	2656	0.1398E-01	0.8968E-01	0.0495

Table 3.3.: Tidal beach. Relative errors at $t = 30\,000s$ (RK22, $\Delta t = 0.25s$) for different resolutions. As a uniform fine grid we took one with $\Delta\mathbf{x}_{min} = 26.52$ for comparison.

$\Delta\mathbf{x}_{min}$	$\Delta\mathbf{x}_{max}$	θ_{ref}	θ_{crs}	#elmt	$\mathcal{E}_2(\phi)$	$\mathcal{E}_\infty(\phi)$	CPU time
26.517	53.033	0.9	0.1	44344	0.1091E-01	0.2606E-00	0.3887
37.500	75.000	0.9	0.1	25204	0.1114E-01	0.2978E-00	0.2028
53.033	150.000	0.9	0.1	5296	0.3669E-01	0.2061E-00	0.0526

Table 3.4.: Tidal beach with Reservoir. Relative errors at $t = 50\,000s$ (RK22, $\Delta t = 0.25s$) for different resolutions. As a uniform fine grid we took one with $\Delta\mathbf{x}_{min} = 26.52$ for comparison

without introducing significant numerical oscillations. For the computations, we used an explicit Runge-Kutta scheme of second order (RK22) with a timestep of $\Delta t = 0.5$ in order to fulfill the CFL restrictions from (3.6) and the limiter from Kuzmin (2010). The numerical flux had been computed with the rather diffusive Rusanov Riemann solver and we did not take into account any additional numerical viscosity. For the adaptive simulations we chose the refinement indicator from (3.13), that leads to refinement in nearly dry areas, since the observed flow is almost laminar in most areas and we therefore assumed that the errors are more likely to occur in regions where a state change from wet to dry is going to take place. An adaptive grid for the tidal beach with bathymetry b_2 is given in figure 3.5 (right). Comparisons between several dynamically adaptive simulations with a uniformly fine one are given in tables 3.3 - 3.4. In both cases we observe that the use of an adaptive mesh leads to a significant reduction of computational costs, but in return does not significantly affect the accuracy. The observed relative L_2 errors are still within a range of $\mathcal{O}(10^{-2})$ and $\mathcal{O}(10^{-1})$ for the uniformly sloping bathymetry and for the test case with the reservoir respectively.

To further test the robustness of the implementation, we simulated a so called cold start of the two test cases, i.e. with an initial water height of $h(\mathbf{x}, 0) = 0$ and a modified ingoing tide of the form

$$h(\widetilde{\mathbf{x}}, t) = h_{bou}(t) := h_R(\widetilde{\mathbf{x}}, t) + 2\sin\left(\frac{2\pi t}{43200}\right)^2$$

with $\widetilde{\mathbf{x}} \in \{\mathbf{x} : x = 0 \wedge y \in [0, W]\}$ a point on the tidal boundary and $h_R(\widetilde{\mathbf{x}}, t)$ its neighbor over the boundary edge corresponding to the element that fully belongs to the domain.

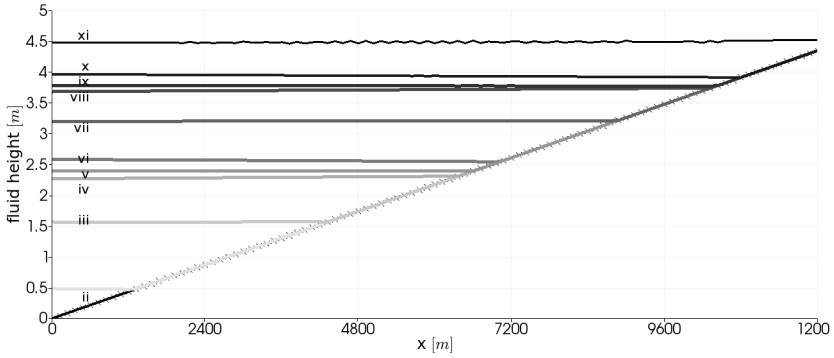

Figure 3.6.: Tidal beach with ingoing tide. Uniform simulation: Uniformly sloping bathymetry b_1. Bathymetry (crossed line) and total water height (solid lines) every $2500s$ over the linear crosssection with $y = 1725$. The larger the number of time-steps, the darker and thinner the line and the higher the roman numbering

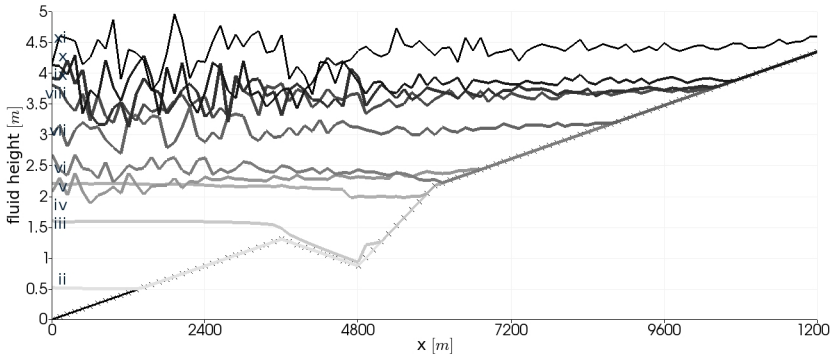

Figure 3.7.: Tidal beach with ingoing tide. Uniform simulation: Sloping beach with reservoir and bathymetry b_2. Bathymetry (crossed line) and total water height (solid lines) every $2500s$ over the linear crosssection with $y = 1725$. The larger the number of time-steps, the darker and thinner the line and the higher the roman numbering.

The results are depicted in figures 3.6 and 3.7. Again, the roman numbering indicates the simulation time and we observe, that only slight wiggles are introduced on the surface of the fluid for the case with the uniform sloping bathymetry b_1. The plot corresponding to bathymetry b_2 develops oscillations after $10000s$ which seems to be due to the reflection of the ingoing wave from the reservoir. This reflection produces a wave in the contrary direction to the tide which then produces oscillations.

Testcase 4. *Circulating wind*
We consider the shallow water equations in a square domain $\Omega = [0, 30]^2$ with linear bathymetry $b(\mathbf{x}) = 60 - (x + y)$, reflecting boundaries and initial conditions

$$\begin{cases} h(\mathbf{x}, 0) = & x + y + 80 \\ \mathbf{u}(\mathbf{x}, 0) = & \mathbf{0} \\ \boldsymbol{\tau}(\mathbf{x}, t) = & (-10, 10)^\top \quad \text{for all } t \in [0, T] \end{cases}$$

and wind blowing constantly diagonal to the bathymetry with parameters $n = 2/g$, where $g = 9.81$ is the gravitational constant and $\gamma_\tau = 10$.

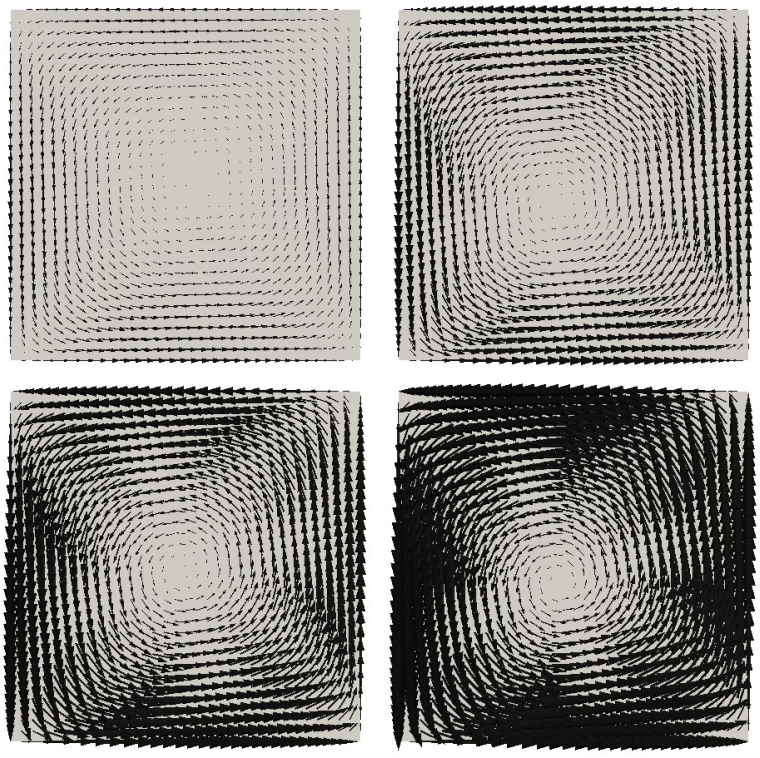

Figure 3.8.: Circulating wind. Velocities $\mathbf{u}$ every $20s$ from upper left to bottom right. Intensity of vectors indicates magnitude of $\mathbf{u}$.

In figure 6.1 we plot the computed velocity vectors every 4000 time-steps (= $20s$). The length and intensity of the vectors correspond to the magnitude of the velocity. The longer the simulation runs, the more pronounced the circular movement of the water

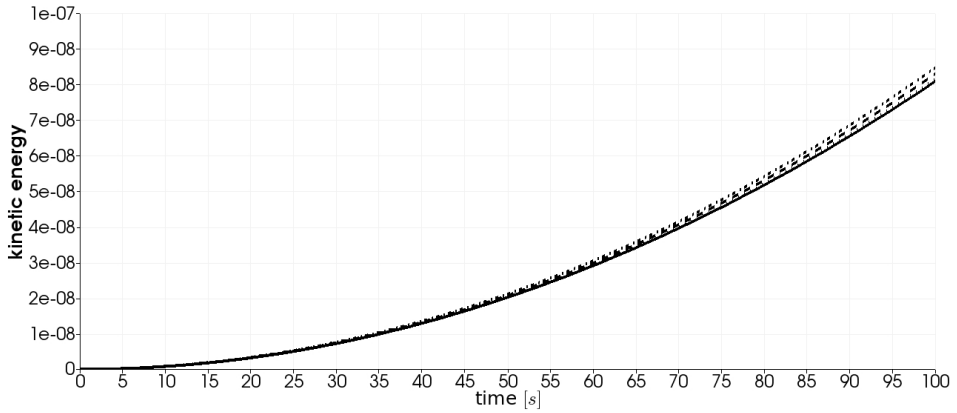

Figure 3.9.: Circulating wind. Kinetic energy over time for the uniform simulation (solid line) and several adaptive (dashed lines). The minimal uniform resolution is $\Delta x_{min} = 1.326$. The adaptive solutions show a converging behavior with $\Delta x_{min} = 1.326$ for all testruns and $\Delta x_{max} \in \{3.75, 5.303, 10.607\}$

is. The motion does not start right from the beginning though. It takes approximately 2000 time-steps for the system to develop the circular motion. The computations were performed using a Runge Kutta scheme of order two (RK22) with a time-step $\Delta t = 0.0025$ in combination with the Rusanov Riemann solver. For the uniform simulation, the spatial resolution was $\Delta x = 1.875$ and the total amount of elements was 512. Throughout the computations, the bathymetry as well as the fluid height are linear (for $t = 0$) and remain mostly homogeneous. Therefore, we identified the vorticity, $\eta_{\Omega_e} = \xi_e := (\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y})$ as a suitable refinement criterion. For comparison of adaptive and the uniform simulation, we computed the kinetic energy over time which is defined as $\int_\Omega \left(u(\mathbf{x})^2 + v(\mathbf{x})^2 \right) d\mathbf{x}$. Figure 3.9 shows that the kinetic energy from the adaptive simulations does not significantly deviate from the one of the uniform simulation.

3.5. Concluding Remarks

In this chapter we have presented a two-dimensional DG shallow water model in combination with a dynamically adaptive triangular mesh aiming for twofold: good agreement with wet/dry interfaces as well as the reduction of computational costs. The mesh had been locally adapted according to heuristic user-defined refinement indicators and tolerances as shown in (3.9). Slope limiters are employed for modeling wetting and drying with the purpose of reducing spurious oscillations and maintaining positivity of the fluid height. Moreover, in case of a resting water body, a correction of the pressure gradients in the right-hand side had been carried out for partially dry cells in order to preserve still water states exactly.

We have confirmed that the presented nodal DG method yields robust results when being applied to dynamically adaptive meshes. With the discussed modification of the pressure gradients in the strong flux form in (3.4), we were able to preserve steady

states and obtain errors that were in the range of machine precision. For the nodal approach that we had chosen in this paper, the limiter presented in Kuzmin (2010), which works with local Taylor expansions, yielded the best results in terms of positivity preservation, accuracy and robustness. The computational overhead that emerged from the use of the limiter was of quadratic order in the degrees of freedom. For the test cases, that we have shown, the use of adaptivity can reduce the computational costs while not affecting the accuracy significantly. For example, in the test cases that were taken from Balzano (1998), we could reduce the computational costs by up to 80% if we accepted a loss of precision of the order of 1%. Here, we have to stress though, that the high-number for the cost reduction also comes from the fact that these tests are quasi one-dimensional. In the steady state preserving case, we even achieved higher accuracy with the introduction of an adaptive mesh. However, this is probably a consequence of a damping of artificial waves that travel from the finer part of the mesh to the coarser.

In the present chapter, the implementation of the wind stress was done using a prescribed wind vector field. For a wider class of applications, it seems sensible to work with the storm model presented in Holland (1980). The latter as well as the performance of the model when being applied to real data will be discussed in detail in chapter 6.

4. Approximations in a Discontinuous Galerkin Inundation Model

4.1. Abstract

Discontinuous Galerkin (DG) models for inundation simulations comprise several numerical approximations. Examples are the Riemann solvers, that are used to compute fluxes over element boundaries, and slope limiters that reduce numerical oscillations. These on the one hand introduce limitations but on the other hand ensure that stable and accurate solutions are obtained. We study their effects on the accuracy and computational efficiency of the numerical solution and, in this respect, propose a computationally inexpensive technique for well-balancedness.

4.2. Introduction

The numerical solution of a set of partial differential equations with a DG method requires numerical approximations in several parts of the model. These serve multiple purposes: Stabilization of the numerics, balancing of the discretization and reduction of numerical errors. However, it is important that these modifications do not violate desirable properties of the computed solution such as boundedness of the total variation, at least in the mean, in order to theoretically obtain convergence and satisfy entropy conditions.

The occurrence of spurious oscillations is one challenge of shallow water modeling. From Godunov (1959) we know that higher order numerical methods for non-linear problems tend to produce oscillations especially in the vicinity of steep gradients. These can lead to negative water heights which then influence the stability of the model for two reasons. Computations of the largest possible timestep Δt require the computation of the maximal eigenvalue $\lambda_{max} = \langle \mathbf{u}, \mathbf{n} \rangle + \sqrt{\phi}$ which is only well defined for non-negative water heights. Moreover, the determination of the velocity from momentum, $\mathbf{u} = \frac{\phi \mathbf{u}}{\phi}$, is bad conditioned for small ϕ and a division by zero has, in any case, to be avoided.

Dry states also lead to two collapsing eigenvalues, as we can see from (2.5), so that the system is not longer hyperbolic as in section 2.1. The loss of hyperbolicity is known to lead to instabilities of the simulation and a loss of wellposedness (see e.g. Xing and Zhang (2013)).

Possibilities to reduce oscillations can be found in Hesthaven and Warburton (2008). Low-pass filters are one option because they remove high-frequency parts of the numerical solution which stabilizes the computation and smooths the result in a post-processing routine. The same effect can be achieved by adding artificial viscosity to the equations.

This requires the computation of a discrete Laplace operator Δ, which on unstructured grids with piecewise linear basis functions is a non-trivial task. Yet another method is slope limiting. Slope limiters are sophisticated post-processing filters that are only active in dynamically determined rough parts of the solution. In this work, we will exclusively focus on them, because they only affect the solution in critical regions and often take into account physics-based criteria.

The communication between the cells, or the fluxes over the edges, contains a further approximation, that takes into account the dynamics of the system. This edge problem is given by a Riemann problem as in section 2.1, which can be solved with any exact or approximate Riemann solver.

The balance between the pressure gradients, i.e. the gradients of the surface height h and the bathymetry b, on a discrete level is also of importance as it severely influences the creation of artificial waves at bathymetry gradients when the system is at rest and therefore has influence on the preservation of steady states.

All these approximations and their effect on the accuracy of the method are discussed in this chapter. In 4.3.1 we introduce a general characterization of a large class of slope limiters and show how commonly used ones fit into this context. Special emphasis will be put on a theoretical and numerical comparison. Finally, we study Riemann solvers and their influence on flooding and drying in section 4.4 and show how we achieve preservation of states at rest using the strong form of the integral equations in 4.5.

4.3. Slope Limiters

One of the main challenges of inundation modeling is to accurately compute interfaces between wet and dry cells. As we have seen in section 4.2, flooding and drying simulations with shallow water equations can develop spurious oscillations. To reduce them we employ slope limiters. We remark though that limiters are needed because of the hyperbolic nature of the problem. Problems that involve diffusive fluxes or other viscosity terms do not require any limiting according to Cockburn et al. (1999).

In current literature, we find different approaches for the reduction of numerical oscillations. Most of them are concerned with the development of slope limiters. In general, we distinguish between methods employing a moving and a static boundary. The moving boundary requires the definition of boundary conditions as was done in Lynch and Gray (1978) and in Leclerc et al. (1990) for flooding and drying scenarios with a finite element model. On the one hand moving boundaries restrict the computations to the wet domain, on the other hand they require a computation the boundary of zero water height dynamically in every timestep.

Static boundaries in turn require to perform the computations in the whole domain and to decide on which element is wet or dry with the help of element deactivation. For the dry domain one can then apply a thin layer approach, i.e. the actual dry regions are artificially wetted with a small amount of water, so that no special care has to be taken about wet/dry interfaces. Most of the existing models employ this approach, though it is possible to allow for negative water heights as presented in Heniche et al. (2000).

The limiter presented in Bunya et al. (2009) is a combination of a positive-depth operator, therefore belonging to the class of thin layer approaches, and a conventional limiter as in Cockburn and Shu (1998). However, the modification of the water height to ensure positive depth at any time lead to a scheme whichs wellbalancedness depends on a free parameter H_0, because for mean water depths below this threshold H_0, the nodal values of the limited solution are set to the mean water depth without modifying the bathymetry. A similar approach is performed for the models in Chen et al. (2008) and Gopalakrishnan and Giraldo (2014). In addition thereto in the model of Gopalakrishnan and Giraldo (2014) a simple minmod limiter as presented in LeVeque (2002) is employed to control the oscillations.

The total variation of the numerical solution gives information about the existence and severity of oscillations. If the total variation is bounded, we call the method total-variation-bounded, or TVB. The boundedness is essential for the minimization of unphysical oscillations and the proof of convergence in the mean depends on the boundedness of the total variation in the mean. The limiter presented in Xing et al. (2010) for a well-balanced scheme, that has been presented in a computationally improved way in Zhang and Shu (2011), has this property. The main principle behind it is that in case of oscillations a linear scaling around the cell-average is applied. This does not only lead to a reduction of oscillations but also ensures positivity of the sea surface height and maintains the wellbalancedness of the scheme.

Yet another approach had been developed in Kärnä et al. (2011) that includes a transformation of the bathymetry such that with the modified bathymetry, the whole domain is wet and no special treatment has to be developed for dry cells.

Finally, especially in the case of operational models that are in use, estimators for the inundation run-up are used as is done for example with a one-dimensional Boussinesq model in Kennedy et al. (2012) for the model developed in Westerink et al. (2008).

In our model, we will focus on a special class of limiters that work on partially dry domains. Their performance will be the subject of the next section.

4.3.1. General Characterization

Slope limiting routines can be considered as advanced post-processing filters. In our explicit s-stage strong stability preserving Runge Kutta time-stepping scheme, the limiter will take effect during every intermediate step of the evolution process in order to suppress oscillations and negative fluid heights in every partial step by modifying the right-hand side $\mathbf{H}(\mathbf{U})$ of the ordinary differential equation (3.7) in the following way:

$$\begin{aligned}
\mathbf{U}_h^{(0)} &= \mathbf{U}_h^n \\
\mathbf{U}_h^{(i)} &= \widetilde{\mathbf{H}}\left(\sum_{l=0}^{i-1}\left(\alpha_{il}\mathbf{U}_h^{(l)} + \beta_{il}\Delta t\mathbf{H}\big(\mathbf{U}_h^{(l)}\big)\right)\right) \qquad \text{for } i = 1\ldots s \\
\mathbf{U}_h^{n+1} &= \mathbf{U}_h^{(s)}
\end{aligned}$$

where α_{il}, β_{il} are the SSP Runge Kutta coefficients and $\widetilde{\mathbf{H}}$ is the slope limiter. In principle, limiters can also be employed in other types of time-stepping schemes. In the recent paper by Kuzmin (2013a), slope limiting techniques are applied to a semi-implicit scheme. We will, however, only focus on explicit schemes for two reasons. First, explicit time-stepping schemes are easier to implement and parallelize. Second, with the application we have in mind, we do want to simulate the fast gravity waves which a (semi-)implicit scheme would possibly damp in order to achieve a larger possible timestep.

All the slope limiters that we consider in this work are post-processing filters of the general form

$$\widetilde{\mathbf{H}}(f_h) = \sum_{e\in\mathcal{E}} \chi_e^{\widetilde{\mathbf{H}}} \Big(\overline{f}_{e,h} + \sum_{k=2}^{m_e} \ell_k \mathbf{L}_k(f_{e,h}) \Big) + \big(1 - \chi_e^{\widetilde{\mathbf{H}}}\big) f_{e,h} \tag{4.1}$$

with a switch

$$\chi_e^{\widetilde{\mathbf{H}}} = \begin{cases} 1 & f_h \text{ on } \Omega_e \text{ has to be limited,} \\ 0 & \text{otherwise.} \end{cases},$$

$\overline{f}_{e,h}$ is the cell average of the discrete numerical solution f_h restricted to the element Ω_e, $m_e \in \mathbb{R}$, $\ell_k \in \mathbb{R}, k = 2, \ldots, m_e$ are parameters that are limiter-specific and $\mathbf{L}_k : \mathcal{P}_n(\Omega_e) \to \mathcal{P}_n(\Omega_e)$ are also limiter-specific operators that modify the numerical solution within the local space of ansatz functions which in our case is a polynomial space of order n.

Numerically, we compute the cell averages using an interpolatory Gauss Lobatto quadrature of the form

$$\overline{f}_{e,h} = \frac{1}{\mathrm{vol}(\Omega_e)} \int_{\Omega_e} f_h(\mathbf{x}) d\mathbf{x} \approx \frac{1}{\mathrm{vol}(\Omega_e)} \sum_{i=1}^{m\prime} \boldsymbol{\omega}_i f_h(\boldsymbol{\xi}_i), \tag{4.2}$$

with Gauss quadrature points $\boldsymbol{\xi}_i$, weights $\boldsymbol{\omega}_i$ and $\mathrm{vol}(\Omega_e)$ the volume of the triangle Ω_e. The $m\prime$-point Gauss Lobatto rule on Ω_e is exact up to degree $2m\prime - 3$.

Following Cockburn (1998), we impose four requirements on slope limiters:

- Preservation of (high-order) accuracy away from critical points,
- Decrease of the gradient of the limited solution compared to the solution without limiting, i.e. $|\nabla \mathbf{U}_h| \geq |\nabla \tilde{\mathbf{H}}(\mathbf{U}_h)|$ pointwise in Ω,
- Boundedness of the limited solution , i.e. the existence of $\mathbf{U}_h$-dependent constants $c, C \in \mathbb{R}$ with $c \leq C$, so that $c \leq \widetilde{\mathbf{H}}(\mathbf{U}_h)(\mathbf{x}) \leq C$ for all $\mathbf{x} \in \Omega$, and
- Local conservation of mass and momentum i.e. $\int_{\Omega_e} \widetilde{\mathbf{H}}\big(\mathbf{U}_h\big) d\mathbf{x} = \int_{\Omega_e} \mathbf{U}_h d\mathbf{x}$ for all Ω_e. A sufficient condition therefore is $\int_{\Omega_e} \mathbf{L}_k(\mathbf{U}_h) d\mathbf{x} = 0$ for all $k = 2, \ldots, m_e$.

The requirements ensure, that $\widetilde{\mathbf{H}}$ does not affect the convergence of the numerical solution, which is strongly linked to the study of its total variation.

Total Variation

The main purpose of slope limiters is the reduction of numerical oscillations to compute meaningful solutions. Therefore, we introduce the total variation as a measure for the strength of oscillatory behavior. For a multivariate function $f \in L^1(\Omega)$ it is defined as:

$$TV(f)(t) := \sup \left\{ \int_\Omega f(\mathbf{x}, t) \mathrm{div} \varphi(\mathbf{x}) d\mathbf{x} : \varphi \in C_c^1(\Omega, \mathbb{R}^n), \|\varphi\|_\infty \leq 1 \right\},$$

where $\varphi \in C_c^1$ is a compactly supported and differentiable test function that is bounded in the L_∞-norm by 1. We remark, that this definition takes account of changes that occur inside the element as well as the potentially modified jumps over element boundaries. Ideally, we would wish for our limited numerical solution to fulfill a maximum principle of the form

$$\mathbf{U}_h(\mathbf{x}, t = 0) \in [m, M] \Rightarrow \mathbf{U}_h(\mathbf{x}, t) \in [m - \delta_M h^2, M + \delta_M h^2] \text{ for all } t > 0, \tag{4.3}$$

with $m, M \in \mathbb{R}$ and non-negative numbers δ_M and h. In Zhang and Shu (2011), high-order accuracy of the method is proven under the assumption (4.3). In order to not degrade the accuracy at local extrema, we can only achieve a total variation boundedness in the mean (TVBM):

$$\overline{\mathbf{U}_h}(\mathbf{x}, t = 0) \in [m, M] \Rightarrow \overline{\mathbf{U}_h}(\mathbf{x}, t) \in [m - \delta_M h^2, M + \delta_M h^2] \text{ for all } t > 0.$$

A proof in Cockburn (1998) shows, that under the assumption that the Runge Kutta (RK) coefficients α_{il} sum up to one, which is ensured for strong stability preserving RK methods, there will be at least a subsequence of the sequence of mean values that converges to a weak solution of the problem. For the rest of this section, we will focus on concrete examples for TVBM limiters.

The Simple Modal L_2 Limiter (L2)

The main principle is to redistribute the first mode (the mean value) of the numerical solution onto the degrees of freedom, which is computationally inexpensive as only precomputed matrices are involved. To illustrate the concept, we consider the global numerical nodal solution $\mathbf{U}_h$ at time $t = t_n$:

$$\mathbf{U}_h(\mathbf{x}, t_n) = \sum_{e \in \mathcal{E}} \Big(\sum_{k=1}^{m_e} \mathbf{U}_{e,h}^k(t_n) \psi_k(\mathbf{x}) \Big),$$

where $\mathbf{U}_{e,h}^k(t_n)$ is the kth nodal coefficient of the element Ω_e at time $t = t_n$ and ψ_k the nodal Lagrange basis functions. This solution is now projected into the elementwise modal space $\mathcal{P}_n = \text{span} \{\bigcup_{j \in \mathcal{J}_E} \mu_j\}$ that is spanned by orthogonal Jacobi polynomials. For simplicity we assume that $\mathcal{J}_e = \{1, \ldots, m_e\}$. Utilizing the elementwise Vandermonde matrix $\mathbf{V} := \mu_j(\mathbf{x}_i)_{1 \leq i,j \leq m}$, which contains the evaluation of the modal basis

functions at the Lagrange interpolation points, we obtain the equivalent representation

$$\mathbf{U}_h(\mathbf{x}, t_n) = \sum_{e \in \mathcal{E}} \Big(\sum_{k=1}^{m_e} \widetilde{\mathbf{U}_{e,h}^k}(t_n) \mu_k(\mathbf{x}) \Big).$$

Here, the tilded coefficients are the modal values for each element. Redistributing the first mode $\widetilde{\mathbf{U}_{e,h}^1}$ onto the nodal values of the respective element Ω_e gives the limited solution

$$\widetilde{\mathbf{H}}(\mathbf{U}_h)(\mathbf{x}, t_n) = \sum_{e \in \mathcal{E}} \chi_e^{\widetilde{\mathbf{H}}} \Big(\widetilde{\mathbf{U}_{e,h}^1}(t_n) \cdot \chi_e(\mathbf{x}) \Big) + \big(1 - \chi_e^{\widetilde{\mathbf{H}}}\big) \mathbf{U}_{e,h}(t_n),$$

with the indicator function χ_e for element Ω_e, i.e. $\chi_e(\mathbf{x}) = \{1 \text{ for } \mathbf{x} \in \Omega_e; 0 \text{ for } \mathbf{x} \notin \Omega_e$, that has the property that its integral over Ω_e equals the volume of the element and the general coefficients $\ell_k = 0$ for $k = 2, \dots m_e$ in (4.1).

This modification fulfills the requirements on limiters as stated in the previous section. The limiter preserves first order accuracy in the limited areas because it ignores higher-order information. It is mass and momentum conserving in the limited elements as we can see from

$$\int_\Omega \widetilde{\mathbf{H}}(\mathbf{U}_h)(\mathbf{x}, t_n) d\mathbf{x} = \sum_{e \in \mathcal{E}} \Big(\int_{\Omega_e} \widetilde{\mathbf{U}_{e,h}^1}(t_n) \cdot \chi_e(\mathbf{x}) \Big) d\mathbf{x} = \int_\Omega \mathbf{U}_h(\mathbf{x}, t_n) d\mathbf{x}.$$

where we assumed $\chi_e^{\widetilde{\mathbf{H}}} = 1$, because if an element is not limited, the mass is not changed. Furthermore, it leads to at most first order accuracy in regions where the limiting takes place, with the advantage, that mass is conserved and oscillations are completely suppressed. Thus, we obtain element- and componentwise $|\nabla(\widetilde{\mathbf{H}}(\mathbf{U}_h))| = 0 \leq |\nabla \mathbf{U}_h|$ because it yields a bounded solution as long as $\mathbf{U}_h \in C^\infty$ is bounded. Also, we can maintain that $0 \leq c$ because the mean value of the solution is always non-negative, at least for at most second order Lagrange functions. For higher-order Lagrange functions this is not always true. We will discuss this aspect in detail in section 5.3 and show in section 5.4 how an approach using Bernstein polynomials can prevent this issue.

The Edge-based Limiter (EB)

The work presented first in Cockburn and Shu (1998), discusses a limiter that linearly relates function values detected at edge midpoints $\mathbf{x}_m^k$ of all three edges $k = 1, 2, 3$ to element mean values of edge neighbors in order to detect extrema inside the element. It assumes that oscillations in the numerical solution are present, iff they are present in the $\mathcal{P}_1$- part of the solution— an assumption that could not be proven up to date and has been critically reflected upon in for example Kuzmin (2013b). To be precise, of interest is d_m, the distance from the mean value to the edge midpoints

$$d_m(\phi_{e,h}, \mathbf{x}_m^k) := \phi_{e,h}(\mathbf{x}_m^k) - \overline{\phi_{e,h}}, \qquad k = 1, 2, 3.$$

with $\overline{\phi_{e,h}}$ the mean value of ϕ_h, which for linear functions can be attributed to the center point $\mathbf{x}_e^c$, in Ω_e that will be computed using (4.2) and

$$\Delta\overline{\phi_{e,h}}(\mathbf{x}_m^k) := \alpha_1(\overline{\phi_{e_1,h}} - \overline{\phi_{e,h}}) + \alpha_2(\overline{\phi_{e_2,h}} - \overline{\phi_{e,h}})$$

for some parameters $\alpha_1, \alpha_2 \in \mathbb{R}$. We remark, that in this case, Δ is not the Laplace operator. The indices e_1 and e_2 are chosen in a way that the distance from $\mathbf{x}_m^k$ to the midpoint of the element Ω_e can be represented as a linear combination of the distances between the mean values of ϕ in Ω_e and Ω_{e_1}, and Ω_e and Ω_{e_2} as depicted in figure 4.1. In principle, this construction can be used on triangular and quadrilateral meshes. For triangular meshes and using the general notation introduced in (4.1), the limiter is defined as

$$\begin{aligned} \ell_k &= \widetilde{m}\left(d_m(\phi_{e,h}, \mathbf{x}_m^k), \nu_{lim}\Delta\overline{\phi_{e,h}}(\mathbf{x}_m^k)\right) \\ L_k(\phi_{e,h})(\mathbf{x}) &= \psi_k^{NC}(\mathbf{x}). \end{aligned}$$

The free parameter $\nu_{lim} \in \mathbb{R}$ is user-defined, ψ_k^{NC} the k-th linear non-conforming basis function with respect to the edge k and $\psi_k^{NC}(\mathbf{x}_m^l) = \delta_{kl}$ and $\widetilde{m}$ a TVB corrected minmod function with another free parameter $M_{lim} \in \mathbb{R}$, which is defined as

$$\widetilde{m}(a_1, a_2) = \begin{cases} a_1, & \text{for } |a_1| \leq M_{lim}(\Delta\mathbf{r}_e)^2, \\ m(a_1, a_2) & \text{otherwise,} \end{cases}$$

where $\Delta\mathbf{r}_e$ the radius of the largest inscribed circle of the respective element Ω_e and m the classical minmod function defined as

$$m(a_1, a_2) = \begin{cases} a_1, & \text{if } |a_1| \leq |a_2| \text{ and } a_1 a_2 > 0, \\ a_2 & \text{if } |a_2| \leq |a_1| \text{ and } a_1 a_2 > 0, \\ 0 & \text{otherwise,} \end{cases}$$

for arbitrary real numbers $a_1, a_2 \in \mathbb{R}$. For linear approximations the equality

$$d_m(\phi_{e,h}, \mathbf{x}_m^k) = \Delta\overline{\phi_{e,h}}(\mathbf{x}_m^k), \qquad \text{for } k = 1, 2, 3,$$

obviously always holds. The limiter is then defined for two different cases. For the case where $\sum_k \ell_k = 0$ it is

$$\widetilde{\mathbf{H}}(\phi_h)(\mathbf{x}) = \sum_{e \in \mathcal{E}} \chi_e^{\widetilde{\mathbf{H}}} \left(\overline{\phi_{e,h}} + \sum_{k=1}^{3} \ell_k \psi_k^{NC}(\mathbf{x}) \right) + \left(1 - \chi_e^{\widetilde{\mathbf{H}}}\right)\phi_{e,h}. \tag{4.4}$$

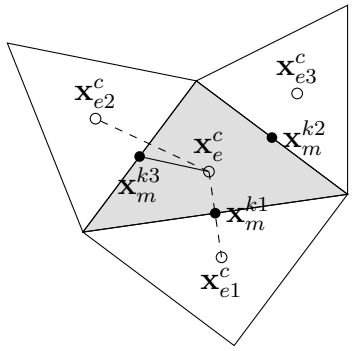

Figure 4.1.: Schematic of edge-based limiter. Cell mid point $\mathbf{x}^c_\cdot$ are marked as open circles, edge midpoints $\mathbf{x}^m_\cdot$ are full circles.

Otherwise we compute a modification of the scaling factors as follows:

$$\text{pos} = \sum_k (0, \ell_k), \quad \text{neg} = \sum_k (0, -\ell_k)$$
$$\theta^+ = \min\left\{1, \frac{\text{neg}}{\text{pos}}\right\}, \quad \theta^- = \min\left\{1, \frac{\text{pos}}{\text{neg}}\right\}.$$

In this case the limiter takes the form

$$\widetilde{\mathbf{H}}(\phi_h)(\mathbf{x}) = \sum_{e\in\mathcal{E}} \chi_e^{\widetilde{\mathbf{H}}} \left(\overline{\phi_{e,h}} + \sum_{k=1}^{3} \hat{\ell}_k \psi_k^{NC}(\mathbf{x}) \right) + \left(1 - \chi_e^{\widetilde{\mathbf{H}}}\right) \phi_{e,h}, \tag{4.5}$$

with $\hat{\ell}_k := \theta^+ \max_i(0, \ell_i) + \theta^- \max_i(0, -\ell_i)$. As this limiter is only defined for linear functions the general application to higher-order polynomials takes the following form:

- Compute L_2 projection onto the space of linear polynomials for each element.
- Compute limited version of linear approximation as in (4.4) or (4.5).
- If on the interpolation points the computed limited solution and the linear approximation coincide, no limiting needs to be done, if not the limited solution is the computed quantity.

The limiter fulfills four requirements. It conserves mass and momentum because the integrals of $\mathbf{L}_k$ over all Ω_e vanish:

$$\int_{\Omega_e} L_k(\mathbf{U}_h) d\mathbf{x} = \int_{\Omega_e} \psi_k^{NC} d\mathbf{x} = 0, \qquad \forall k.$$

As shown in Cockburn et al. (1999), the limiter fulfills a maximum principle which is equivalent to preservation of high-order accuracy (see previous section on total variation). Boundedness of the limited solution then follows automatically from the maxi-

mum principle as well. The reduction of total variation follows from the restriction to first order polynomials at critical points and the limiting of the first order derivatives.

An addition to the edge-based limiter is presented in Xing et al. (2010) and we will denote it with $\mathbf{H}_+$. It is applied right after the edge-based limiter in the fashion $\widetilde{\mathbf{H}}_+\widetilde{\mathbf{H}}(\mathbf{U}_h)$ and we will denote the product of both limiters with (EB+). The advantage is that it preserves positivity–a desirable property for inundation simulations. The proof of positivity preservation for this limiter itself relies on the underlying numerical quadrature and for the computation of the limiter, the set of Gauss quadrature points on the triangle $\Xi := \{\boldsymbol{\xi}_k \in \mathbb{R}^2, k \in \mathcal{K}\}$ with the corresponding index set $\mathcal{K}$ is important. The positivity preserving limiter is then defined as

$$\widetilde{\mathbf{H}_+}(\mathbf{U}_h)(\mathbf{x}, t_n) = \sum_{e\in\mathcal{E}} \left(\theta_e \left(\mathbf{U}_{e,h}(\mathbf{x}, t_n) - \overline{\mathbf{U}_{e,h}(t_n)}\right) + \overline{\mathbf{U}_{e,h}(t_n)}\right),$$
$$\text{with } \theta_e = \min\left(1, \frac{\overline{\phi_{e,h}}}{\overline{\phi_{e,h}} - m_{e,h}}\right),$$

where $m_{e,h} = \min_{\mathbf{x}\in\Xi} \phi_{e,h}(\mathbf{x})$ is the minimum of the free surface height on the Gauss quadrature points. It can basically be seen as a linear scaling around cell averages. From Xing et al. (2010), we can also see that the limiter is only active on dry or near dry elements.

The application of the limiter does not affect the conservation property or the boundedness of the solution as can be seen in Xing et al. (2010). Later, in section 4.3.2, we will confirm that the extension does improve the result of the limiter with respect to positivity preservation and well-balancing. High-order accuracy is preserved under the CFL condition

$$\lambda_{max}\frac{\Delta t}{\text{vol}(\Omega_e)}\text{vol}_1(\Omega_e) \leq 1, \tag{4.6}$$

with λ_{max} the maximum eigenvalue, $\text{vol}(\Omega_e)$ the volume of Ω_e and $\text{vol}_1(\Omega_e)$ the perimeter of Ω_e as can be seen from Zhang and Shu (2011).

The Vertex-based Limiter (VB)

The vertex-based limiter presented in Kuzmin (2010, 2013b) is a modification of the limiter presented in Barth and Jespersen (1989). It is a nodal modification using a Taylor-series-based approach which was first presented in Bell et al. (1988). The limiter takes into account values from patches $P(\mathbf{x}_i)$ surrounding a node $\mathbf{x}_i$, i.e. $P(\mathbf{x}_i) := \{\Omega_e \subset \Omega | \mathbf{x}_i \in \Omega_e\}$. A schematic of a node patches can be found in figure 4.2. The limited Taylor-expansion-based approach is then performed using scaling factors $\gamma_{e,|\boldsymbol{\alpha}|} \in \mathbb{R}$:

$$\tilde{\mathbf{H}}(\phi_h)(\mathbf{x}) = \sum_{e\in\mathcal{E}} \left(\overline{\phi_{e,h}} + \sum_{|\boldsymbol{\alpha}|=1}^{n} \gamma_{e,|\boldsymbol{\alpha}|}\frac{(\mathbf{x} - \mathbf{x}_e^c)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}(D^{\boldsymbol{\alpha}}\phi_{e,h})(\mathbf{x}_e^c)\right), \tag{4.7}$$

with multiindices $\boldsymbol{\alpha} \in \mathbb{N}^2$, $\phi_{e,h}$ the numerical solution for ϕ in element $\Omega_e \subset \Omega$ and the point $\mathbf{x}_e^c$ the center point of element Ω_e. Using the general notation from (4.1), we have

$$\begin{aligned} \chi_e^{\widetilde{\mathbf{H}}} &= 1 \qquad \forall e \in \mathcal{E} \\ \ell_k &= \gamma_{e,k} \\ \mathbf{L}_k &= \sum_{|\boldsymbol{\alpha}|=k} \frac{(\mathbf{x}-\mathbf{x}_e^c)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} D^{\boldsymbol{\alpha}} \Big|_{\mathbf{x}=\mathbf{x}_e^c}. \end{aligned}$$

The parameters $\gamma_{e,k} \in \mathbb{R}$ with $k \in \mathbb{N}$ are in this case determined using the formula

$$\gamma_{e,k} = \min_{|\boldsymbol{\alpha}|=k-1} \min_i \begin{cases} \min\left\{1, \frac{(D^{\boldsymbol{\alpha}}\phi_e)^{max}(\mathbf{x}_i) - \overline{D^{\boldsymbol{\alpha}}\phi_e}}{(D^{\boldsymbol{\alpha}}\phi_e)(\mathbf{x}_i) - \overline{D^{\boldsymbol{\alpha}}\phi_e}}\right\}, & \text{if } (D^{\boldsymbol{\alpha}}\phi_e)(\mathbf{x}_i) - \overline{D^{\boldsymbol{\alpha}}\phi_e} > 0 \\ 1, & \text{if } (D^{\boldsymbol{\alpha}}\phi_e)(\mathbf{x}_i) - \overline{D^{\boldsymbol{\alpha}}\phi_e} = 0 \\ \min\left\{1, \frac{(D^{\boldsymbol{\alpha}}\phi_e)^{min}(\mathbf{x}_i) - \overline{D^{\boldsymbol{\alpha}}\phi_e}}{(D^{\boldsymbol{\alpha}}\phi_e)(\mathbf{x}_i) - \overline{D^{\boldsymbol{\alpha}}\phi_e}}\right\}, & \text{if } (D^{\boldsymbol{\alpha}}\phi_e)(\mathbf{x}_i) - \overline{D^{\boldsymbol{\alpha}}\phi_e} < 0, \end{cases}$$

where $(f_i)^{max} = \max_{\Omega_e \in P(i)} f(\mathbf{x}_e^c)$ and $(f_i)^{min} = \min_{\Omega_e \in P(i)} f(\mathbf{x}_e^c)$ are the patch-minima and maxima of the centroid at the ith degree of freedom. The factors $\gamma_{e,k}$ are scaling parameters for the kth derivative in the Taylor expansion of (4.7) and contain information of nodal and mean values of the $(k-1)$st derivative of the function. The switch $\chi_e^{\widetilde{\mathbf{H}}} \equiv 1$ will be used here, since the scaling parameter determine whether or not limiting is required. So that no further criterion is needed.

For higher order approximations the parameters $\gamma_{e,k}$ are determined in a hierarchical approach starting with the highest order derivative and then set

$$\gamma_{e,k} := \max_{k \le l} \gamma_{e,l} \quad \text{for } k \ge 1,$$

until we determine a $\gamma_{e,\tilde{k}}$ to be 1, then all the $\gamma_{e,k} = 1$ for $k \le \tilde{k}$. We remark that this approach is contrary to the assumption that oscillation are only present if they are present in the $\mathcal{P}_1$-part of the function.

The choice of scaling parameters $\gamma_{e,k}$ lead to a geometric constraint of the form:

$$(\phi_i)^{min} \le \phi(\mathbf{x}_i) \le (\phi_i)^{max}. \tag{4.8}$$

A truncation error analysis can show, that the limiter retains the desired accuracy even for higher order functions. A detailed discussion on this issue will be presented in chapter 5. In Aizinger (2011) it has also been shown that the solution computed with this limiter solves a local elementwise minimization problem of the form

$$\min_{(\widetilde{\phi_{e,1}},\ldots,\widetilde{\phi_{e,m}})^\top \in \mathbb{R}^m} \sum_{k=1}^{m_e} \left(\widetilde{\phi_{e,k}} - \phi_{e,k}\right)^2,$$

subject to

$$0 \le \sum_{k=1}^{m_e} \widetilde{\phi_{e,k}} \psi_k \le \max_{\mathbf{x}_i \in \Omega_e} (\phi_i)^{max},$$

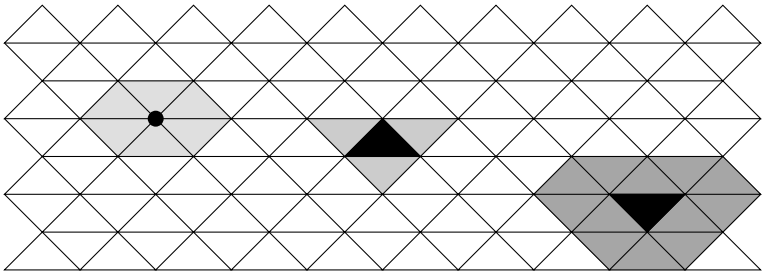

Figure 4.2.: Schematic of patches and neighborhoods. Depicted are a nodepatch (light gray) for the black node, the edge-neighborhood (gray) and patch-neighborhood (dark gray) for the black triangle.

i.e. the limiter provides the modified coefficients $\widetilde{\phi_{e,k}}$ that deviate least from the actual computed ones and that satisfy pointwise a boundedness condition as in (4.8). In that sense, the limiter leads to optimal results with respect to the coefficients of the approximation. However, the lower zero bound for the limited solution for the nodal limiter, that is needed to fully ensure that no negative fluid heights are being computed, needs to be imposed explicitly. In that sense, the limiter is not fully parameter-free. Also, it conserves mass and momentum since the integrals over the operators $\mathbf{L}_k$ vanish and a simple Taylor expansion shows that the gradient of the limited solution is lessened by the limiting procedure.

4.3.2. Theoretical Comparison

In section 4.3.1 we introduced limiters that are conceptually different, yet all fulfill typical requirements that are posed on limiters following Cockburn and Shu (1998). We theoretically study these limiters with respect to further criteria such as computational cost, wellbalancedness, positivity preservation and dependence on free parameters. Our findings are summarized in table 4.1 and elaborated in the following.

Computational cost

Relative to numerical methods such as FEM or FV the number of degrees of freedom for DG methods is large. Therefore, and because the limiting takes place in every intermediate time step of the Runge Kutta scheme, the additional costs, that originate from limiting have to be preferably low. In order to gain insight into the computational costs of limiters, we count the respective floating point operations and remark, that with this approach the cell and grid communication as well as IF loops are not taken into account, despite their importance.

The costs of every limiter from 4.3.1 is composed of the cost of the limiter per element times the number of elements that have to be limited. The latter depends on the definition of $\chi_e^{\widetilde{\mathbf{H}}}$, which usually detects the rough regions, that need limiting. We assume that the index set corresponding to the number of degrees of freedom per

element is of cardinality m, i.e. $|\mathcal{J}_e| = m$. To improve the readability, we kept the following computations as brief as possible.

For the vertex-based limiter (VB), $\chi_e^{\widetilde{\mathbf{H}}} \equiv 1$ holds, so that it is applied to all elements and the limiting is solely controlled by the computation of the scaling parameters $\gamma_{e,k}$. In contrary thereto, the other limiters require the elementwise computation of $\chi_e^{\widetilde{\mathbf{H}}}$.

The (VB) limiter is an extension of the L_2 limiter (L2). If we set all $\gamma_{e,k}$ in (4.7) equal to zero, we obtain (L2). The latter only involves the computation of cell averages, which using the numerical quadrature from (4.2) costs $2(m^2 + m)$ flops.

The remaining flops of the (VB) limiter depend on how many higher-order terms are taken into account. We will denote this order with n for the moment and introduce a running index $k = 1, \dots, n$. The first order term ($k = 1$) requires the computation of the mean values of the two derivatives in x- and y-direction at a total cost of $2(2m + 2m^2)$ and further $3m+1$ and $2(3+3)$ operations for the computation of ℓ_1 and $\mathbf{L}_1$ respectively.

For the higher-order terms additional mean values of the $(k-1)$st order derivatives are needed to evaluate $\gamma_{e,k}$. This amounts for the general kth order term with $k > 1$ to:

$$2^k \left\{(2m + 2m^2) + (3 + 3k)\right\} + 2^{k-1} \left[(2m + 2m^2) + (1 + 3m)\right],$$

where the terms in the curly bracket contain computation costs of the kth order derivative at the element centroid and the operator $\mathbf{L}_k$ and the square bracket contains the additional computation costs of the $(k-1)$st derivatives and the parameters ℓ_k. Finally, the overall summation costs $2n - 1$ operations per element. Putting all this together, for $n \geq 2$, this all sums up to

$$\begin{aligned} |\mathcal{E}|\Big(&[(2m + 2m^2)] + [13 + 7m + 4m^2] + (2n - 1) \\ &+ \sum_{k=2}^{n} 2^k(2m^2 + 2m + 3 + 3k) + 2^{k-1}(2m^2 + 5m + 1)\Big). \end{aligned}$$

The first line of the display shows the number of flops for the zeroth and first order term in square brackets while the second line lists all the higher order terms. Of interest is the coefficient of m^2. Multiplying the sum out reveals that the leading coefficient is 2 for the (L2) limiter, 6 for the first order (VB) and $(4 + 3 \cdot 2^{n-2})$ for higher order n.

The edge-based limiter (EB) requires a projection onto $\mathcal{P}_1$ and back, which, computationally, is realized using the precomputed Vandermonde matrix $\mathbf{V}$. Using Weierstrass' Theorem the projection onto the space of linear polynomials costs $3 \cdot 2m$ flops per element for computing the first three modes, while the back projection costs $2m^2$ flops, due to the matrix vector multiplication with a dense inverse Vandermonde matrix. Then, for all three edges we compute d_m at a cost of 3 operations times $(m^2 + 2m)$, and Δ at a cost of 4 flops for solving a 2×2 linear system. The ℓ_k cost 1 multiplication and the minmod function 3 operations. Then, the sum criterion is evaluated in 3 additions, the computation of pos and neg in 2 times 3 additions and multiplications for -ℓ_k. The parameters θ cost 2 divisions and the computation of $\hat{\ell_k}$, if needed, cost 3 times 1 addition and 2 multiplications. Finally the summation for $\widetilde{\mathbf{H}}$ costs 4 additions and 3

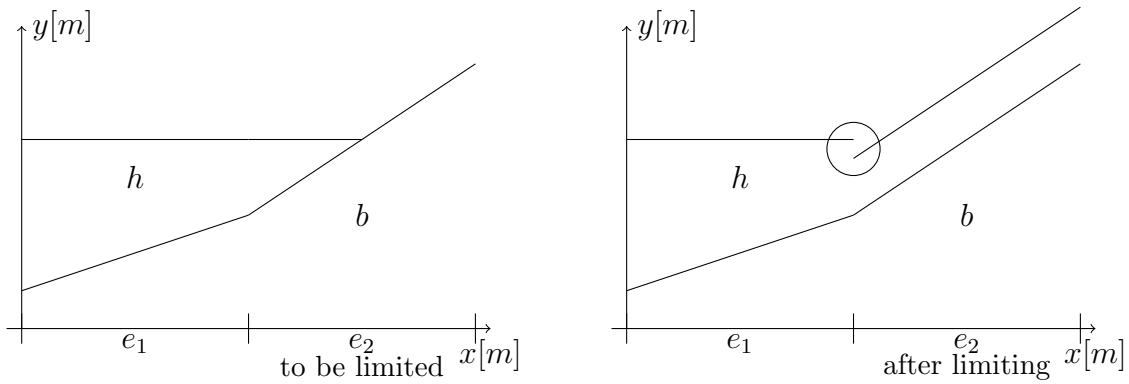

Figure 4.3.: Simple L_2 limiter. 1D schematic of violation of well-balancedness

multiplications. Summing up we have

$$6m + 2m^2 + 3\big(3 + m^2 + 2m + 4 + 4\big) + 3 + 6 + 3 + 2 + 9 + 7 = 5m^2 + 9m + 63.$$

The computation of (EB+) involves 5 additional operations, because it is a processing of already computed quantities, which has no effect on the leading order of m^2.

Wellbalancing

Preservation of steady states is an important aspect in order to prevent artificial velocities from developing. In section 4.5 we describe the wellbalancing scheme in our model. An important aspect thereby is that dry nodes are not artificially wetted by the limiter in order for the algorithm to detect partially wet areas correctly.

It is shown in Xing and Zhang (2013); Xing et al. (2010) that the limiter (EB+) does not destroy the well-balanced property.

We observe, that the (L2) limiter is not well-balanced as a modification of the sea surface height will not automatically result in a modification of the bathymetry and thus change the derivative of the water height which will introduce artificial waves into the simulation if not a separate modification of bathymetric data is taken into account. Figure 4.3.2 illustrates this issue.

The wellbalancedness of (VB) strongly depends on the order of approximation. For $n = 0$ it coincides with the L_2 limiter and is therefore not wellbalanced. But for n equal to the order of polynomials used, we achieve wellbalancing, because dry nodes will not be wetted during the limiting procedure due to the choice of $\gamma_{e,k}$. For partially dry cells the $\gamma_{e,k}$ will be 1 and therewith not affect the nodal values of the transition cells.

Non-negativity preservation

Preservation of non-negativity is equivalent to requiring the lower bound c as described in section 4.3.1 to be bounded below by 0. For (L2) this the non-negativity of mean values over the element ensures the bound. For the (VB) to be non-negativity preserv-

	(L2)	(EB)	(VB)	(EB+)
Accuracy	1	p	p	p
Computational Cost (leading coefficient)	2	5	$\min(6, 4+3\cdot 2^{n-2})$	5
Positivity Preservation	yes	no	yes	yes
Free Parameter	no	yes (M_{lim}, ν_{lim})	yes	yes(M_{lim}, ν_{lim})
Wellbalancing	no	no	yes	yes

Table 4.1.: Comparison of different slope limiters for DG methods.

ing, we have to numerically force the patch minima to stay non-negative by introducing a lower bound:

$$(f_i)^{min} = \max\left(\min_{\Omega_e \in P(\mathbf{x}_i)} f(\mathbf{x}_e^c), 0\right)$$

The (EB) itself is not non-negativity preserving, because shape preserving requirements as in Cockburn (1998) can be violated at local extrema, which, in the case of wetting and drying problems, can lead to undershoots at wet/dry interfaces and therewith to negative water heights. However with the extension, (EB+) does preserve non-negativity under the CFL condition (4.6) as can be seen from Xing et al. (2010).

We remark that the preservation of positivity for discontinuous bathymetry is achieved by reconstructing the sea surface height at every time-step t in the following way:

$$\phi_h(\mathbf{x}, t) = \max\left(0, \phi_h(\mathbf{x}, t) + b_h(\mathbf{x}) - \max(b_h(\mathbf{x}), b_h(\mathbf{x\prime}))\right)$$
$$\phi_h(\mathbf{x\prime}, t) = \max\left(0, \phi_h(\mathbf{x\prime}, t) + b_h(\mathbf{x\prime}) - \max(b_h(\mathbf{x}), b_h(\mathbf{x\prime}))\right)$$

with $\mathbf{x}, \mathbf{x\prime}$ neighbors over an element edge.

Parameters

Most of the mentioned limiters include tuning parameters that are of essential importance in obtaining good results. In turn, they make the limiters less user-friendly and an adjustment necessary for every testcase anew. The limiter (VB) requires to impose a lower boundary of zero for patch minima in order to preserve non-negativity of the fluid height. The limiters (EB) and (EB+) contain parameters for the modified minmod function (M_{lim}) as well as for the limiter coefficients (ν_{lim}).

The findings of our analysis are summarized in table 4.1. They reveal the different strengths of the limiters. For example, the weak (L2) limiter is the most inexpensive and, since it maintains non-negativity, it can be used for trial runs of wetting and drying scenarios, that do not require high-order accuracy. A further advantage of the low order accuracy will also be that one can work with rather large timesteps because

in the rough regions the polynomial order is drastically reduced. The limiter (VB) seems to be preferable because of the wellbalancedness and non-negativity preservation. However, our computations revealed it, in terms of flops, to be the most expensive one. We remark, however, that this is only due to the order of approximation and that we can also obtain reasonable results with $n = 1$. The limiter (EB) does not seem to be a good choice for wetting and drying problems, while however (EB+) is. The choice of parameters make them the hardest to tune though.

4.3.3. Numerical Comparison

We study numerical convergence and conservation properties of our model in combination with the limiters from section 4.3.2 with two partially dry test problems with known analytical solutions and different uniform spatial resolutions.

Testcase 5. *Homogenous Wet/Dry Shocktube*
We consider a channel of length $[0, 2] \times [0, 10]$ with reflecting boundaries and initial conditions

$$\begin{aligned} h(\mathbf{x}, 0) &= \begin{cases} 0.25 & 0 \leq x \leq 5 \\ 0.0 & \text{otherwise.} \end{cases} \\ \mathbf{u}(\mathbf{x}, 0) &= \mathbf{0}. \end{aligned}$$

Testcase 6. *Paraboloid 2D Basin*
We consider a domain: $\Omega = [-L, L]^2$ with parameters: $D_0 = 50, L = 430620, \eta_0 = 2$, reflecting boundaries, Coriolis forcing and

$$\begin{aligned} A &= \frac{(D_0 + \eta_0)^2 - D_0^2}{(D_0 + \eta_0)^2 + D_0^2}, \qquad D(\mathbf{x}) = D_0 \left(1 - \frac{x^2 + y^2}{L^2}\right) \\ \eta(\mathbf{x}) &= D_0 \left(\frac{\sqrt{1 - A}}{1 - A} - 1 - \frac{x^2 + y^2}{L^2} \left(\frac{1 - A^2}{(1 - A)^2} - 1\right)\right) \\ h(\mathbf{x}, 0) &= \begin{cases} D(\mathbf{x}) + \eta(\mathbf{x}) & \text{if } D(\mathbf{x}) + \eta(\mathbf{x}) \geq 0 \\ 0 & \text{otherwise.} \end{cases} \\ \mathbf{u}(\mathbf{x}, 0) &= \mathbf{0} \\ b(\mathbf{x}) &= D_0 \frac{x^2 + y^2}{L^2}. \end{aligned}$$

The analytical solutions for these problems can be found in Toro (2001) and Thacker (1981) respectively. The expected convergence rate of the model is $n + 1$ for a smooth problem and basis functions of degree $n \in \mathbb{N}$ (see Hesthaven and Warburton (2008)). Our test problems are continuous. Shocks can not occur at the wet/dry front as can be seen from Toro (2001) where we find a proof that wet and dry states are always connected by a contact discontinuity. However, at the wet/dry interface the analytical solutions are not differentiable any longer which will effect the convergence rates.

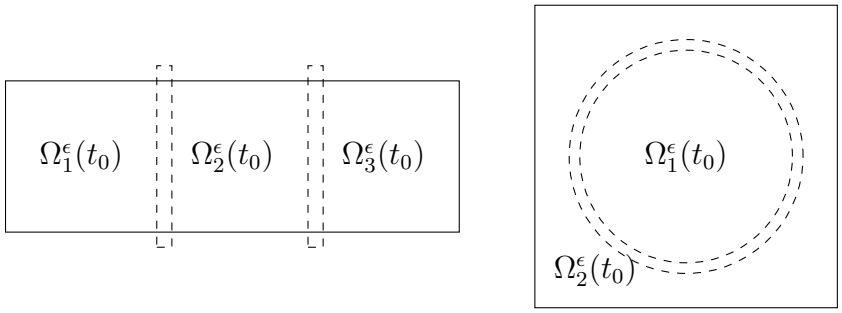

Figure 4.4.: Schematic of partial areas for wet/dry shocktube (left) and paraboloid basin (right) at time $t = t_0$. The dashed regions show the ϵ neighborhoods that are taken out of computing error norms.

Error measures for split domains

Due to the missing global differentiability of the test problems, we will not be able to obtain the theoretical convergence rates. To get more insight into the behavior, we propose to compute global errors as introduced in (3.11) as well as separate errors for only the smooth parts of the domain.

As is outlined in figure 4.4, we will neglect the irregular parts of the domain and only compute errors for the remaining connected regions. The error norms are then defined as:

$$E_p^\epsilon(\mathbf{f}_h) = \sum_k \|\mathbf{f}_h - \mathbf{f}\|_{p,\Omega_k^\epsilon}, \qquad \text{with } p \in \{1, 2, \infty\}, \tag{4.9}$$

with $\mathbf{f}_h$ the numerical and $\mathbf{f}$ the analytical solution. The parameter $\epsilon \in \mathbb{R}$ determines the width of the irregular regions that are not considered for the computation of the error and p indicates which norm we are using. The areas Ω_k^ϵ with $k = k(t)$ are defined for every test separately and vary over time.

For the wet/dry shocktube we have $k = 3$ since the critical points of the solution are the head and the tail of the rarefaction wave. The speed of them can be determined analytically, so that the domains take on the form $\Omega_1^\epsilon(t) := \{\mathbf{x} \in \Omega | x \leq t(5 - \sqrt{\phi(\mathbf{x}, 0)} - \epsilon\}$, $\Omega_2^\epsilon(t) := \{\mathbf{x} \in \Omega | x \in [t(5 - \sqrt{\phi(\mathbf{x}, 0)} + \epsilon, t(5 + 2\sqrt{\phi(\mathbf{x}, 0)}) - \epsilon\}$ and $\Omega_3^\epsilon(t) := \{\mathbf{x} \in \Omega | x \geq t(5 + 2\sqrt{\phi(\mathbf{x}, 0)}) + \epsilon\}$.

In the paraboloid 2D basin test, we know that the wet/dry boundary is the only irregular region, so that the domains are then defined as $\Omega_1^\epsilon(t) := \{\mathbf{x} \in \Omega | \phi(\mathbf{x}, t) > \epsilon\}$ and $\Omega_2^\epsilon(t) := \{\mathbf{x} \in \Omega | \phi(\mathbf{x}, t) = 0\}$.

Throughout this section we work with linear basis functions, i.e. $n = 1$, because the use of higher order Lagrange polynomials is not readily possible. A discussion on the occurring difficulties with higher-order functions and approaches to overcome them, can be found in section 5.2. Examples of simulation results are depicted in figure 4.5 for the

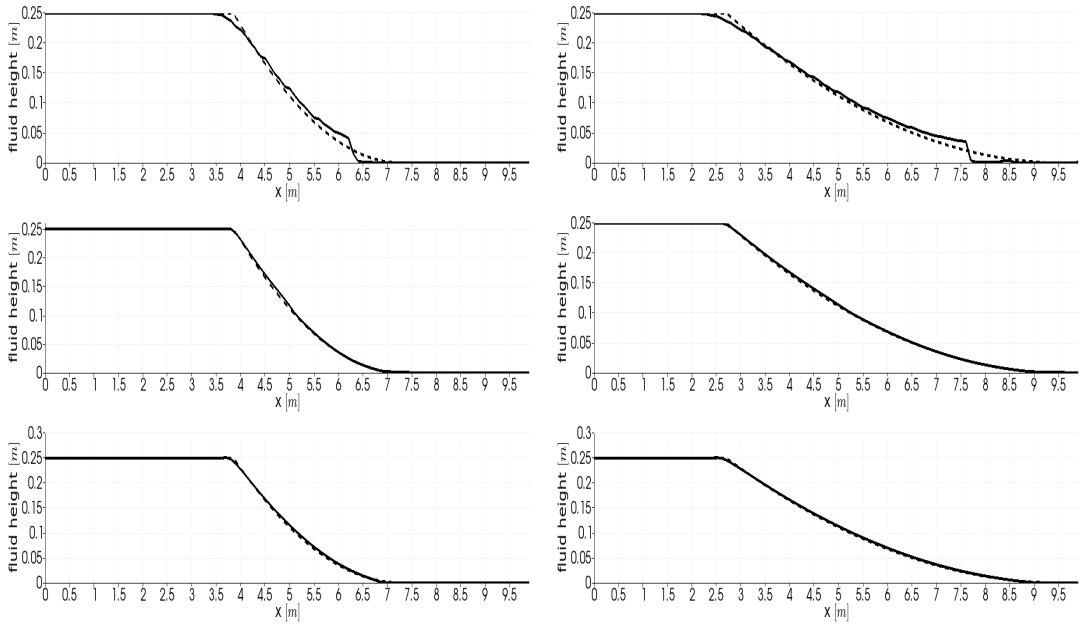

Figure 4.5.: Wet/dry shocktube after $0.75s$(left) and $1.5s$(right) with the simple L_2 limiter (top row), the vertex-based limiter (middle row), and the edge-based limiter (bottom row). Depicted are the analytical (dashed line) and the numerical solution.

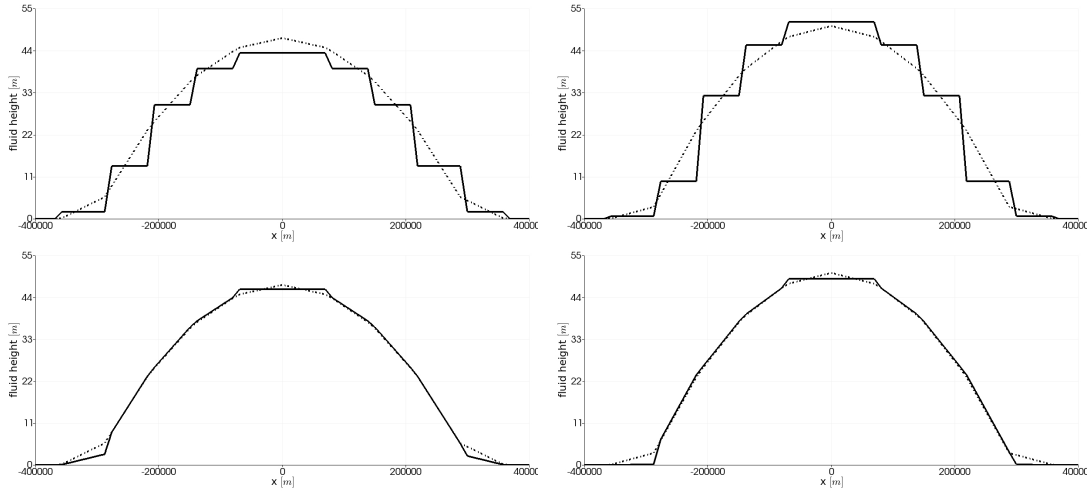

Figure 4.6.: Paraboloid Basin. Simulation with 512 elements. Plot over line $x = y$ for the simple limiter (top row) and the nodal limiter (bottom row). Depicted are the results after $t = 190s$(left) and $t = 380s$(right). The plot shows the analytical (dashed line) and the numerical solution (solid line).

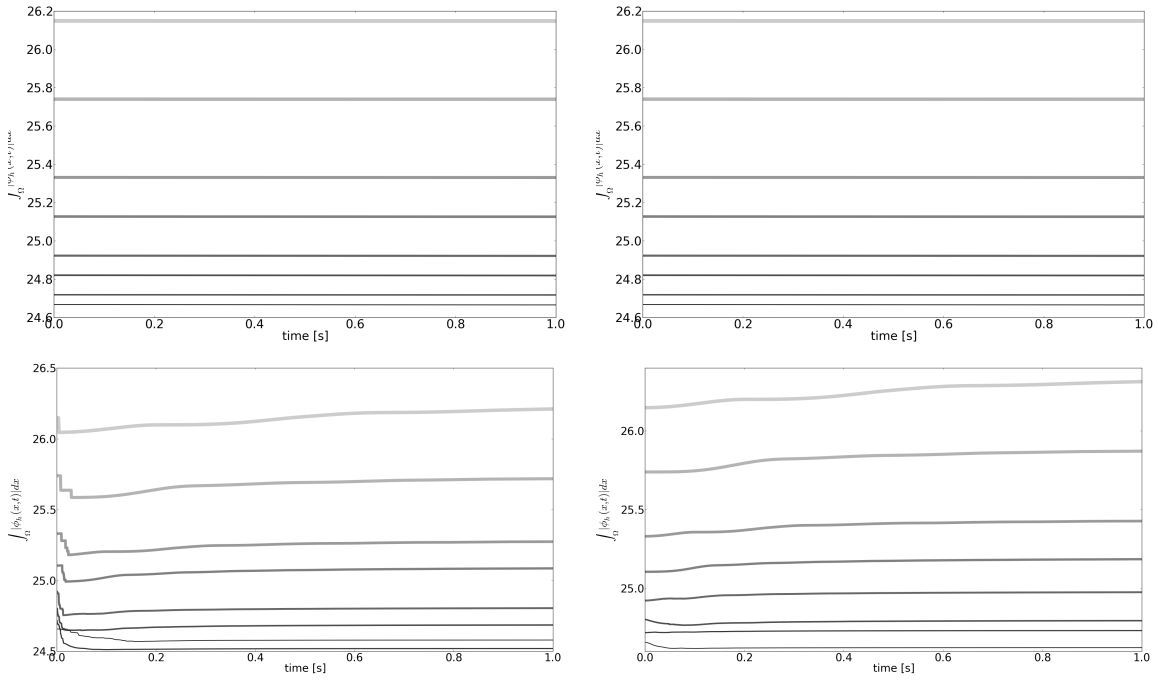

Figure 4.7.: Wet/dry Shocktube. Plots of mass in L_1 norm for $0-1s$ for simulations with the L_2-limiter (top left), vertex-based limiter (top right), Taylor-expansion-based limiter (bottom left), and edge-based limiter (bottom right).

wet/dry shocktube and in figure 4.6 for the paraboloid basin. The simple (L2) limiter gave the weakest results in terms of pointwise accuracy. In figure 4.5, we see, that its redistribution of the first mode onto the nodes influenced the wet/dry interface with the result that the arrival time of the wave is underestimated. In both simulations we observed that the largest possible timestep was dependent on the accuracy of the limiter. The less accurate the limiter, the larger the timestep. For example for the paraboloid basin test, we chose a timestep of $\Delta t = 5.0$ for the simulation in combination with the (L2) limiter and a timestep of $\Delta t = 2.5$ for the one in combination with the (VB) limiter for a spatial resolution of $\Delta \mathbf{x}_{min} = 17994.48$. Maximal timesteps are not of only interest in our numerical study. In the following, we test our implementation with respect to conservation and convergence properties.

Conservation Properties

Maintaining a constant mass, provided sensible boundary conditions, is a desirable property of wet/dry simulations. Figure 4.7 shows the mass measured in the discrete L_1 norm for the first second of simulation time. The different pictures correspond to simulations with the different limiters as introduced in section 4.3.1. The coloring and strength of the lines refer to the level of refinement in the following way: The lighter and thicker the line, the coarser the simulations. We started with a shortest edge of $\Delta \mathbf{x}_{min} = 0.5$ and then reduced it by bisection to the finest mesh with the shortest

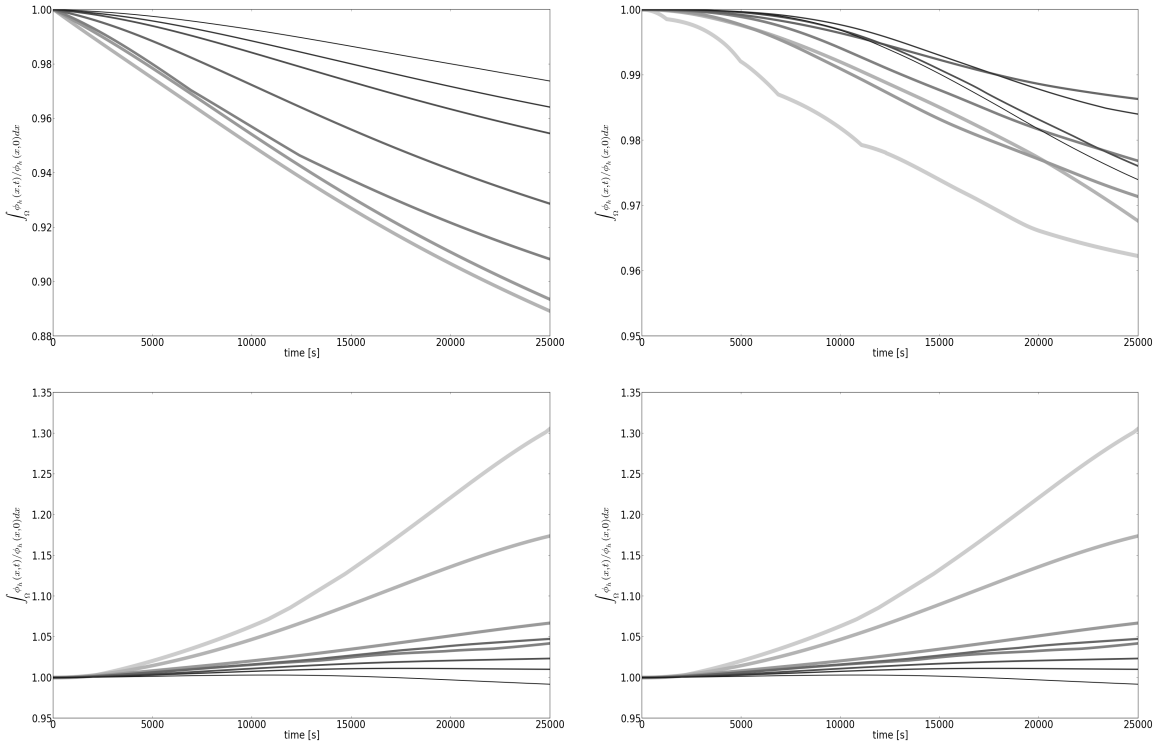

Figure 4.8.: Paraboloid Basin. Plots of mass L_1 norm for $0 - 25\ 000s$ for simulations with the L_2-limiter (top left) and the vertex-based limiter (top right), Taylor-expansion-based limiter (bottom left), and edge-based limiter (bottom right).

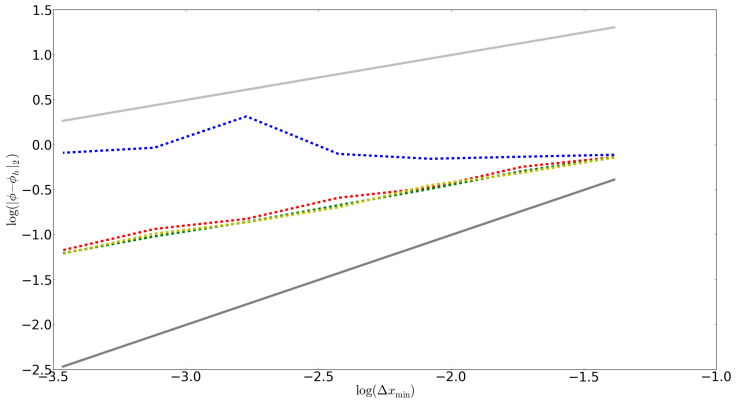

Figure 4.9.: Wet/dry shocktube. Double logarithmic plot of global convergence study in L_2-norm with different limiters: with different limiters: nodal(red), shu(green), l2 (blue) and xing (yellow). The grey lines indicate linear function with slopes 0.5 and 1 for comparison.

	(L2)		(VB)		(EB)		(EB+)	
Testcase	L_2	L_1	L_2	L_1	L_2	L_1	L_2	L_1
Wet/Dry Shocktube	0.00	0.00	0.50	0.99	0.52	0.94	0.67	1.18
Paraboloid 2D Basin	0.42	0.49	0.94	0.94	1.24	1.29	1.35	1.38

Table 4.2.: Global convergence orders for linear Lagrange ansatz in combination with different limiters

edge being $\Delta\mathbf{x}_{min} = 0.0442$. The results obtained with (L2) and (VB) are satisfactory. The computed mass for the (EB)-and (EB+)-limited solutions show some irregularities and mass lost for the first $0.2s$ which might be due to the initial discontinuity. For all simulations, a converging behavior is observed. We remark that the observation, that the discrete mass is changing with resolution depends on the change of accuracy of the numerical integration with increasing resolution The more quadrature points are used the lesser the interpolation error. Analogously, figure 4.8 shows the normalized mass for the first 25 000 s for simulations with the four limiters. We observe a converging behavior with increasing resolution and a mass lost of approximately 3% (L2) and 1.5% (VB, EB, EB+) for the finest mesh with $\Delta\mathbf{x}_{min} = 50896.06$. The reason, that perfect mass conservation is not achieved is due to filters in our model that, for example, prevent the division of two very small quantities when computing velocity from momentum and the long simulation time.

Accuracy

To study the convergence properties of our model, we decreased the spatial resolution of our model by repeated bisection of triangles from $\Delta\mathbf{x}_{min} = 0.2231$ to $\Delta\mathbf{x}_{min} =$

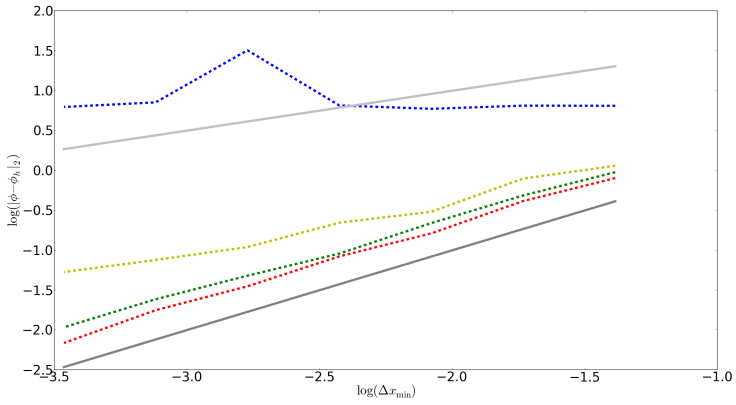

Figure 4.10.: Wet/dry shocktube. Double logarithmic plot of global convergence study in L_1-norm with different limiters: with different limiters: nodal(red), shu(green), l2 (blue) and xing (yellow). The grey lines indicate linear function with slopes 0.5 and 1 for comparison.

	(L2)		(VB)		(EB)		(EB+)	
Testcase	L_2	L_1	L_2	L_1	L_2	L_1	L_2	L_1
Wet/Dry Shocktube (Ω_1^ϵ)	6.69	7.22	5.75	5.77	9.64	10.08	5.92	5.74
Wet/Dry Shocktube (Ω_2^ϵ)	0.0	0.0	0.84	0.86	0.0	0.0	1.19	1.22
Wet/Dry Shocktube (Ω_3^ϵ)	0.71	0.71	4.05	3.98	0.71	0.71	4.15	4.09
Paraboloid 2D Basin (Ω_1^ϵ)	0.42	0.49	0.83	0.57	1.25	1.24	1.25	1.01
Paraboloid 2D Basin(Ω_2^ϵ)	0.0	0.0	0.82	1.20	1.09	1.62	1.14	1.67

Table 4.3.: Local convergence orders for linear Lagrange ansatz in combination with different limiters

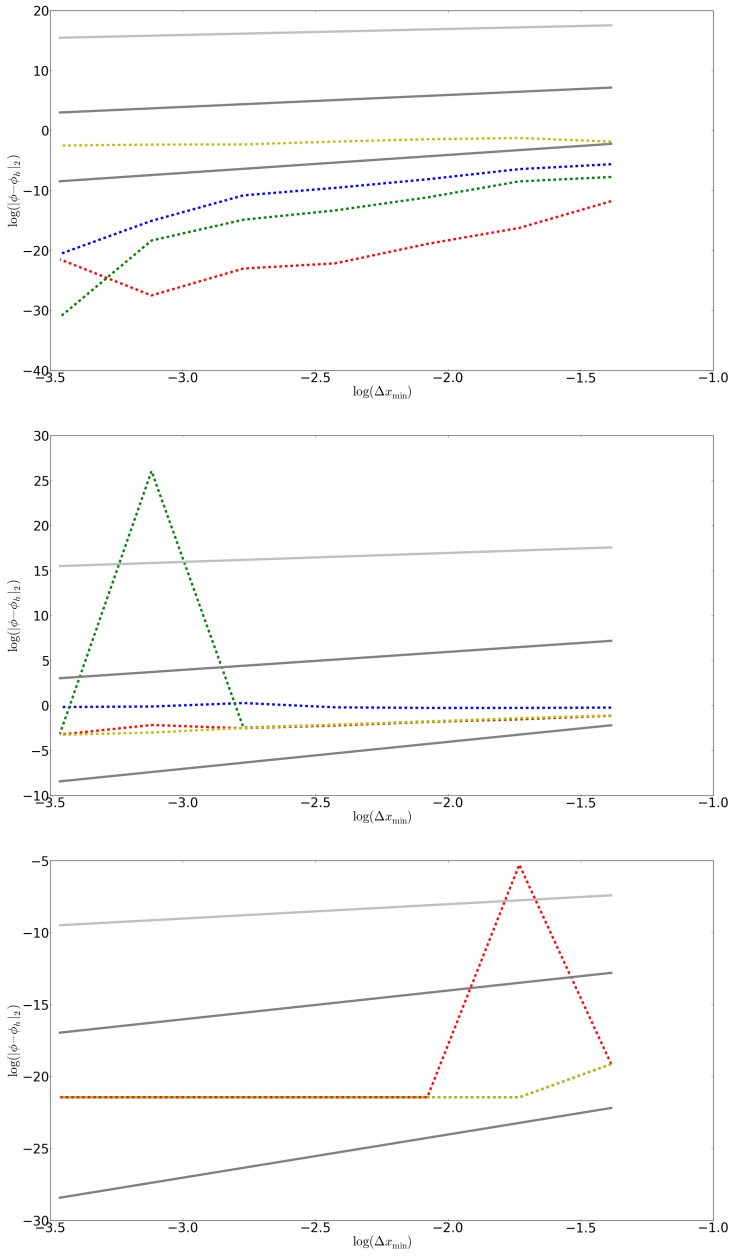

Figure 4.11.: Wet/dry shocktube. Local L_2 error in areas Ω_1^ϵ (top), Ω_2^ϵ (middle), and Ω_3^ϵ (bottom). Double logarithmic plot of convergence study with different limiters: nodal (red), shu (green), l2 (blue) and xing (yellow). The grey line indicate linear functions with slope 1, 2 and 3.

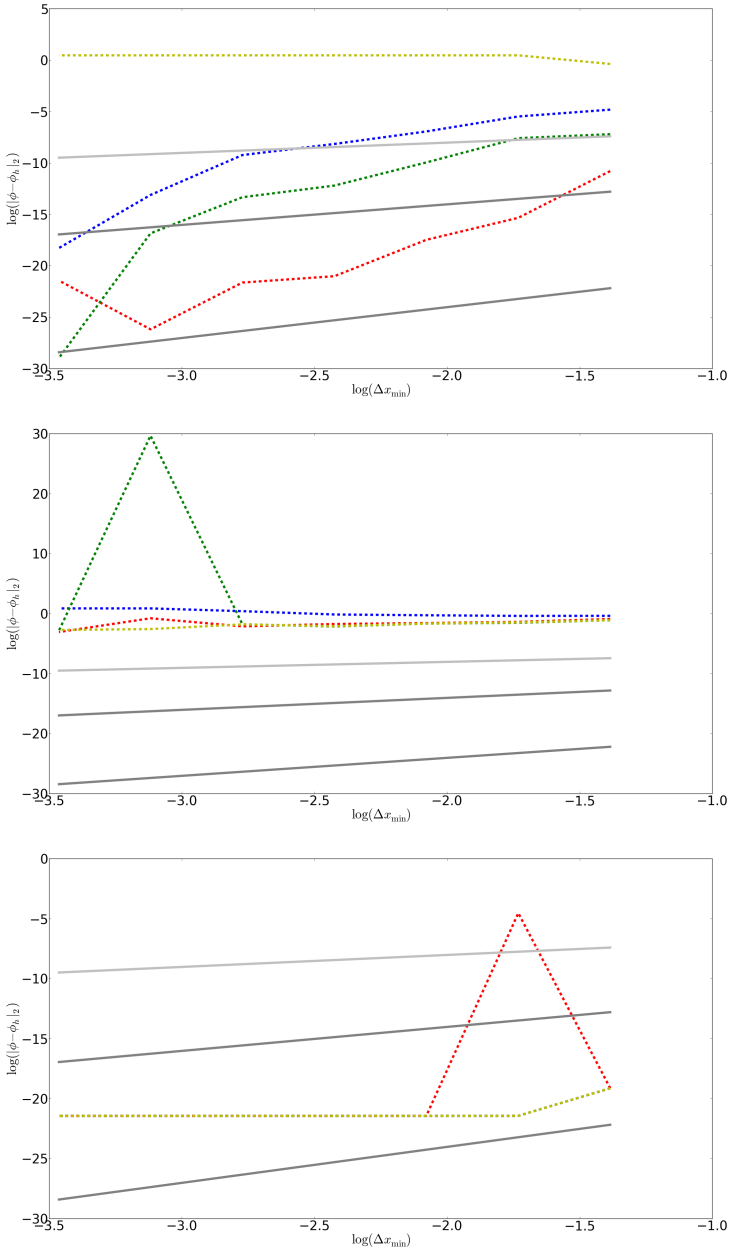

Figure 4.12.: Wet/dry shocktube. Local L_∞ error in areas Ω_1^ϵ (top), Ω_2^ϵ (middle), and Ω_3^ϵ (bottom). Double logarithmic plot of convergence study with different limiters: nodal (red), shu (green), l2 (blue) and xing (yellow). The grey line indicate linear functions with slope 1, 2 and 3.

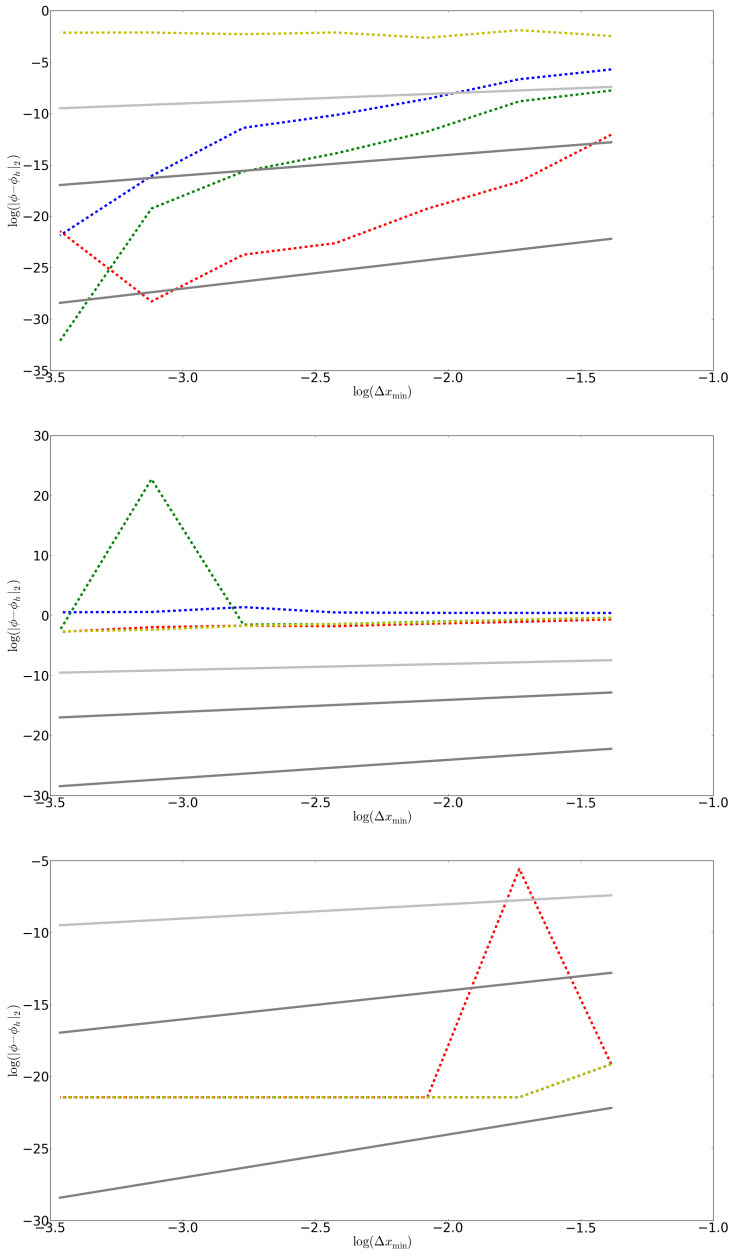

Figure 4.13.: Wet/dry shocktube. Local L_1 error in areas Ω_1^ϵ (top), Ω_2^ϵ (middle), and Ω_3^ϵ (bottom). Double logarithmic plot of convergence study with different limiters: nodal (red), shu (green), l2 (blue) and xing (yellow). The gray line indicate linear functions with slope 1, 2 and 3.

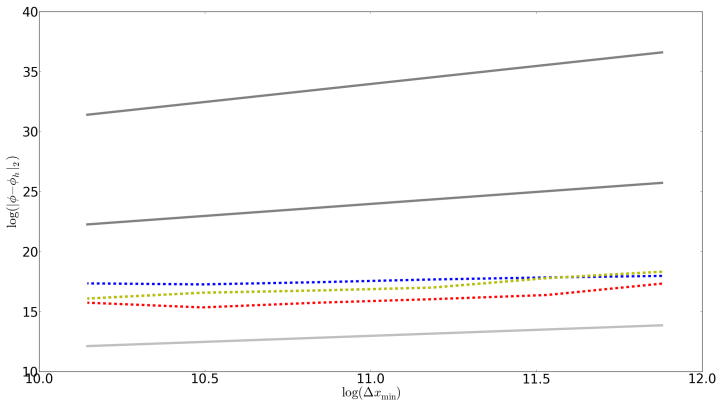

Figure 4.14.: Paraboloid Basin. Double logarithmic plot of global L_2 convergence study with different limiters: global. The gray line indicates a linear function with slopes 1, 2 and 3. Depicted are results for the modal (blue) and nodal (yellow) limiter.

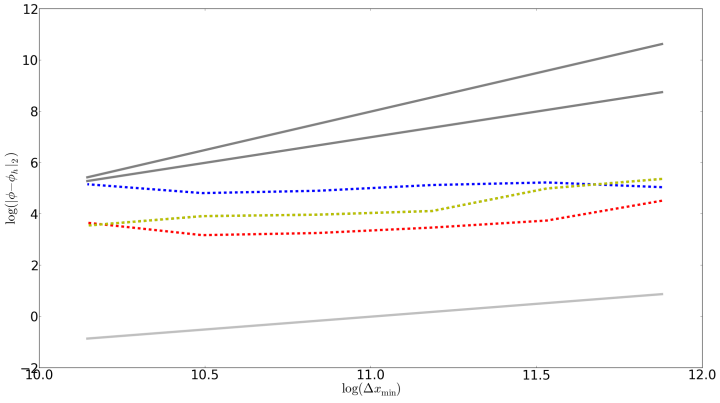

Figure 4.15.: Paraboloid Basin. Double logarithmic plot of global L_∞ convergence study with different limiters. The gray line indicates a linear function with slopes 1, 2 and 3.

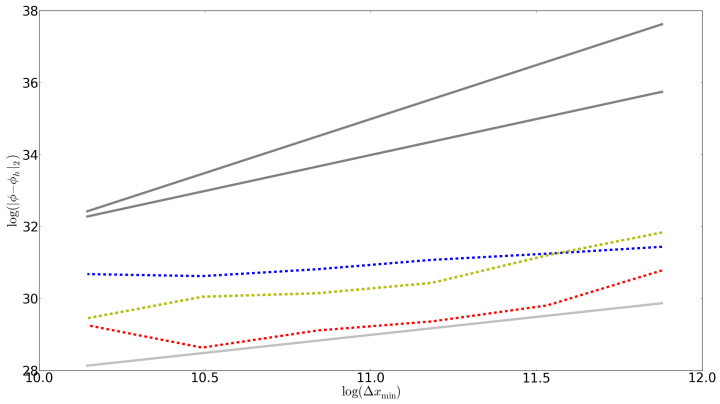

Figure 4.16.: Paraboloid Basin. Double logarithmic plot of global L_1 convergence study with different limiters. The gray line indicates a linear function with slopes 1, 2 and 3.

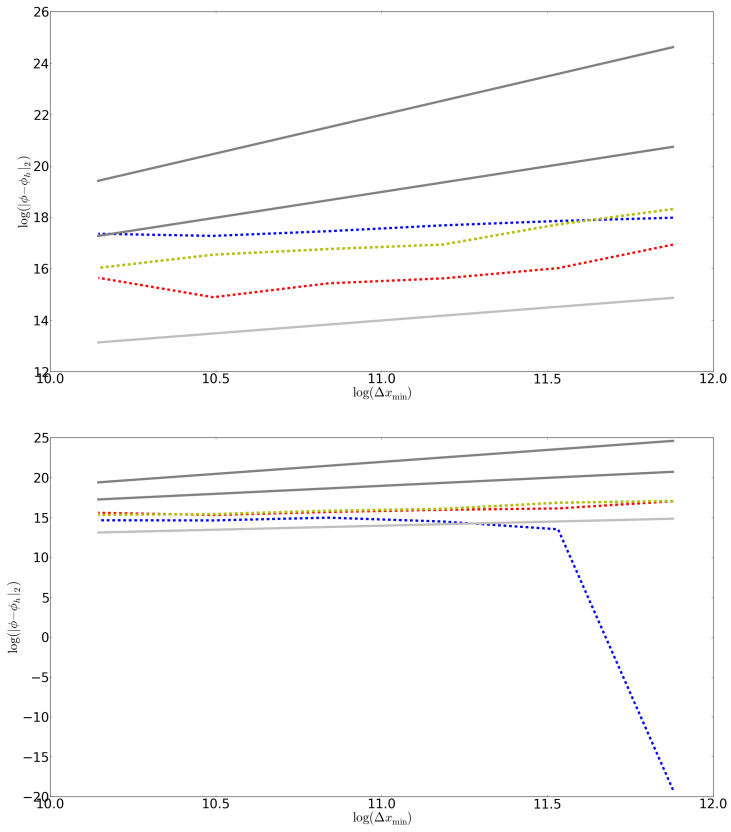

Figure 4.17.: Paraboloid Basin. Double logarithmic plot of L_2 convergence study with different limiters: global. The gray line indicates a linear function with slopes 1, 2 and 3. Depicted are results for the modal (blue) and nodal (yellow) limiter. Ω_1 (top), Ω_2 (bottom).

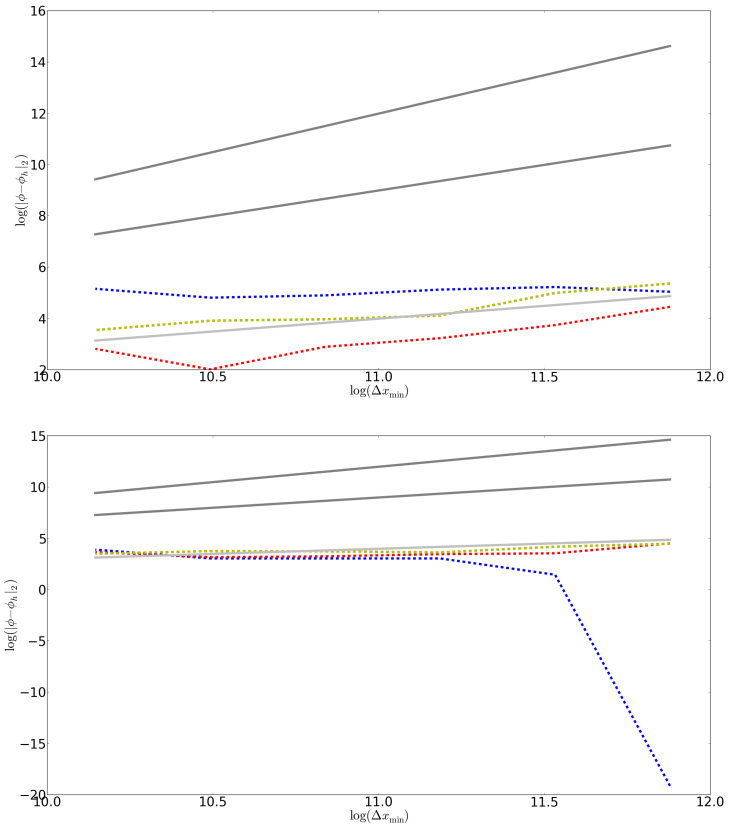

Figure 4.18.: Paraboloid Basin. Double logarithmic plot of L_∞ convergence study with different limiters: global. The gray line indicates a linear function with slopes 1, 2 and 3. Depicted are results for the modal (blue) and nodal(yellow) limiter. Ω_1 (top), Ω_2 (bottom).

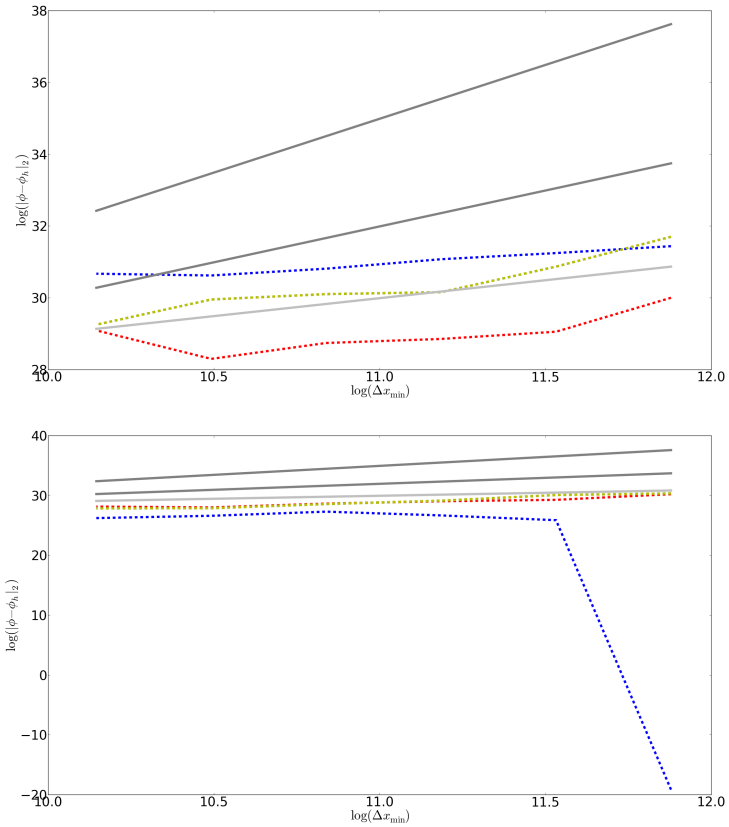

Figure 4.19.: Paraboloid Basin. Double logarithmic plot of L_1 convergence study with different limiters: global. The gray line indicates a linear function with slopes 1, 2 and 3. Depicted are results for the modal (blue) and nodal(yellow) limiter. Ω_1 (top), Ω_2 (bottom).

0.0302 for the wet/dry shocktube and from $\Delta\mathbf{x}_{min} = 287911.62$ to $\Delta\mathbf{x}_{min} = 50896.06$ for the paraboloid 2D basin and measured the maximal error over simulation time. The convergence plots are employing a double-logarithmic scale for reasons of better comparisons.

A plot of global convergence orders can be found in figure 4.9 for the wet/dry shocktube. We observe that all higher than first order limiters yield equal results and that a global order of convergence of 1 could not be exceeded. This might be due to twofold: the jump in the velocity profile and the two not differentiable points at the head and tail of rarefaction. The resulting convergence rates, computed with linear least squares approximations, can be found in table 4.2. The (L2) limiter did not yield converging results. We suggest that the reason therefore is, that the head and tail of the rarefaction are not captured as depicted in figure 4.6, which leads to a non-decreasing error at these local points.

Global convergence rates for the paraboloid 2D basin in the L_2-, L_1- and L_∞-norms are depicted in figures 4.14, 4.16 and 4.15. The different colors refer to different limiters and the gray lines are drawn for comparison. The plots reveal that in the L_2-norm we barely achieve convergence of order 1. For the other norms, the results look a little bit better, but even there the expected order of 2 can not be obtained.

For the partial norms, however, we observe a better behavior. Figures 4.11, 4.12, and 4.13 shows that most of the limiters lead to a better convergence rate locally. Again, the exact numerical convergence rates are shown in table 4.3.

The same seems to be true for the paraboloid basin test. We refer the reader to table 4.3 and figures 4.17, 4.18, and 4.19 for the corresponding plots.

4.4. Riemann Problem

The communication between two neighboring cells in our DG model requires the computation of fluxes across element boundaries (edges). As the derivation of the underlying equations in chapter 3 demonstrated, the fluxes are formally taken into account through the right-hand term of equation (3.4), where $\mathbf{F}^*$ is a numerical approximation. In general, $\mathbf{F}^*$ is a solution to the Riemann problem that was described in chapter 2 (see figure 2.2), which mostly comprises to computation of the wave speeds S_L, S_R, and S_* as well as approximations for the values of the prognostic variables inside the star region: $\mathbf{U}_{L*}, \mathbf{U}_{R*}$. Though our model is two-dimensional, the fluxes over edges are assumed to be one-dimensional, into the direction normal to the edge.

A concise overview over commonly used established Riemann solvers can be found in Toro (2001). They mainly differ in estimates for the computational velocities and the assumed underlying wave patterns. Roughly, they can be differentiated into exact and approximate Riemann solvers. For shallow water equations, there exists an exact Riemann solver whose main drawback is the relatively high computational cost. In addition to the computation of $\mathbf{F}^*$, it can also be used to determine analytical solutions for quasi one-dimensional initial value problems such as the wet/dry shocktube. Approximate Riemann solvers save computational costs by making use of simplifying assumption e.g. on the underlying wave pattern. From our experience, in combination

with a DG model, the quality of the approximate solvers usually suffices and the relative influence of the edge flux even declines with increasing order of approximation.

Probably one of the simplest Riemann solvers is the Rusanov solver defined as

$$\mathbf{F}^* = \frac{1}{2}\left(\mathbf{F}(\mathbf{U}_L) + \mathbf{F}(\mathbf{U}_R) - |\lambda_{max}|(\mathbf{U}_R - \mathbf{U}_L)\right). \tag{4.10}$$

Here, we denote with $\mathbf{U}_L = (\phi_L, \phi_L\mathbf{u}_L)^\top, \mathbf{U}_R = (\phi_R, \phi_R\mathbf{u}_R)^\top$ the vector of balanced quantities on the left and right of the edge and λ_{max} an estimate for the maximal wave speed. For shallow water equations $\lambda_{max} = \langle\mathbf{u}, \mathbf{n}\rangle + \sqrt{\phi}$. The Rusanov solver is computationally inexpensive and contains a relatively high amount of numerical diffusion (see e.g. LeVeque (2002)). On the one hand this stabilizes the numerics, on the other hand it might lead to a smearing of shocks or an error in the detection of contact discontinuities. Another advantage is that it only comprises evaluations of already known quantities, so that additional approximations for the star region are not needed, which is in contrast to the approximate solvers, that we will present in the following. They require the computation of normal and tangential velocities, so that from now on and until the end of this section, we will denote the normal velocity with $u = \langle\mathbf{u}, \mathbf{n}\rangle$ and the tangential velocity with $v = \langle\mathbf{u}, \mathbf{t}\rangle$, where $\mathbf{n}$ and $\mathbf{t}$ are the normal and the tangential vector with respect to the current edge respectively and $\langle\cdot,\cdot\rangle$ is the standard L_2 inner product.

The solver, that was developed by Roe solves a linearized system of equations and uses averages for the characteristic speed a, as well as the fluid height and the velocities of the form

$$\begin{aligned} \phi_{roe} &= \sqrt{\phi_L\phi_R}, \quad a_{roe} = \sqrt{0.5(\phi_L + \phi_R)} \\ u_{roe} &= \frac{u_L\sqrt{\phi_L} + u_R\sqrt{\phi_R}}{\sqrt{\phi_L} + \sqrt{\phi_R}}, \quad v_{roe} = \frac{v_L\sqrt{\phi_L} + v_R\sqrt{\phi_R}}{\sqrt{\phi_L} + \sqrt{\phi_R}}. \end{aligned}$$

With these definitions, averaged eigenvalues $\boldsymbol{\lambda}_k$ and eigenvectors $\mathbf{r}_k$ are determined as

$$\begin{aligned} &\mathbf{r}_1 = (1, \boldsymbol{\lambda}_1, v_{roe})^\top, \qquad \boldsymbol{\lambda} = (u_{roe} - a_{roe}, u_{roe}, u_{roe} + a_{roe})^\top \\ &\mathbf{r}_2 = (0, 0, 1)^\top \\ &\mathbf{r}_3 = (1, \boldsymbol{\lambda}_3, v_{roe})^\top \end{aligned}$$

in order to obtain a decomposition of the flux into different waves with wave strengths

$$\boldsymbol{\alpha} = \left(\frac{1}{2}\frac{\boldsymbol{\lambda}_3\boldsymbol{\delta}_1 - \boldsymbol{\delta}_2}{a_{roe}}, -v_{roe}\boldsymbol{\delta}_1 + \boldsymbol{\delta}_3, \frac{1}{2}\frac{\boldsymbol{\delta}_2 - \boldsymbol{\lambda}_1\boldsymbol{\delta}_1}{a_{roe}}\right)^\top,$$

where $\boldsymbol{\delta} = \mathbf{U}_R - \mathbf{U}_L$ is the jump of the state variables. The corresponding flux is then

$$\mathbf{F}^* = \mathbf{F}(\mathbf{U}_L) + \sum_{\boldsymbol{\lambda}_i \leq 0} \boldsymbol{\alpha}_i \; \boldsymbol{\lambda}_i \; \mathbf{r}_i$$

This solver, however, requires an entropy fix in order to yield a correct representation of sonic rarefaction waves. For the star region, we therefore define values

$$\mathbf{U}_{L*} = \mathbf{U}_L + \boldsymbol{\alpha}_1 \mathbf{r}_1, \qquad \mathbf{U}_{R*} = \mathbf{U}_R - \boldsymbol{\alpha}_3 \mathbf{r}_3,$$

and compare the speeds in the star region from the left and right on both sides of the non-linear waves

$$\begin{aligned} \boldsymbol{\lambda}_{1,L} &= u_L - \sqrt{\phi_L}, \qquad & \boldsymbol{\lambda}_{1,R} &= u_{L*} - \sqrt{\phi_{L*}}, \\ \boldsymbol{\lambda}_{3,L} &= u_R + \sqrt{\phi_R}, \qquad & \boldsymbol{\lambda}_{3,R} &= u_{R*} + \sqrt{\phi_{R*}} \end{aligned}$$

In the presence of a transsonic rarefaction as an either left or right non-linear wave, the jump is split into two smaller jumps of the form

$$\begin{aligned} &\text{If } \boldsymbol{\lambda}_{1,L} < 0 \text{ and } \boldsymbol{\lambda}_{1,R} > 0 \quad \Rightarrow \quad \overline{\lambda} = \boldsymbol{\lambda}_1 \frac{\boldsymbol{\lambda}_{1,R} - \boldsymbol{\lambda}_1}{\boldsymbol{\lambda}_{1,R} - \boldsymbol{\lambda}_{1,L}} \qquad \mathbf{F}^* = \mathbf{F}(\mathbf{U}_L) + \overline{\lambda} \boldsymbol{\alpha}_1 \mathbf{r}_1 \\ &\text{If } \boldsymbol{\lambda}_{3,L} < 0 \text{ and } \boldsymbol{\lambda}_{3,R} > 0 \quad \Rightarrow \quad \overline{\lambda} = \boldsymbol{\lambda}_3 \frac{\boldsymbol{\lambda}_{3,R} - \boldsymbol{\lambda}_3}{\boldsymbol{\lambda}_{3,R} - \boldsymbol{\lambda}_{3,L}} \qquad \mathbf{F}^* = \mathbf{F}(\mathbf{U}_R) - \overline{\lambda} \boldsymbol{\alpha}_3 \mathbf{r}_3 \end{aligned}$$

Other solvers are the HLL and HLLC solvers which assume a two-wave and a three-wave solution pattern respectively. Both require a solution for the star region as was introduced in subsection 2.1.1 in order to determine $\mathbf{F}^*$. The HLL solver is defined as

$$\mathbf{F}^* = \begin{cases} \mathbf{F}(\mathbf{U}_L) & \text{if } S_L \geq 0 \\ \frac{S_R \mathbf{F}(\mathbf{U}_L) - S_L \mathbf{F}(\mathbf{U}_R) + S_R S_L (\mathbf{U}_R - \mathbf{U}_L)}{S_R - S_L} & \text{if } S_L \leq 0 \leq S_R \\ \mathbf{F}(\mathbf{U}_R) & \text{if } S_R \leq 0 \end{cases}$$

where $S_R = u_R + \sqrt{\phi_R}, S_L = u_L - \sqrt{\phi_L}$ are the left and right wave speeds in fully wet domains. The computation of these speeds for partially dry problems is different and will be discussed in the next subsection. In contrast, the HLLC solver is defined as

$$\mathbf{F}^* = \begin{cases} \mathbf{F}(\mathbf{U}_L) & \text{if } S_L \geq 0 \\ \mathbf{F}(\mathbf{U}_L) + S_L(\mathbf{U}_{L*} - \mathbf{U}_L) & \text{if } S_L \geq S_* \\ \mathbf{F}(\mathbf{U}_R) + S_R(\mathbf{U}_{R*} - \mathbf{U}_R) & \text{if } S_* \geq S_R \\ \mathbf{F}(\mathbf{U}_R) & \text{if } S_R \leq 0 \end{cases}$$

with S_* the speed of the contact, and $\mathbf{U}_{*L}, \mathbf{U}_{*R}$ the left and right values of the conserved quantities in the star region (separated by the contact). Common approximation for the values in the star regions and further details can be found in Toro (2001). Problems, however, can arise, when partially wet or dry states occur on the interfaces.

Inundation Modeling with Riemann Solvers

The two major problems that might occur in case of partially dry simulations are addressed in Toro (2001): Artificial bed wetting and conservation errors.

Test	$h_L[m]$	$u_L[m/s]$	$h_R[m]$	$u_R[m/s]$	$x_0[m]$	$t_{out}[s]$
1	1.0	2.5	0.1	0.0	10.0	7.0
2	1.0	-5.0	1.0	5.0	25.0	2.5
3	1.0	0.0	0.0	0.0	20.0	4.0
4	0.0	0.0	1.0	0.0	30.0	4.0
5	0.1	-3.0	0.1	3.0	25.0	5.0

Table 4.4.: Initial Data for test problems with exact solution

There are estimates to determine the wave speeds for wet/dry problems. The resulting waves, however, are very fast. Artificial bed wetting is used to slow down the fastest wave of the system and relax the resulting CFL condition. Formally the dry areas are wetted with a small amount $\alpha_{wet} \in \mathbb{R}$ of water, $h = \max(h, \alpha_{wet})$, which changes the physics of the problem. An artificial shock front will develop that travels with a speed smaller than the actual wet/dry front. The more the actually dry bed is wetted, the larger is the underestimation of the wave speed.

Conservation errors are caused by the computation of the speed $\mathbf{u} = \frac{\phi \mathbf{u}}{\phi}$ within the model. This is a division of two possibly very small quantities, which is a great source of numerical errors, because the problem is bad conditioned. We remark, that during the computation a division by zero has in any case to be avoided.

For more realistic problems, more physics such as viscosities or frictions are involved, which make it impossible to determine the errors as possibly no analytical solution would be present.

4.4.1. Testcases with Analytical Solutions

To study the effects of the Riemann solver on the accuracy and computational cost of the solution, we simulated the test suite of quasi one-dimensional wet/dry problems with analytical solutions, that was introduced in Toro (2001). For all solvers, we employed a bed wetting as described in the previous section with a bed wetting coefficient of $\alpha_{wet} = 10^{-7}$.

The test problems that we will consider can be described as the following

Testcase 7. Toro's Wet/Dry Shocktubes
We consider the homogeneous shallow water equations in a channel $\Omega = [0, 50] \times [0, 10]$ with discontinuous initial conditions of the form

$$h(\mathbf{x}, 0) = \begin{cases} h_L & \text{for } x \leq x_0 \\ h_R & \text{otherwise.} \end{cases}, \qquad u(\mathbf{x}, 0) = \begin{cases} u_L & \text{for } x \leq x_0 \\ u_R & \text{otherwise.} \end{cases}$$
$$v(\mathbf{x}, 0) = 0$$

for times $t \in [0, t_{out}]$. The values for h_L, h_R, u_L, u_R, x_0, and t_{out} are specified in table 4.4.

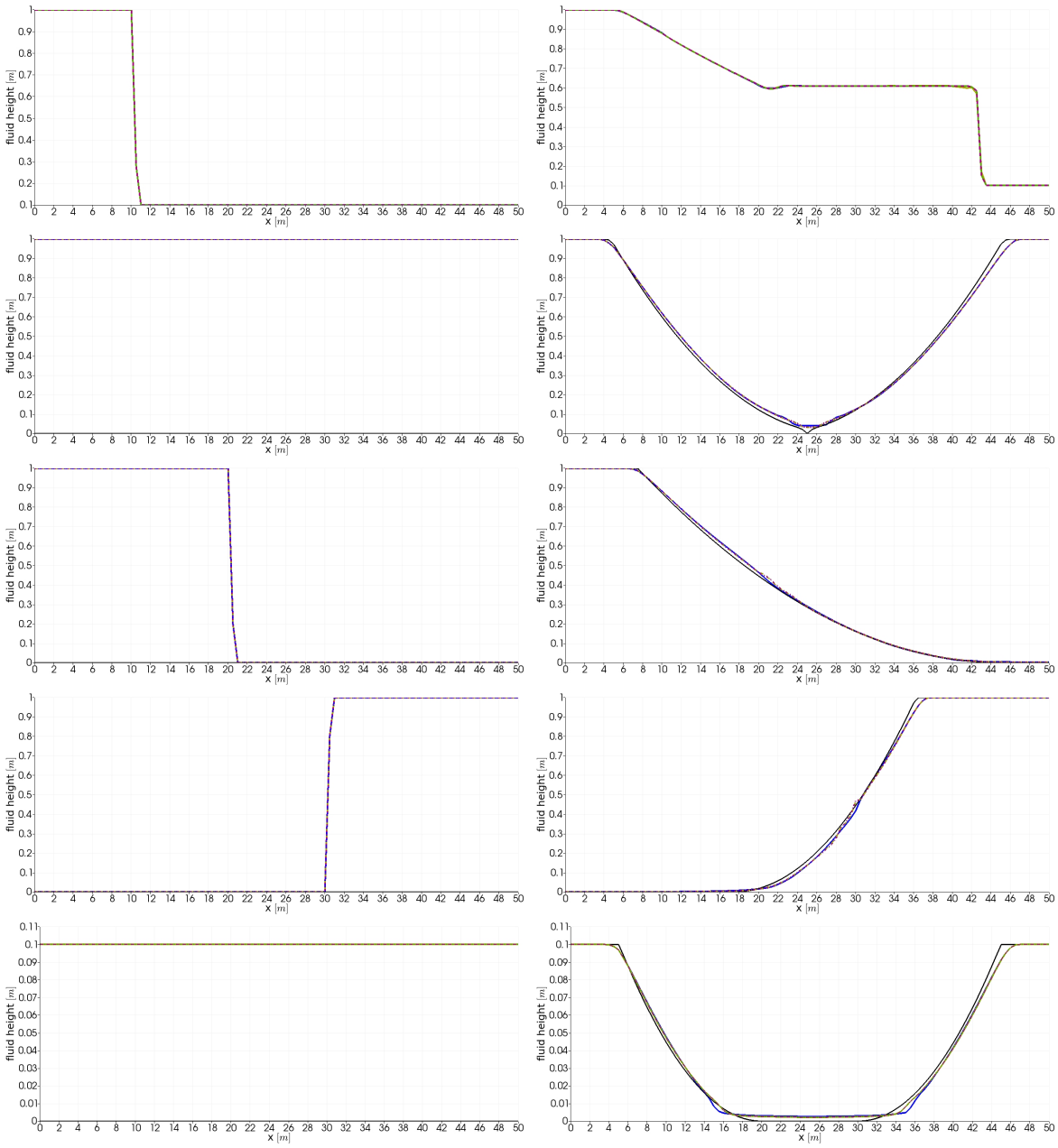

Figure 4.20.: Toro's Wet/Dry Shocktubes. Depicted are the numerical solution with Rusanov (blue), Roe (yellow), HLL (green) and HLLC (red) Riemann solvers and the analytical solution (black) at $t = 0$ and $t = t_{out}$. The kth row corresponds to the kth test.

Test	RUSANOV	ROE	HLL	HLLC
1	0.2562E+3	0.2600E+3	0.2572E+3	0.2535E+3
2	0.4750E+2	0.4700E+2	0.4725E+2	0.4675E+2
3	0.7375E+2	0.7375E+2	–	0.7000E+2
4	0.7500E+2	0.7450E+2	–	0.7225E+2
5	0.9375E+2	0.9500E+2	0.9250E+2	0.9050E+2

Table 4.5.: Runtime of Toro's Wet/Dry Shocktubes with different Riemann solvers.

The results obtained with our DG model after t_{out} s are depicted in figure 4.20 (right column). We observe that for all five tests we obtain similar results with all Riemann solvers and remark, however, that we were not able to obtain robust results with the HLL solver for the test cases 3 and 4. As is shown in table 4.5 although the computational costs of all solvers are different the overall cost of the model run is not significantly affected. In conclusion, we have demonstrated, that as long as care is taken of bed wetting and avoidance of division by zero, any approximate Riemann solver can be used in combination with our DG model to simulate wetting and drying problems.

4.5. Wellbalanced Schemes

The preservation of steady states, or wellbalancing, is one crucial aspect of the modeling of flooding and drying. If a model is not wellbalanced, then in quasi steady regimes, small perturbations can be amplified and a pure numerical storm can develop. The main difficulty is to balance pressure gradients, i.e. the terms ∇b in the source term and $\nabla\phi$ in the flux term of (3.2), on a discrete level in order to not introduce artificial waves. Limiters are therefore also required to not modify this balance. A simplified set of associated partially and fully wet test problems is given by

Testcase 8. *General Lake at rest*
We assume a bounded domain Ω with a polynomial bathymetry $b(\mathbf{x}) = p(\mathbf{x}) \in \mathcal{P}_s$, $s \in \mathbb{N}$, a parameter $H \in \mathbb{R}$, and initial conditions:

$$\begin{aligned} h(\mathbf{x},t) &= \max\Big(0, H - b(\mathbf{x})\Big) \\ u(\mathbf{x},t) &= \mathbf{0} \qquad\qquad \text{for all } t \in [0,T]. \end{aligned}$$

In the current literature, wellbalancing has mainly been studied in the context of finite volume methods. For a DG framework, a technique is proposed in Bastian et al. (2012); Dedner et al. (2011) which comprises to rewrite the source term for the bathymetry and treat the latter as an additional unknown. We will introduce a simpler and computationally less expensive scheme. For the fully wet cases, it will only depend on the use of the strong form of the equations and the succession of discretization and differentiation of the flux tensor. This idea is motivated by the observation that, on a discrete level,

there is a difference between the two discretizations

$$\mathrm{div}\mathbf{F}(\mathbf{U})(\mathbf{x},t) = \mathrm{div}\Big(\sum_k \big(\mathbf{F}(\mathbf{U}_k(t))\big)\psi_k(\mathbf{x})\Big) \tag{4.11a}$$

$$\mathrm{div}\mathbf{F}(\mathbf{U})(\mathbf{x},t) = \sum_k \mathbf{F}_{div}(\mathbf{U}_k(t))\psi_k(\mathbf{x}). \tag{4.11b}$$

where $\mathbf{F}_{div}(\mathbf{U}_k)$ and $\big(\mathbf{F}(\mathbf{U}_k)\big)$ are the nodal coefficients of the divergence of the flux and the flux respectively. From now on, we will drop the time variable to improve the readability. The difference is that in (4.11a) we first discretize and then differentiate and in (4.11b) we do it the other way around.

For demonstration purposes, we transform the integral equation (3.5):

$$\int_{\Omega_e} \left(\frac{\partial \mathbf{U}}{\partial t} + \nabla\cdot\mathbf{F}(\mathbf{U}) - S(\mathbf{U})\right)\ \psi_j(\mathbf{x})\ d\mathbf{x} = -\int_{\partial\Omega_e} \Big(\mathbf{F}^* - \mathbf{F}(\mathbf{U})\Big)\ \psi_j(\mathbf{x})\cdot\mathbf{n}\ dS$$

into matrix vector representation. With the finite polynomial representation for the prognostic variables $\mathbf{U} = \sum_k \mathbf{U}_k\psi_k$ and for the flux as described in (4.11), we obtain for every element Ω_e, time $t \in \mathbb{R}$ and $j = 1,..,m$ for the left side of (3.5):

$$\sum_{i=1}^m \int_{\Omega_e} \psi_i(\mathbf{x})\psi_j(\mathbf{x})\ d\mathbf{x} \left(\frac{\partial \mathbf{U}_i}{\partial t} - S(\mathbf{U}_i)\right) + \sum_{i=1}^m \int_{\Omega_e} \psi_i(\mathbf{x})\ \mathrm{div}\mathbf{F}(\mathbf{U}_i)\ \psi_j(\mathbf{x})\ d\mathbf{x} \tag{4.12a}$$

$$\sum_{i=1}^m \int_{\Omega_e} \psi_i(\mathbf{x})\psi_j(\mathbf{x})\ d\mathbf{x} \left(\frac{\partial \mathbf{U}_i}{\partial t} + \mathbf{F}_{div}(\mathbf{U}_i) - S(\mathbf{U}_i)\right). \tag{4.12b}$$

Note, that (4.12a) corresponds to the discretization (4.11a) and (4.12b) to (4.11b). For the edge-based terms on the right of (3.5) we have for both cases

$$\sum_{i=1}^m \int_{\partial\Omega_e} \psi_i(\mathbf{x})\psi_j(\mathbf{x}) dS\ \Big(\mathbf{F}^* - \big(\mathbf{F}(\mathbf{U}_i)\big)\Big)\ \cdot\mathbf{n}. \tag{4.13}$$

We solve the integrals numerically using an interpolatory Gauss quadrature with corresponding Gauss Lobatto points $\boldsymbol{\xi}_i, i = 1,\dots m\prime$ and weights $\boldsymbol{\omega}_i, i = 1,\dots m\prime$. To simplify the equations, we introduce the notations

$$\mathbf{M} := \Big(\int_{\Omega_e} \psi_i(\mathbf{x})\psi_j(\mathbf{x})d\mathbf{x}\Big)_{1\le,i,j\le m}, \quad \mathbf{M}^E := \Big(\int_{\partial\Omega_e} \psi_i(\mathbf{x})\psi_j(\mathbf{x})dS\Big)_{1\le,i,j\le m}, \quad \boldsymbol{\Omega} := \mathrm{diag}(\boldsymbol{\omega})$$

$$\boldsymbol{\Psi} := \Big(\psi_i(\boldsymbol{\xi}_j)\Big)_{\substack{1\le i\le m\\ 1\le j\le m\prime}}, \quad \mathbf{D}^x := \Big(\frac{\partial}{\partial x}\psi_i(\boldsymbol{\xi}_j)\Big)_{\substack{1\le i\le m\\ 1\le j\le m\prime}}, \quad \mathbf{D}^y := \Big(\frac{\partial}{\partial x}\psi_i(\boldsymbol{\xi}_j)\Big)_{\substack{1\le i\le m\\ 1\le j\le m\prime}}.$$

where we will call $\mathbf{M}$ the mass matrix, $\mathbf{M}^E$ the mass matrix corresponding to the edges, $\boldsymbol{\Omega}$ is a diagonal matrix containing the Gauss quadrature weights, $\boldsymbol{\Psi}$ the interpolation from Lagrange points $\mathbf{x}_i$ to Gauss quadrature points $\boldsymbol{\xi}_i$ and the matrices $\mathbf{D}^q, q \in \{x, y\}$ contain the derivatives of the basis functions in directions x and y evaluated at the interpolation points. We use the notation to combine (4.12a) and (4.12b)

with (4.13). This yields

$$\mathbf{M}\frac{\partial \mathbf{U}}{\partial t} + \mathbf{\Psi\Omega}\big(\mathbf{D}^x + \mathbf{D}^y\big)\mathbf{\Psi}\mathbf{F}(\mathbf{U}) - \mathbf{M}S(\mathbf{U}) = \mathbf{M}^E\big(\mathbf{F}(\mathbf{U}) - \mathbf{F}^*\big)\cdot\mathbf{n} \tag{4.14a}$$

$$\mathbf{M}\frac{\partial \mathbf{U}}{\partial t} + \mathbf{M}\mathbf{F}_{div}(\mathbf{U}) - \mathbf{M}S(\mathbf{U}) = \mathbf{M}^E\big(\mathbf{F}(\mathbf{U}) - \mathbf{F}^*\big)\cdot\mathbf{n} \tag{4.14b}$$

For the steady state problem we are interested in the left sides only; the right sides can be neglected since their contribution is always zero. That means, that the balances we want to maintain for every element Ω_e and indices $i = 1, \dots m$ are

$$\big(\mathbf{D}^x + \mathbf{D}^y\big)\mathbf{\Psi}\mathbf{F}(\mathbf{U}_i) = \mathbf{\Psi}S(\mathbf{U}_i) \tag{4.15a}$$

$$\mathbf{F}_{div}(\mathbf{U}_i) = S(\mathbf{U}_i) \tag{4.15b}$$

Now the difference between the two formulations is obvious. The equation (4.15b) is fulfilled as long as the flux divergence and the source evaluated at the Lagrange points are balanced. This will always be the case in a fully wet regime with zero initial velocity since

$$\mathbf{F}_{div}(\mathbf{U}_i) = (0, \phi_i\ \nabla\phi_i)^\top = (0, -\phi_i\ \nabla b_i)^\top = S(\mathbf{U}_i) \qquad \forall i = 1, \dots m$$

always holds true.

That the balance in (4.15a) is not always fulfilled, can be shown with a counter example. Therefore, we consider the reference element with vertices $(-1,-1), (1,-1), (-1,1)$ and linear Lagrange functions. In this case the matrices containing element derivatives take the form

$$\mathbf{D}^x = \begin{bmatrix} -0.5 & 0.5 & 0.0 \\ -0.5 & 0.5 & 0.0 \\ -0.5 & 0.5 & 0.0 \end{bmatrix}, \qquad \mathbf{D}^y = \begin{bmatrix} -0.5 & 0.0 & 0.5 \\ -0.5 & 0.0 & 0.5 \\ -0.5 & 0.0 & 0.5 \end{bmatrix}$$

This leads to

$$(4.15a) \Leftrightarrow \begin{cases} 0.5\Big(0.5\big(\phi(\boldsymbol{\xi}_2)^2 - \phi(\boldsymbol{\xi}_1)^2\big)\Big) & = 0.5\phi(\boldsymbol{\xi}_k)\big(b(\boldsymbol{\xi}_2) - b(\boldsymbol{\xi}_1)\big) \\ 0.5\Big(0.5\big(\phi(\boldsymbol{\xi}_3)^2 - \phi(\boldsymbol{\xi}_1)^2\big)\Big) & = 0.5\phi(\boldsymbol{\xi}_k)\big(b(\boldsymbol{\xi}_3) - b(\boldsymbol{\xi}_1)\big) \end{cases} \quad \forall k = 1, \dots m\prime$$

which is clearly only fulfilled for constant functions.

Wellbalancing for Partially Dry Cells

The presentation in this chapter has not yet taken into account that cells may be partially dry. In this case, we propose a correction as is shown in figure 4.21. This comprises to map physically dry but numerically wet cells to zero using the following scheme for every element

- Detect if cell Ω_e is partially dry, i.e. $\min_i \phi_{e,h}(\mathbf{x}_i) = 0$ and $\max_i \phi_{e,h}(\mathbf{x}_i) > 0$,

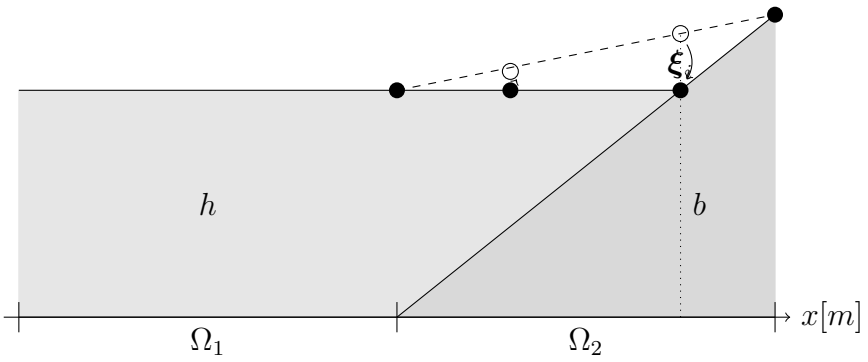

Figure 4.21.: Schematic of fix, 1D projection of cell to one edge, true wet/dry interface (dotted line), water surface computed with nodal approach (dashed line)

- IF yes: Check if bathymetry b is non-constant on Ω_e and momentum $\max_{\Omega_e}(\phi\mathbf{u})$ is sufficiently small,
- IF yes: Determine new values for $\phi_{new}(\boldsymbol{\xi}_i) = \max(0, \phi_{max} - b(\boldsymbol{\xi}_i))$, with ϕ_{max} is the maximum fluid height in the wet part of the domain and correct the the gradients $\nabla\phi_{new} = \nabla b$.

4.5.1. Numerical Tests of Robustness and Accuracy

We tested our proposed scheme with respect to the capability of handling wet and dry states as well as robustness, which was tested by introducing a small perturbation. Therefore, we computed the general lake at rest test with reflecting boundaries and the following data

i. $\Omega = [0,1]^2, H = 0.2$, and the bathymetry is defined as $b(\mathbf{x}) = \max(0, 0.25 - 5((x - 0.5)^2 + (y - 0.5)^2))$ for $t \in [0, 10]$.

ii. $\Omega = [0,30]^2, H = 10$, and the bathymetry is defined as $b(\mathbf{x}) = 8 \cdot \exp(-0.25((x - 15)^2 + (y - 15)^2))$ for $t \in [0, 1]$. The perturbation has the form $h(\mathbf{x}, 0) = 10.1$ for $x \in [10, 12]$.

iii. $\Omega = [0,1]^2, H = 0.2$, and the bathymetry is defined as $b(\mathbf{x}) = \max(0, 0.25 - 5((x - 0.5)^2 + (y - 0.5)^2))$ for $t \in [0, 0.75]$. The perturbation has the form $h(\mathbf{x}, 0) = 0.2 + \max(0, 0.05 - 30((x - 0.2)^2 + (y - 0.5)^2))$

For the partially dry case i. we have run a simulation with $\Delta\mathbf{x}_{min} = 0.022$, 4096 elements and a Runge Kutta RK22 scheme with a timestep of $\Delta t = 0.0005$. We employed the vertex-based limiter that does not destroy the well-balanced property. The results can be seen from figure 4.22. Since the solution is symmetric, we show a plot over the line $y = 0.5$. The left display shows the fluid height every 2.5s and the right display the corresponding magnitude of velocities. Clearly the dry part of the

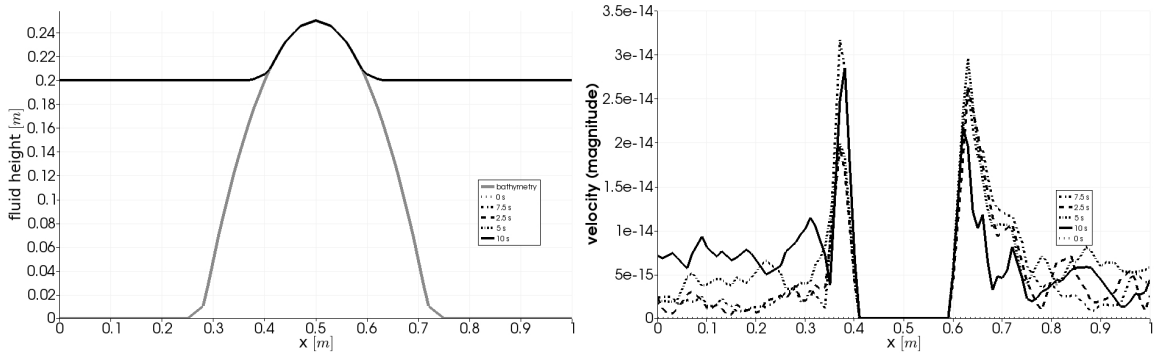

Figure 4.22.: Wet/dry lake at rest (i). Plot over line with y=0.5. Depicted are the fluid height (left) and the magnitude of velocity (right). The plots show the results after 0s (dotted line), 2.5s (dashed line), 5s (dot dot dashed line), 7.5s(dot dashed line) and 10s (solid line).

domain has zero velocities and the artificial velocities stay within a range of machine precision as intended.

The perturbation test ii. was run with $\Delta \mathbf{x}_{min} = 0.022$, 4096 elements and a Runge Kutta RK22 scheme with a timestep of $\Delta t = 0.002$ and, again, the well-balancing preserving vertex-based limiter. The results are depicted in figure 4.23. We observe that there are two waves traveling in opposite directions from which the right one is affected by the bathymetry. It is steeper on that side and also the speed is decreased.

Finally, we ran the partially dry perturbation test iii. with 16384 elements and a spatial resolution of $\Delta \mathbf{x}_{min} = 0.011$ and a Runge Kutta RK22 scheme with a timestep of $\Delta t = 0.0005$. We observe that the perturbation in combination with the wet/dry front of the island do not affect the stability of the simulation. The simulation results are depicted in figure 4.24. We see, that the initial bump travels around the island and the corresponding waves meet after passing the island.

4.6. Concluding Remarks

In this chapter we studied the effects of different numerical approximations within the DG model, namely slope limiters, Riemann solvers as well as a technique to achieve wellbalancedness, on the overall accuracy of the simulation result.

In section 4.3 we presented a general characterization of slope limiters and showed examples for an edge-based as well as for a vertex-based approach and how they fit into this general theoretical framework. We imposed requirements on limiters following Cockburn and Shu (1998) in order to study the influence on the overall accuracy. The main result conveys that, in the mean, the order of accuracy is retained as long as the total variation is bounded in the mean. Moreover, we compared the limiters with respect to computational cost, preservation of non-negativity and well-balancedness and were able to make a recommendation regarding limiters for flooding and drying

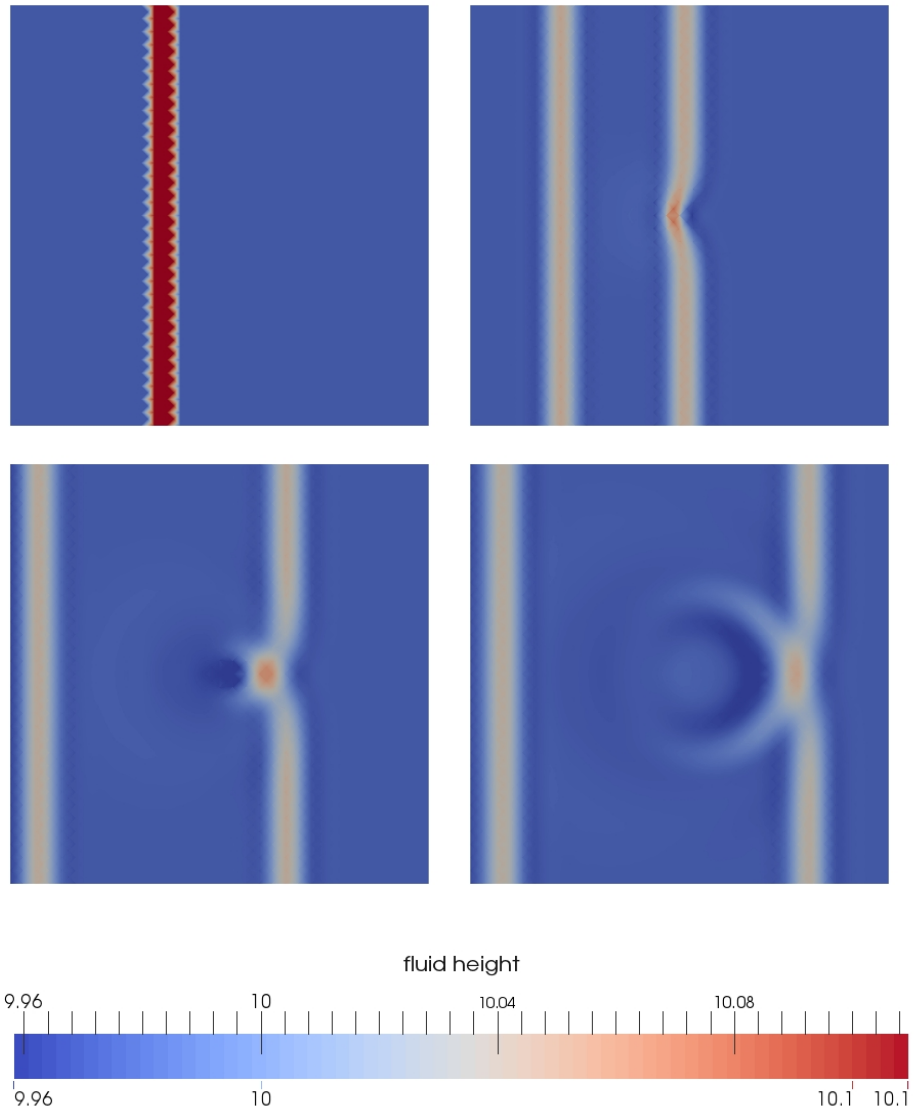

Figure 4.23.: Lake at rest with a perturbation (ii). Plots of the fluid height every 0.04s from top left to bottom right.

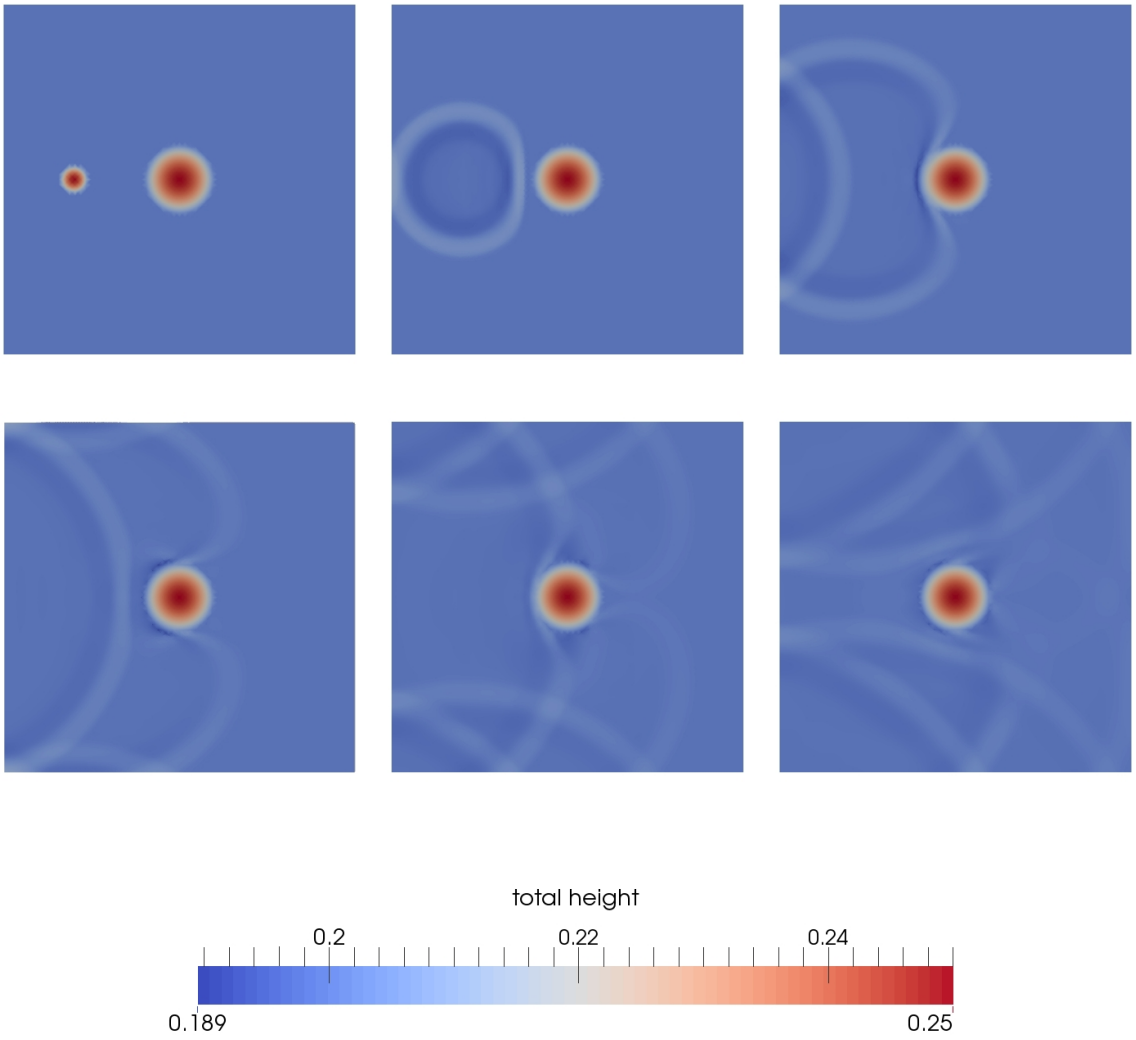

Figure 4.24.: Wet/Dry lake at rest with a perturbation (iii). Plots of the fluid height every 0.125s from top left to bottom right.

problems. From a theoretical point of view, the extended edge-based limiter (EB+) as well as the vertex-based limiter (VB) were preferable because of their high-order accuracy, wellbalancedness and non-negativity preservation, which is crucial for inundation modeling. In terms of computational cost, however, the simple L_2-limiter is the least expensive, though it does not comprise all of the just mentioned desired properties.

Our numerical study with linear basis functions included two wet/dry test problems with analytical solutions for comparison: A quasi one-dimensional wet/dry shocktube and a paraboloid 2D basin as introduced in section 4.3.3. We tested a set of four different limiters in combination with our model and observed that all of them yielded acceptable results throughout this part of the study. With respect to conservation properties, we obtained satisfactory results. A convergence study revealed, however, that globally the theoretical convergence order 2 for linear basis functions could not be achieved. To study the influence of the not regular enough points of the solution on the global convergence order, we introduced error measures that only considered the smooth parts of the solution. A corresponding convergence study showed an improvement of the order of convergence towards the theoretically expected convergence rate. In all, the (VB) limiter performed best in combination with our DG model, because of its robustness, the easy way to tune the inherent parameter and the conservation properties. Despite its theoretically higher computational cost, the (EB) and (EB+) limiters were slower in terms of measured CPU run time. In all, we recommend to use the vertex-based limiter for simulations of wetting and drying with our DG model.

A short introduction of Riemann solvers was given in section 4.4. We showed different Riemann solvers and explained the different wave pattern and wave speeds, that are assumed as well as their impact on the simulation results. We discussed known problems with Riemann solvers for flooding and drying and stated techniques to overcome them such as bed wetting. Finally, we tested a test suite of quasi one-dimensional wet/dry problems with the implemented approximate Riemann solvers. As a result, we confirmed that in terms of accuracy and computational cost, all of them lead to comparable results.

In section 4.5 we presented a computationally inexpensive way to preserve steady state solutions in completely wet domains with the strong form of the shallow water equations. For partially dry domains, we introduced an algorithm to prevent artificial waves at the wet/dry interface from forming. Numerical tests of different configurations of the general wet/dry lake at rest, with and without perturbations, showed that the implementation is robust and capable of ensuring the balance of the pressure gradients up to machine precision.

5. High-Order Nodal Discontinuous Galerkin Inundation Modeling

5.1. Abstract

A nodal discontinuous Galerkin (DG) model in combination with a vertex-based slope limiter simulates flooding and drying scenarios. To increase the accuracy of the method, we study the viability of higher than second order basis functions and identify stability problems that occur using a nodal Lagrange basis. Therefore, we develop a new quasi-nodal approach based on Bernstein polynomials to stabilize the computations. The subsequent numerical study demonstrates an improvement of conservation properties and local convergence rates with the new method.

5.2. Introduction

The overall accuracy of our DG model is determined by the degree n of the employed polynomials. If we neglect the effects of the approximations, that are made and have been the subject of the previous chapter 4, we can decrease the numerical error either by grid refinement or by increasing the approximation order n as is depicted in sketch 5.1. In general, a refinement of the mesh can be advantageous for representing discontinuities in the numerical solution while higher order basis functions are favorable for representing local extrema inside the elements and, primarily, yield a higher convergence rate.

To date, the known operational models for storm surges work with at most second order accuracy in the partially dry regions if not throughout (see for example Westerink et al. (2008)) although an at least local improvement of convergence could theoretically be achieved with higher than second order basis functions.

The use of lower than second order basis functions has several advantages. It simplifies the implementation, because first and second order nodal models solely require the values at the vertices or the mean values. This prevents difficulties of numbering the degrees of freedom and also permits the use of easy-to-compute criteria for deciding if a cell is wet or dry, which is an essential aspect of some of the present inundation models (e.g. Bunya et al. (2009)) as we have seen in chapter 4. To the authors' knowledge a formulation of such criteria for polynomials of degree higher than $n = 1$ is not known. Another reason for the little attention that has been paid to higher than second order basis functions for inundation problems is the lack of global smoothness of their solutions, e.g. at the interfaces between wet and dry states, the solution might not be differentiable. Because of that, the global convergence rate of the model will be limited by these critical, non-smooth, points. We remark, however, that for the

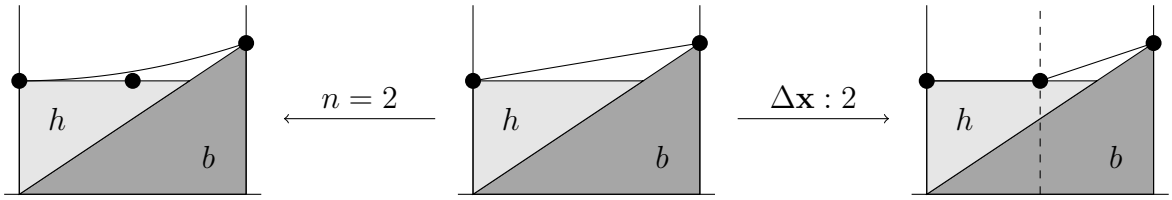

Figure 5.1.: Schematic h- and p-refined solution of total fluid height with Bernstein polynomials in a wet/dry cell. Projection onto one-dimensional domain.

problems of interest the non-smooth points are strongly localized and that the solution is mainly smooth. Furthermore, depending on the slope limiting technique, the nodal Lagrange basis, which is used because it simplifies the computations and has good approximation properties, is not straightforwardly usable, because it develops stability problems for wetting and drying simulations, that we will elaborate later.

By the above-mentioned reasons, an investigation of the viability of third-order basis functions for simulating flooding and drying with a nodal DG model in combination with the vertex-based limiter (VB) has not been performed yet. Albeit, higher-order basis functions locally better approximate smooth extrema and theoretically give a better order of convergence in smooth regions. They allow for a larger spatial resolution, which, lessens the effect of heuristic refinement indicators for the adaptive mesh on the numerical solution. Finally, the impact of Riemann solvers, and errors that are accompanied by it, decrease as relatively fewer degrees of freedom are involved in the edge quadrature.

In the following section we investigate the viability of higher-order basis functions in a nodal DG model for the simulation of flooding and drying. We will demonstrate how stability issues occur, when using third-order nodal Lagrange basis functions in section 5.3 and suggest a modification to stabilize the model. In section 5.4 we introduce a novel approach using monotone Bernstein polynomials as basis functions which do not suffer from stability problems. Finally, we show an improvement of local convergence rates and conservation properties for simplified inundation problems with analytical solutions in section 5.5.

5.3. Non-negativity Violation of the Lagrange Approach

The main reason for stability problems with higher than second order Lagrange basis functions is their lack of monotonicity together with the (VB) limiter, that has so far only been applied to polynomials of degree $p = 2$ for convection equations (see Kuzmin (2013b)). In close to dry regions, negative values of the numerical approximation $\phi_{e,h}$ on an element Ω_e can not always be prevented, because the slope limiter takes into account the values on interpolation points and mean values and modifies nodal values only. This can lead to negative values off the set of interpolation points. We observe, that the transformation from interpolation points $\mathbf{x}_i$ for $1 \leq i \leq m$ to the Gauss Lobatto quadrature points $\boldsymbol{\xi}_i$ for $1 \leq i \leq m\prime$, where in general $m\prime \neq m$, can lead to negative

function values on the quadrature points in near dry areas and even negative values of integrals such as the integral of the fluid height.

To illustrate the problem, we consider an arbitrary triangle $\Omega_e \subset \Omega$ with quadratic Lagrange basis functions, i.e. $p = 2$. The number of degrees of freedom is $m = 6$. For simplicity, we choose an equal number of quadrature points inside Ω_e, i.e. $m\prime = 6$. We assume that on only one degree of freedom the fluid height is positive and zero otherwise, which is equivalent to the existence of an index k with

$$\phi_{e,h}(\mathbf{x}_k) > 0 \text{ and } \phi_{e,h}(\mathbf{x}_l) = 0 \quad \text{for } l = 1, \ldots, m \text{ with } l \neq k.$$

The transformation from interpolation onto quadrature points can be performed by matrix vector multiplication. Therefore, we define $\mathbf{\Psi}_{ij} = \psi_i(\boldsymbol{\xi}_j)$, which is the Lagrange basis evaluated at the quadrature points and has the structure

$$\mathbf{\Psi}^\top = \begin{bmatrix} 0.5176 & -0.0748 & -0.0748 & 0.2992 & 0.2992 & 0.0335 \\ -0.0748 & 0.5176 & -0.0748 & 0.2992 & 0.0335 & 0.2992 \\ -0.0748 & -0.0748 & 0.5176 & 0.0335 & 0.2992 & 0.2992 \\ -0.0847 & -0.0482 & -0.0482 & 0.1928 & 0.1928 & 0.7955 \\ -0.0482 & -0.0847 & -0.0482 & 0.1928 & 0.7955 & 0.1928 \\ -0.0482 & -0.0482 & -0.0847 & 0.7955 & 0.1928 & 0.1928 \end{bmatrix}$$

This matrix is constant and valid for all elements and can be precomputed for the simulation. If we now assume that $\phi_{e,h}(\mathbf{x}_1) > 0$ and the values on the remaining interpolation points are zero, i.e. $\phi_{e,h}(\mathbf{x}_k) = 0$ for $k = 2, .., m$, we obtain a negative value on quadrature points by using the transformation

$$\mathbf{\Psi} \cdot (\phi_{e,h}(\mathbf{x}_l))_{1 \leq l \leq m} = (\phi_{e,h}(\boldsymbol{\xi}_l))_{1 \leq l \leq m\prime}$$

since for example $\phi_{e,h}(\boldsymbol{\xi}_2) < 0$ follows after multiplication which contradicts our demand on positivity of the water height throughout the domain.

To ensure non-negativity of the numerical solution on all interpolation and quadrature points and hence stability of the computations we investigate two different strategies:

- One strategy consists of manually setting $\phi_{e,h}(\boldsymbol{\xi}_k) = \max(\phi_{e,h}(\boldsymbol{\xi}_k), 0)$ for all $k = 1, .., m\prime$. This cut-off however will have an effect on the conservation properties, whose severity is subject to the following investigation. It will also lead to errors in the computation of fluxes and therewith have an effect on the wet/dry front as can be seen in figure 5.5.

- The other strategy is to change the basis functions. We propose the use of Bernstein polynomials instead of Lagrange polynomials. Thus, we give up the cardinality of the basis in order to gain monotonicity preservation (see e.g. Farin (1986)). Although the Bernstein polynomials are not cardinal, they allow for a quasi-nodal approach which will be the subject of the next section.

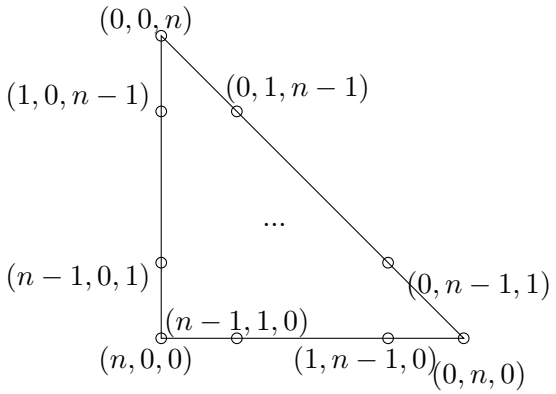

Figure 5.2.: Bernstein polynomials. Schematic of multi-indices that correspond to Lagrange points.

5.4. A Quasi-Nodal Bernstein Approach

Bernstein polynomials are commonly used functions in computer-aided design, because they preserve a lot of shape properties. In the context of DG methods they have so far only been explored to define additional diffusion terms in a Lagrangian DG model (see Loubére et al. (2004)) and for mappings between linear and curved elements in Hindenlang et al. (2010). They are constructed using Beziér patches as described in Farin (1986). With these monotone functions as test and basis functions we reduce numerical oscillations within the elements.

First of all, we introduce barycentric coordinates $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\mathbf{x}) \in \mathbb{R}^3$ of a point $\mathbf{x} \in \Omega$. They can be easily obtained through the solution of a linear system when the vertices of the triangle Ω are known. The coordinates are bounded by $0 \leq \boldsymbol{\sigma}_k \leq 1$ for $k = 1, 2, 3$ and $|\boldsymbol{\sigma}| = \sum_k \boldsymbol{\sigma}_k = 1$. The Bernstein polynomial $B^n_{\boldsymbol{\lambda}}$ of degree n evaluated at a point $\mathbf{x}$ is defined as

$$B^n_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}(\mathbf{x})) = \frac{n!}{\boldsymbol{\lambda}_1!\boldsymbol{\lambda}_2!\boldsymbol{\lambda}_3!}\boldsymbol{\sigma}_1^{\boldsymbol{\lambda}_1}\boldsymbol{\sigma}_2^{\boldsymbol{\lambda}_2}\boldsymbol{\sigma}_3^{\boldsymbol{\lambda}_3}$$

using a multi-index $\boldsymbol{\lambda} \in \mathbb{N}^3$ that is determined through a linear mapping $\mathcal{L}$ that maps the Lagrange interpolation points onto multi-indices, $\mathcal{L} : \mathbf{x}_i \mapsto \boldsymbol{\lambda}$. Therefore, the polynomials with $\max_k \boldsymbol{\lambda}_k = n$ correspond to the nodes of the triangle while the rest are obtained using linear interpolation as is outlined in figure 5.2.

With that definition we have a local Bernstein approximation of the balanced variables $\mathbf{U} = (\phi, \phi\mathbf{u})^\top \in (L^\infty(\Omega))^3$ on every element Ω_e of the form:

$$\mathbf{U}(\mathbf{x}, t) \approx \mathbf{U}_h(\mathbf{x}, t) = \sum_{|\boldsymbol{\lambda}|=n} \mathbf{U}_{\boldsymbol{\lambda}} B^n_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}(\mathbf{x})) = \sum_{i=1}^{m} \mathbf{U}(\mathbf{x}_i, t) B^n_{\mathcal{L}(\mathbf{x}_i)}(\boldsymbol{\sigma}(\mathbf{x})).$$

where the Bernstein coefficients $\mathbf{U}_{\boldsymbol{\lambda}}$ are the function evaluations at the Lagrange interpolation points. This also motivates to characterize this ansatz as quasi-nodal: the functions themselves are not nodal, but the coefficients are nodal point evaluations of the to be approximated function.

Another important aspect in our DG model is the computation of derivatives. For Bernstein polynomials the derivatives of order $r \in \mathbb{N}$ in barycentric direction $\boldsymbol{\alpha}$ are given by

$$D^r_{\boldsymbol{\alpha}} B^n_{\boldsymbol{\lambda}}(\boldsymbol{\sigma}(\mathbf{x})) = \frac{n!}{(n-r)!} \sum_{\boldsymbol{\mu}=|r|} B^r_{\boldsymbol{\mu}}(\boldsymbol{\alpha}) B^{n-r}_{\boldsymbol{\lambda}-\boldsymbol{\mu}}(\boldsymbol{\sigma}(\mathbf{x}))$$

with $\boldsymbol{\mu} \in \mathbb{N}^3$ another multi-index. For our computations, we will only consider the derivatives in the unit directions $\mathbf{e}_1, \mathbf{e}_2$ in two-dimensional space, $\boldsymbol{\alpha} \in \{\boldsymbol{\sigma}(\mathbf{e}_1), \boldsymbol{\sigma}(\mathbf{e}_2)\}$, since other directions such as the diagonal of the triangle will be done respecting the metric terms which are independent of the employed polynomials.

For $n = 1$ the Bernstein polynomials on the triangle coincide with the Lagrange polynomials, but for higher order they are different. For any order n, the corresponding interpolation matrix $\boldsymbol{\Psi} = \left(B_{\mathcal{L}(\mathbf{x}_i)}(\boldsymbol{\xi}_j)\right)_{1 \le i,j \le m\prime}$, where the points $\boldsymbol{\xi}_k, k = 1, \dots m\prime$ are the Gauss interpolation points, has only positive entries due to the monotonicity of the polynomials. One can also show that the corresponding Beziér curve, defined as the linear net that interpolates the Lagrange points of the triangle $(\mathbf{x}_i, \mathbf{U}(\mathbf{x}_i))$ (see figure 5.4), is a boundary for the function. An example for second degree Bernstein polynomials can be seen in figure 5.3. One can also show, that they form a partition of unity on the triangle, $\sum_k B^n_{\mathcal{L}(\mathbf{x}_k)}(\mathbf{x}) = 1$ for $\mathbf{x} \in \Omega$, and, that the ith Bernstein polynomial is monotone with a maximum at the Lagrange interpolation point $\mathbf{x}_i$, $1 \le i \le m$.

For their general evaluation the de Casteljau algorithm leads to good results and is based on a recursive evaluation based on Beziér curves. We remark, though, that our model only handles precomputed information about basis functions, so that an implementation of the de Casteljau algorithm was not necessary. The Bernstein ansatz implies additional computational costs, because we in fact solve for the nodal values, but for visualization purposes we need to compute the values of the Bernstein polynomial expansion at the interpolation points. This adds costs of order m^2 to every element Ω_e. Apart from that point, there are no extra costs to be considered in comparison with the nodal Lagrange ansatz.

We stress, that slope limiters still need to be applied to model flooding and drying correctly. The reason is, that even with monotone basis functions, the non-negativity of the fluid height can be affected during the evolution, so that the contribution of the numerical flux has to be limited using the previously discussed slope limiters.

5.5. Numerical Comparison

To test the two modifications to overcome stability problems in higher-order modeling of flooding and drying problems as described in section 5.2, we will employ the wet/dry

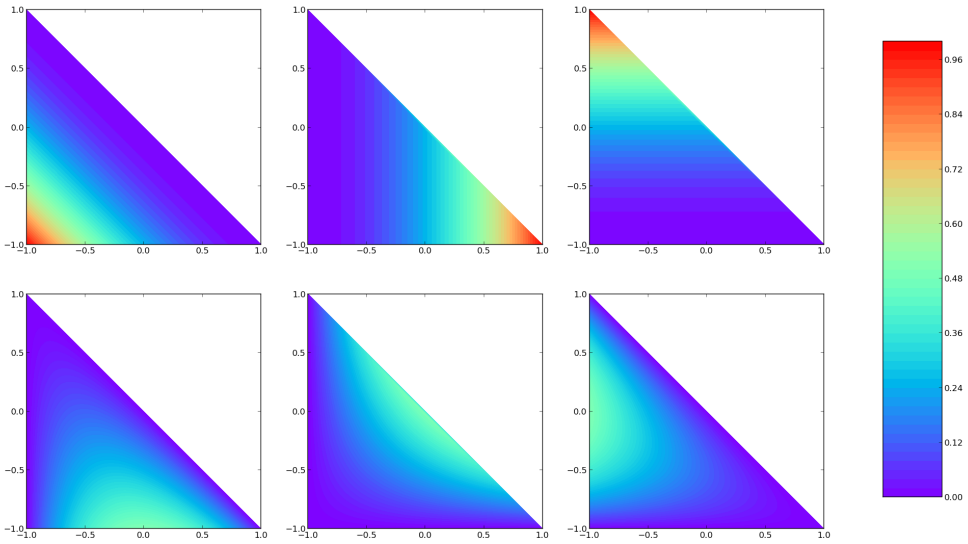

Figure 5.3.: Bernstein polynomials of degree 2. The color scale indicates the values of the polynomial on the reference element with vertices $(-1,-1),(-1,1)$ and $(1,-1)$.

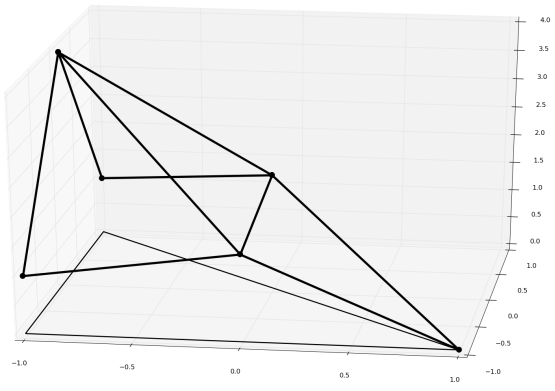

Figure 5.4.: Bernstein-Beziér patch of a triangle with points of order 2

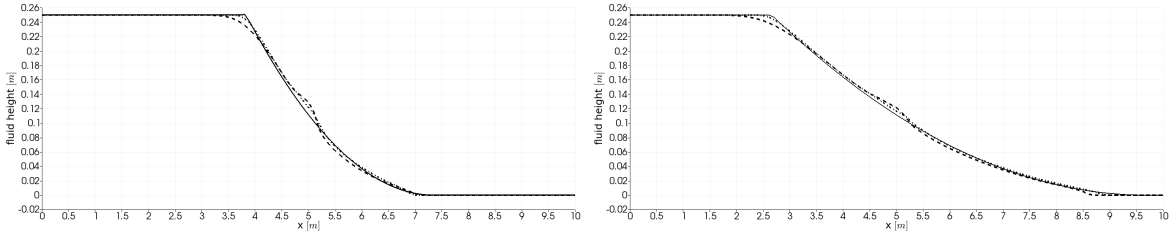

Figure 5.5.: Wet/dry shocktube. Simulation with modified quadratic Lagrange (dashed) and Bernstein (dashed dotted) functions with a linear nodal limiter. Depicted are the numerical and analytical solution (solid line) at times $t = 0.75s$ (left) and $t = 1.5s$ (right).

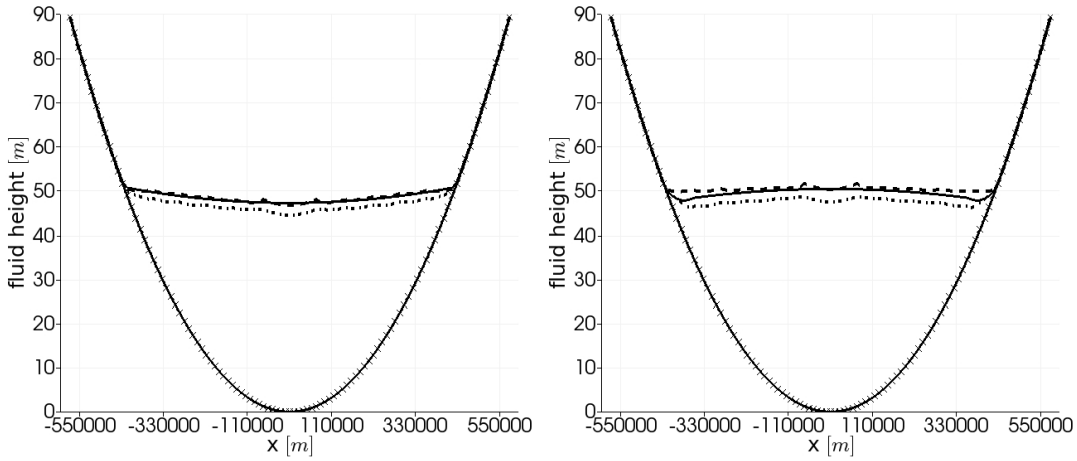

Figure 5.6.: Paraboloid 2D basin. Simulation with modified quadratic Lagrange (dashed dotted) and Bernstein (dashed) functions with a linear nodal limiter. Depicted are the numerical and analytical solution (solid line) at times $t = 190s$ (left) and $t = 380s$ (right).

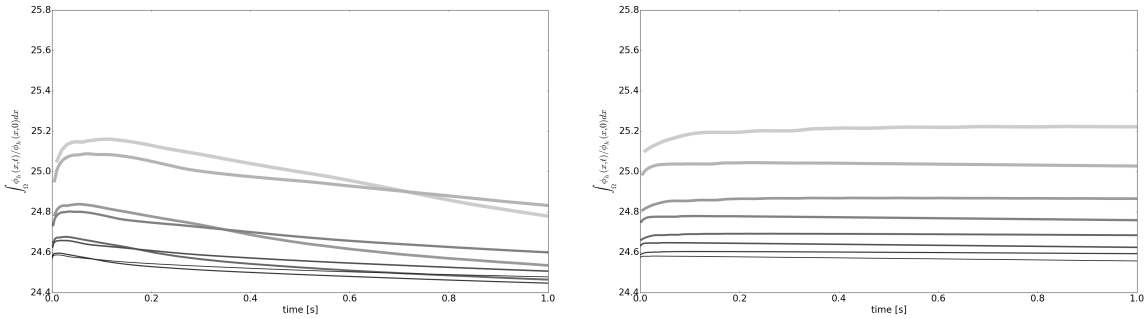

Figure 5.7.: Wet/dry shocktube. Plots of mass for times $0 - 1s$ for simulations with quadratic functions and nodal limiter with linear corrections. Depicted are the Lagrange ansatz (left) and the Bernstein ansatz (right).

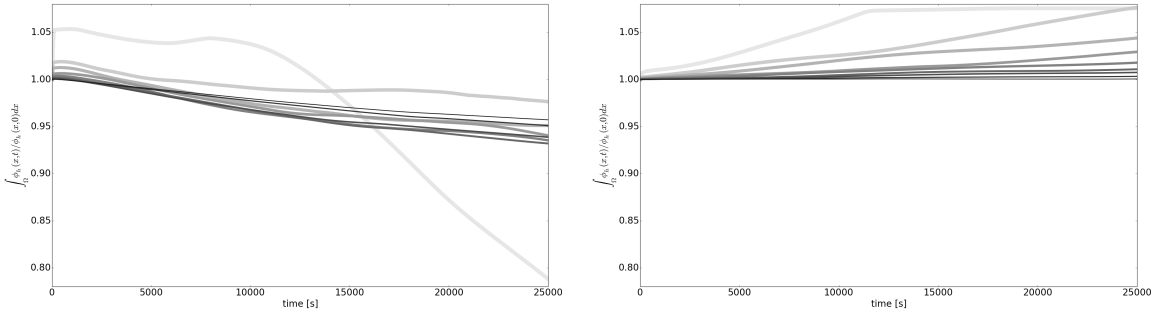

Figure 5.8.: Paraboloid basin. Plots of normalized mass for times $0 - 25\ 000s$ for simulations with quadratic functions and nodal limiter with linear corrections. Depicted are the Lagrange ansatz (left) and the Bernstein ansatz (right).

test problems defined in section 4.3.3. Again, the test results are evaluated with respect to conservation properties and convergence behavior. For the remaining section, we will restrict ourselves to the (VB) limiter, because in our numerical tests in chapter 4 it performed best in terms of robustness and accuracy and exclusively work with polynomials of degree $n = 2$.

Conservation Properties

Conservation properties are of special interest for the treatment of wet/dry simulations. Maintaining a constant mass, provided sensible boundary conditions, is a desirable property.

Figure (5.7) shows the mass, defined as $\int_\Omega |\phi_h(\mathbf{x}, t)| d\mathbf{x}$, measured in the L_1 norm for the first second of simulation time for the wet/dry shocktube. The coloring and strength of the lines refer to the level of refinement in the following way: The lighter and thicker the line, the coarser the simulations. We started with a shortest edge of $\Delta\mathbf{x}_{min} = 0.5$ and then reduced it by bisection to the finest mesh with the shortest edge

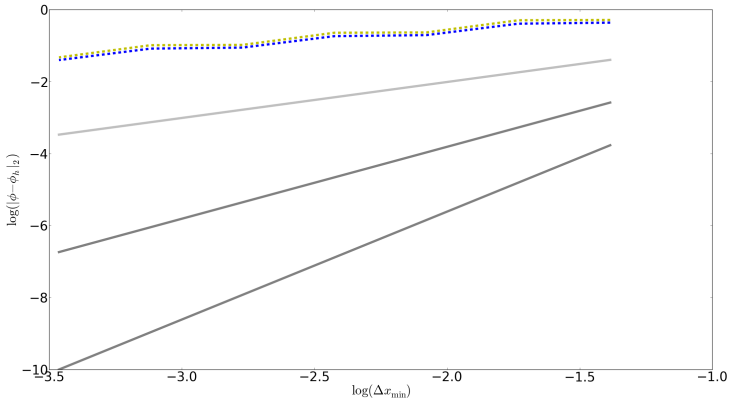

Figure 5.9.: Wet/dry shocktube. L_2 Global convergence plot. Simulations with nodal limiter with linear restriction. Depicted are the Bernstein (yellow) and Lagrange (blue) results after $1s$. The lines have slopes 1,2 and 3 for comparison.

being $\Delta\mathbf{x}_{min} = 0.0442$. The left display shows the results for the quadratic Lagrange polynomials and the right for the Bernstein polynomials. It shows that for this testcase the results of the Bernstein approach shows a better behavior for the wet/dry shocktube. This can be explained by that we do not have to reduce negative values to zero at some of the quadrature points which clearly can lead to a mass loss. The irregularities within the first $0.2s$ can, analogous to the linear case, be explained by the discontinuous initial condition.

Analogously, figure 5.8 shows the normalized mass for the first $25\ 000s$ of the simulation of the paraboloid basin test. The normalization has been carried out with the initial mass at time $t = 0s$ and the resolution was decreased by bisection from $\Delta\mathbf{x}_{min} = 143955.8075$ to $\Delta\mathbf{x}_{min} = 8997.2380$. The results for this test do all show a convergent behavior for both approaches. However, even on the finest mesh, that we employed, the Lagrange approach still led to a mass lost of nearly 3% while the Bernstein approach only led to a loss of less than 0.1%. This can be partly attributed to the very long simulation time, to the only linear correction of the slope limiter, which affected the accuracy in the limited domains, as well as the relatively coarse mesh. In sum, with respect to mass conservation, we can confirm, that the use of Bernstein polynomials is advantageous as expected. The reason for the observed loss of mass of all test cases can probably be attributed to the filters in our model, that cut off very small water heights and set the corresponding momentum to zero in order to avoid the division by very small numbers as discussed in section 4.4.

Accuracy

To study the convergence properties of our model in the L_2, L_∞ and L_1-norms, we decreased the spatial resolution by repeated bisection of triangles from $\Delta\mathbf{x}_{min} = 0.2231$ to $\Delta\mathbf{x}_{min} = 0.0302$ for the wet/dry shocktube and from $\Delta\mathbf{x}_{min} = 287911.62$ to $\Delta\mathbf{x}_{min} =$

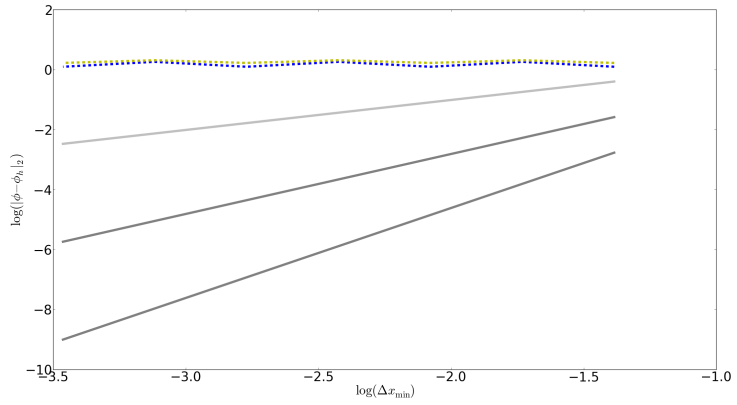

Figure 5.10.: Wet/dry shocktube. L_{inf} Global convergence plot. Simulations with nodal limiter with linear restriction. Depicted are the Bernstein (yellow) and Lagrange (blue) results after $1s$. The lines have slopes 1,2 and 3 for comparison.

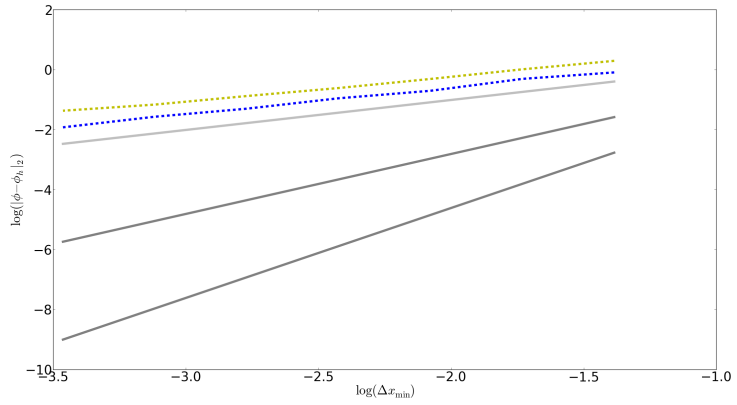

Figure 5.11.: Wet/dry shocktube. L_1 Global convergence plot. Simulations with nodal limiter with linear restriction. Depicted are the Bernstein (yellow) and Lagrange (blue) results after $1s$. The lines have slopes 1,2 and 3 for comparison.

	Bernstein		Lagrange	
Testcase	L_2	L_1	L_2	L_1
Wet/Dry Shocktube	0.5	0.88	0.5	0.75
Paraboloid 2D Basin	1.29	1.59	0.57	0.57

Table 5.1.: Global convergence orders for quadratic basis functions

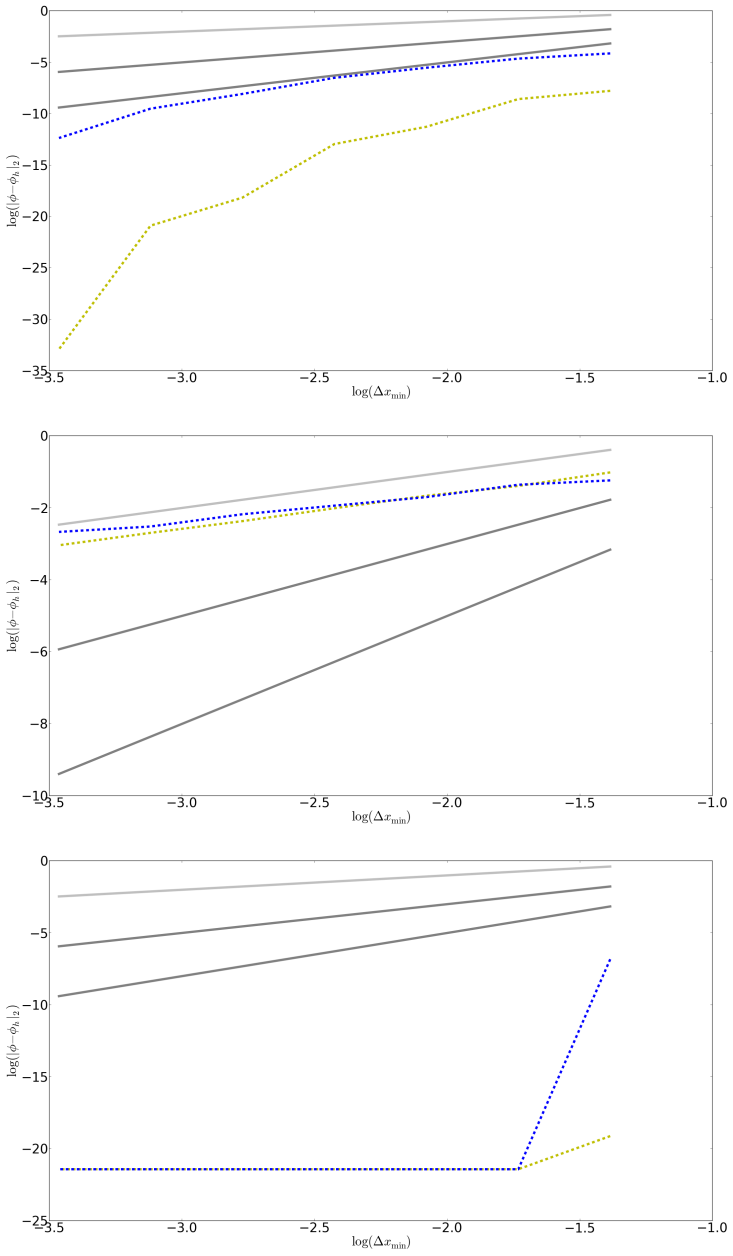

Figure 5.12.: Wet/dry shocktube. L_2 Convergence plot. Simulations with nodal limiter with linear restriction. Depicted are the Bernstein (yellow) and Lagrange (blue) results after $1s$ for the partial regions Ω_1 (top), Ω_2 (middle) and Ω_3 (bottom). The lines have slopes 1,2 and 3 for comparison.

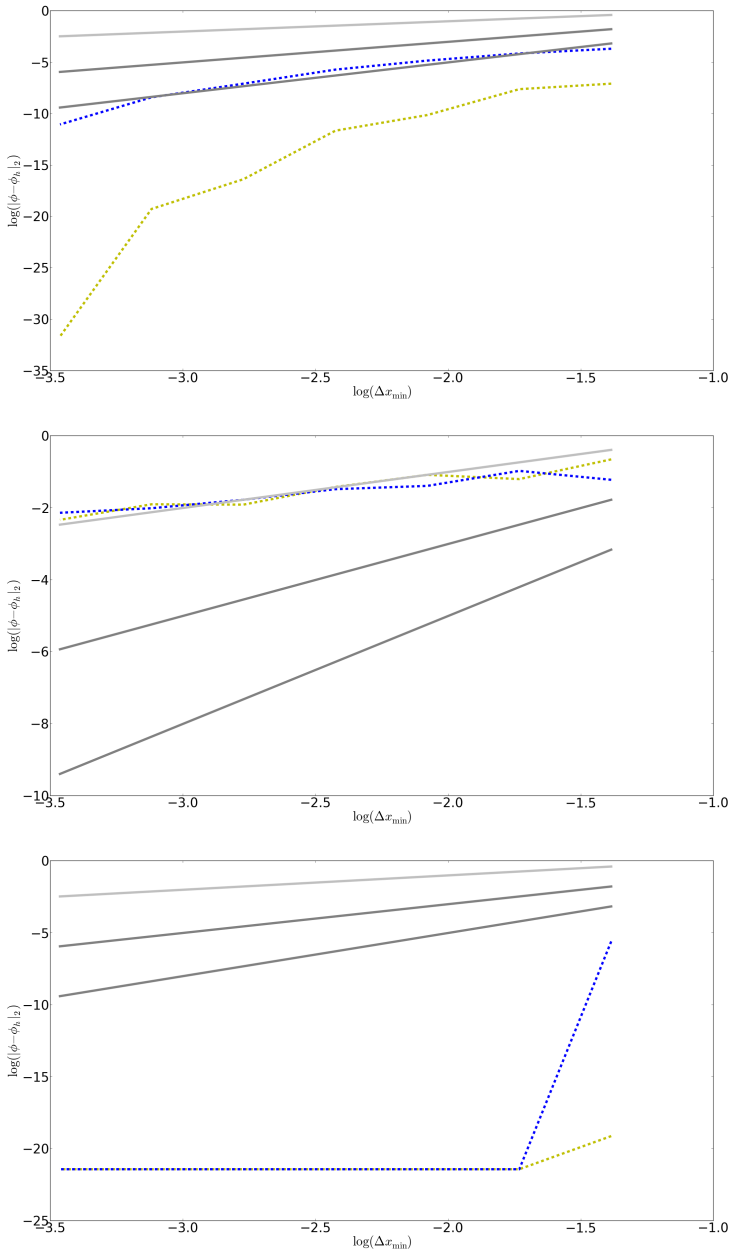

Figure 5.13.: Wet/dry shocktube. L_∞ Convergence plot. Simulations with nodal limiter with linear restriction. Depicted are the Bernstein (yellow) and Lagrange (blue) results after $1s$ for the partial regions Ω_1 (top), Ω_2 (middle) and Ω_3 (bottom). The lines have slopes 1,2 and 3 for comparison.

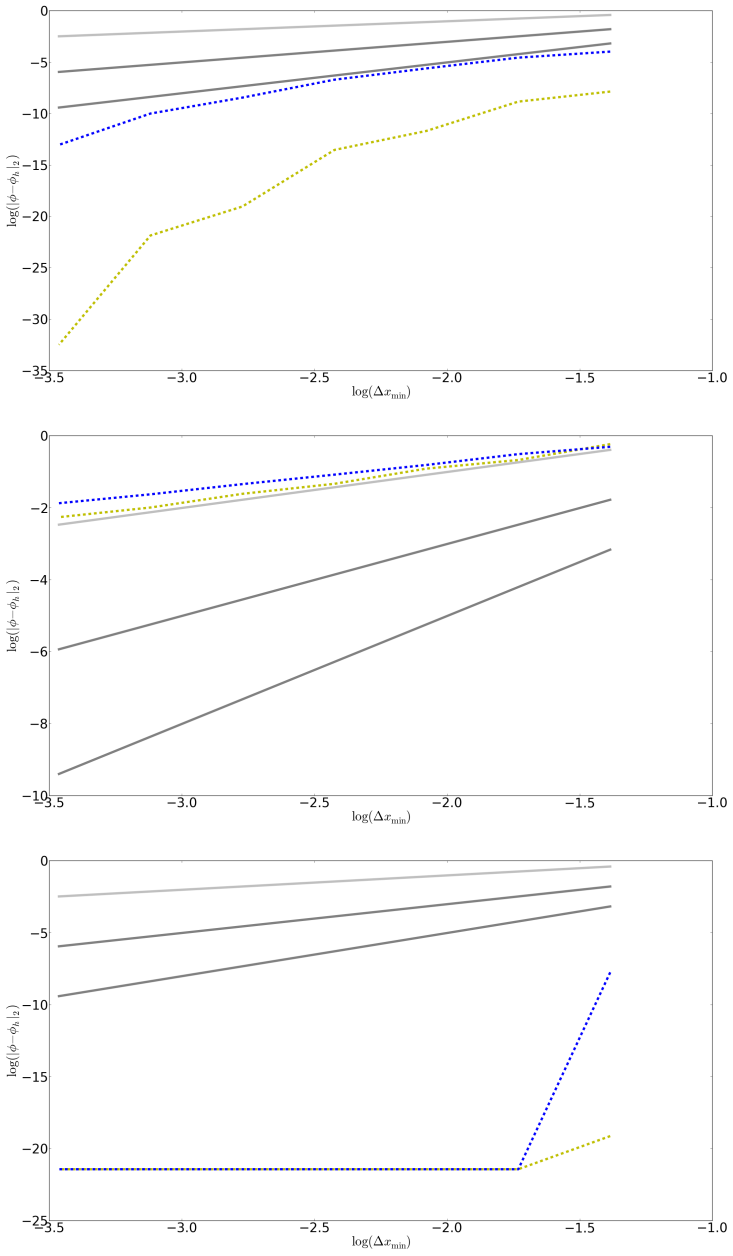

Figure 5.14.: Wet/dry shocktube. L_1 Convergence plot. Simulations with nodal limiter with linear restriction. Depicted are the Bernstein (yellow) and Lagrange (blue) results after $1s$ for the partial regions Ω_1 (top), Ω_2 (middle) and Ω_3 (bottom). The lines have slopes 1,2 and 3 for comparison.

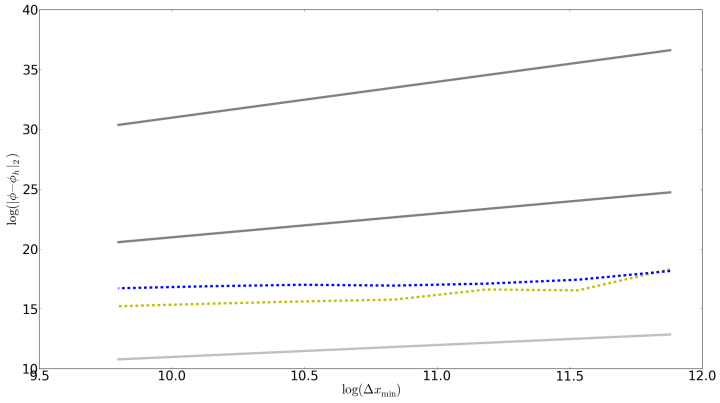

Figure 5.15.: Paraboloid basin. L_2 Global convergence plot. Simulations with nodal limiter with linear restriction. Depicted are the Bernstein (yellow) and Lagrange (blue) results after 30 000s. The lines have slopes 1, 2 and 3 for comparison.

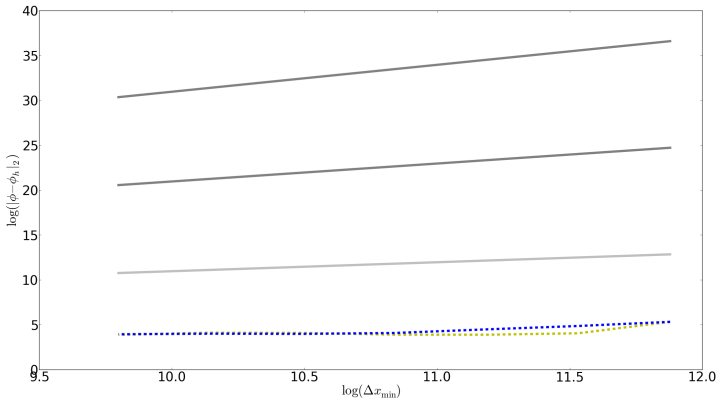

Figure 5.16.: Paraboloid basin. L_∞ Global convergence plot. Simulations with nodal limiter with linear restriction. Depicted are the Bernstein (yellow) and Lagrange (blue) results after 30 000s. The lines have slopes 1, 2 and 3 for comparison.

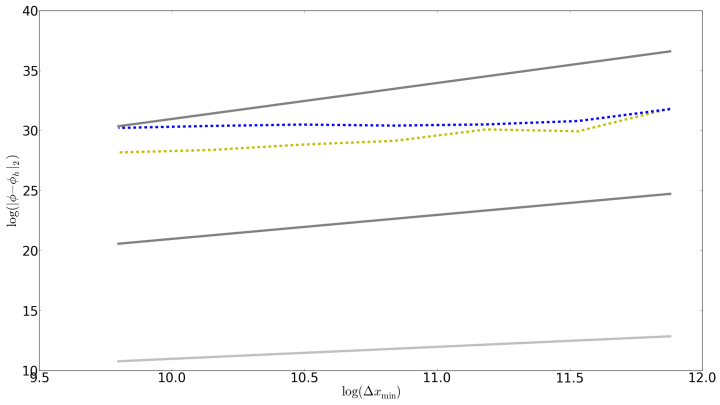

Figure 5.17.: Paraboloid basin. L_1 Global convergence plot. Simulations with nodal limiter with linear restriction. Depicted are the Bernstein (yellow) and Lagrange (blue) results after 30 000s. The lines have slopes 1, 2 and 3 for comparison.

	Bernstein		Lagrange	
Testcase	L_2	L_1	L_2	L_1
Wet/Dry Shocktube (Ω_1^ϵ)	11.02	11.06	3.81	4.22
Wet/Dry Shocktube (Ω_2^ϵ)	0.97	0.97	0.73	0.77
Wet/Dry Shocktube (Ω_3^ϵ)	0.71	0.71	4.51	4.23
Paraboloid 2D Basin (Ω_1^ϵ)	1.77	1.76	0.55	0.46
Paraboloid 2D Basin(Ω_2^ϵ)	0.99	1.39	0.65	1.10

Table 5.2.: Local convergence orders for quadratic basis functions

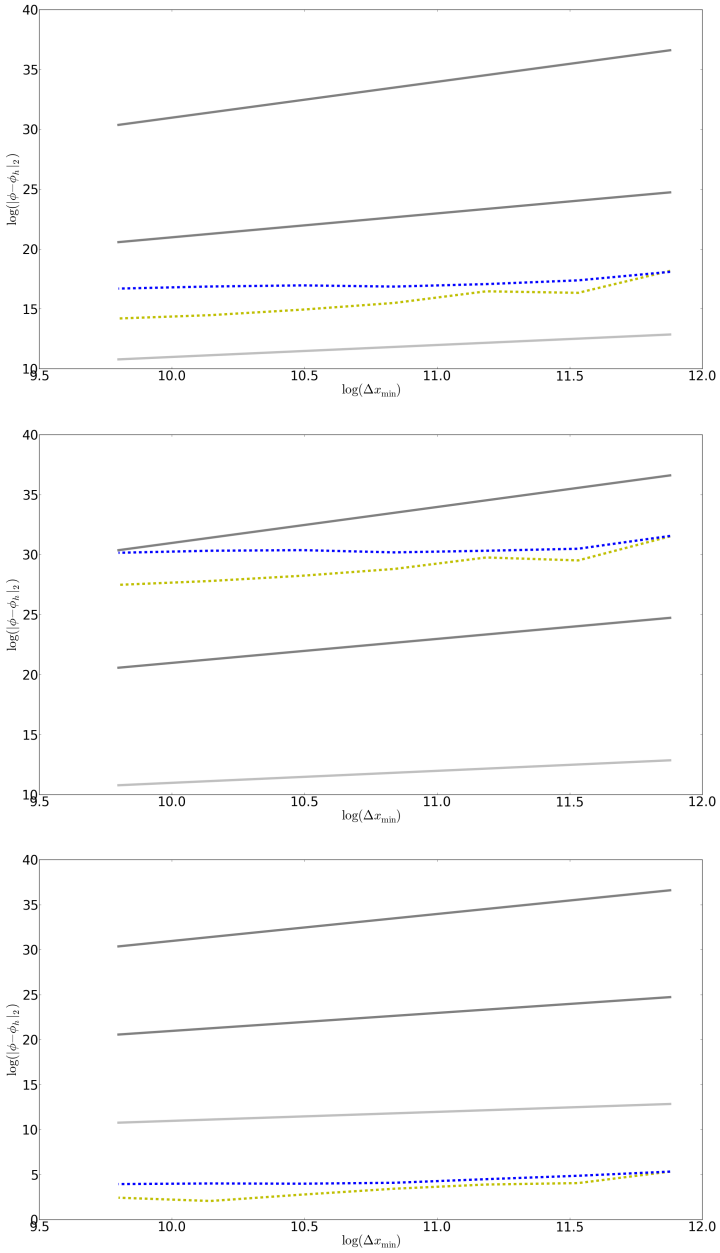

Figure 5.18.: Paraboloid Basin. Convergence plot in L_2 (top), L_∞ (middle), and L_1 (bottom) norms. Simulations with nodal limiter with linear restriction. Depicted are the Bernstein (yellow) and Lagrange (blue) results after $30.000s$ for the partial regions Ω_1^ϵ. The lines have slopes 1, 2 and 3 for comparison.

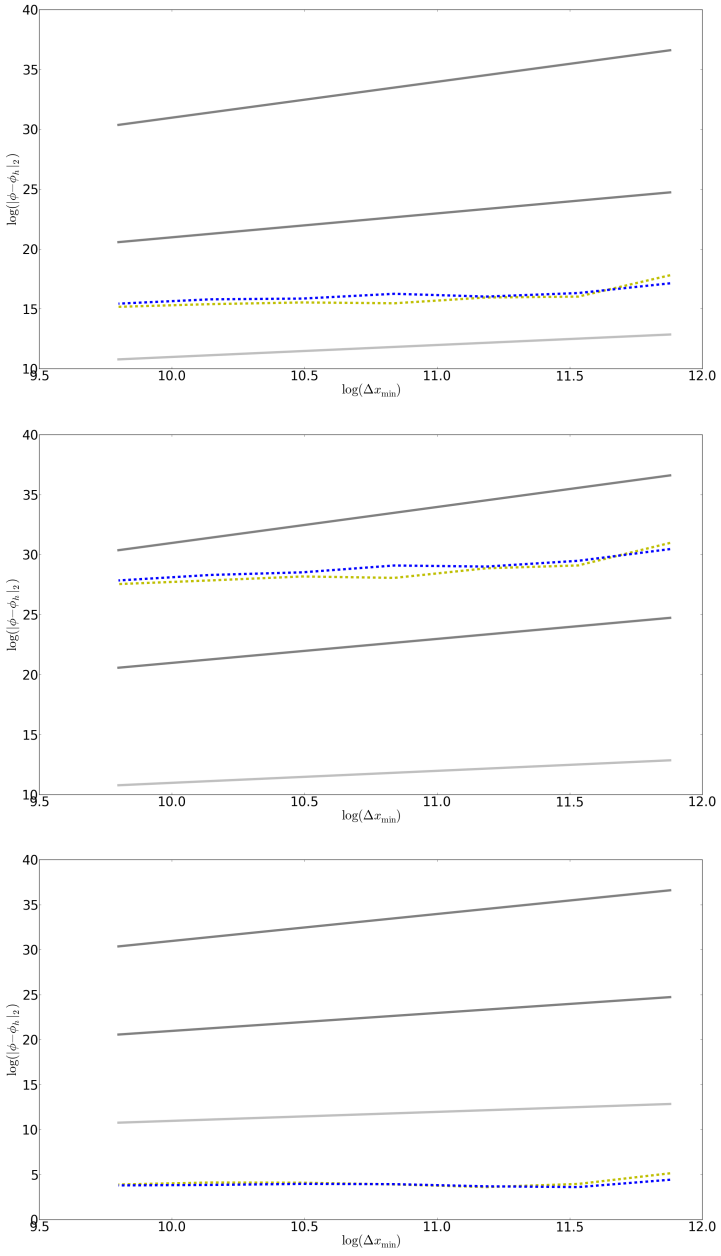

Figure 5.19.: Paraboloid Basin. Convergence plot in L_2 (top), L_∞ (middle), and L_1 (bottom) norms. Simulations with nodal limiter with linear restriction. Depicted are the Bernstein (yellow) and Lagrange (blue) results after $30.000s$ for the partial regions Ω_2^ϵ. The lines have slopes 1, 2 and 3 for comparison.

8997.2380 for the paraboloid 2D basin and measured the maximal error over simulation time. The convergence plots are employing a double-logarithmic scale for reasons of better comparisons. The different colors refer to the two different approaches–yellow refers to the Bernstein polynomials and blue to the Lagrange polynomials– and the straight gray lines are drawn for comparison.

Plots of global numerical errors corresponding to different spatial resolutions can be found in figures 5.9, 5.10 and 5.11 for the wet/dry shocktube in the L_2, L_∞ and L_1-norms respectively. The exact values of the respective convergence rates are computed using a linear least squares approach and are stated in table 5.1. We can see that we can achieve a global order of accuracy of almost 1 in the L_1 norm and 0.5 in the L_2 norm for the wet/dry shocktube. In the L_∞-norm, we do not observe convergence, which is due to distinct irregular points that do not converge.

For the paraboloid 2D basin the corresponding figures for global convergence rates are 5.15, 5.16 and 5.17. In all norms, the global convergence rate for the Bernstein approach is between 1 and 2, while the Lagrange approach does not even yield an order of 1 as can be seen from table 5.1.

The convergence plots for the partial domains for the wet/dry shocktube are figures 5.12 for the L_2 norm, figure 5.14 for the L_1 norm and figure 5.13 for the L_∞ norm. Depicted are the convergence rates for the domains Ω_k^ϵ, that were defined in section 4.3.3 for $k = 1, 2, 3$ from top to bottom. The local convergence rates are depicted in table 5.2. In the left partial domain, the Bernstein polynomials perform better than the Lagrange polynomials, which achieve a convergence between 3 and 4. The unexpected fast convergence of the Bernstein polynomials can be partly explained by the size of the domain Ω_1^ϵ, which is shrinking over time and the constant value of the numerical solution in this part of the domain, which can be better approximated with monotone functions. We observe that in the middle domain a convergence order of higher than 1 can not be achieved. For this partial domain the Bernstein polynomials show a slightly better performance than the Lagrange ansatz, especially in the L_2-norm. Finally in the right part of the domain Ω_3^ϵ, we observe that errors are only present for the coarser simulations and are smaller for Bernstein polynomials. The mostly straight lines are to be explained from a tolerance that is defined in our plotting routine and sets all values that are zero to 10^{-7} to avoid definition problems of the logarithm. The relatively large error of the Lagrange approach for the coarsest simulations, however, leads to a higher convergence rate.

For the paraboloid basin test, we only studied the inner smooth region Ω_1^ϵ as the other region is completely dry. The results for all three norms are depicted in figure 5.18. The improvement of convergence rate is about 0.5 compared to the global convergence and we can see that nearly order 2 is achieved locally in the wet domain without the irregular wet/dry boundary.

Apparently, both approaches yield acceptable results. The error for the Bernstein approach is smaller in general and for the wet/dry shocktube in the third part Ω_3^ϵ of the split domain, we even achieve almost one order of convergence more than we can achieve globally.

5.6. Concluding Remarks

In this chapter, we have studied the viability of third-order basis functions for the simulation of flooding and drying. In section 5.2 we have demonstrated how the lack of monotonicity of nodal Lagrange polynomials lead to stability problems near wet/dry interfaces for flooding and drying problems. To overcome them, we introduced two approaches: a modification of function values on quadrature points with a lower bound of 0 for Lagrange polynomials and Bernstein polynomials as new basis functions. The latter have the advantage that they are monotone, so that the above-mentioned stability problem does not occur. Again, we performed a numerical study following the one in subsection 4.3.3 and measured mass conservation and convergence rates for both tests and approaches.

We ran uniform simulations for the wet/dry shocktube and the paraboloid 2D basin with different spatial resolutions and observed that with increasing resolution, both approaches converge towards mass conservation. The speed of convergence is, however, different. With the modified Lagrange approach we always observe a slight loss of mass over time as expected. For the two-dimensional paraboloid basin test this was shown as a mass lost of still 3% even for the finest spatial resolution of $\Delta\mathbf{x}_{min} = 8997.2380$. This is in contrast to the Bernstein polynomials, which led only to a mass lost of 0.1% for the same test after 25 000s. The same is true for the wet/dry shocktube. With a spatial resolution of $\Delta\mathbf{x}_{min} = 0.0442$, we were able to achieve mass conservation with Bernstein polynomials while with the Lagrange ansatz we lost approximately 0.04% of mass after the simulation time of 1s. So, in conclusion we confirmed superior conservation properties of the Bernstein polynomials.

The theoretical global convergence order of 3 could here, too, not be achieved as in the case of the linear polynomials in chapter 4. With Bernstein polynomials, we achieved a global convergence order (in the L_2- and L_1-norm) of approximately 1.5 for the paraboloid 2D basin and of approximately 0.6 for the wet/dry shocktube. Repeatedly, the investigation of convergence for the partially smooth domains showed that for most partial regions a convergence improvement of up to 2 and even 3 for the dry domain modeled with Bernstein polynomials could be achieved. We remark, that the Bernstein approach that we presented also has the advantage that it is relatively easy to upgrade a nodal model with these quasi-nodal ansatz functions, especially since for DG the basis function-related terms are all precomputed, so that the change of basis can be performed by re-computing the precomputed matrices and vectors.

To study the viability of higher than second order basis functions, we compare the results of the Bernstein approach and the linear Lagrange approach, both obtained in combination with the vertex-based limiter (VB) with linear corrections. Both approaches yielded mass conservation for the wet/dry shocktube that was run for a simulation time of $1s$. The results for the paraboloid 2D basin, that were obtained after 25 000s differ. The third order Bernstein approach led to a mass loss of less than 0.1%, and the linear Lagrange approach to a loss between $1-2$% compared to the initial mass at time $t = 0$ for a spatial resolution of $\Delta\mathbf{x}_{min} = 8997.2380$. If we compare the global convergence rates, we see, that we achieved almost the same convergence rates for the wet/dry shocktube with linear and quadratic polynomials and gained approximately

0.5 of convergence order for the paraboloid 2D basin through the use of quadratic functions. However, as we have seen, a global convergence test might not give the most meaningful results since the solution is not differentiable on some distinct points of the domain.

Studying the local convergence rates from tables 4.3 and 5.2, we see, that in general the convergence rates for the quadratic Bernstein polynomials are higher. The only exception is the solution to the wet/dry shocktube in the domain Ω_3^ϵ. An analysis of the corresponding error plots reveals, however, that the unusual high convergence rate of 4.05 and 3.98 in combination with the (VB) limiter in the L_2 and L_1-norm respectively might be due to the outlier at a resolution of $\log(\Delta\mathbf{x}_{min}) = 1.75$. We observe the largest improvement in the paraboloid 2D basin test: The rate in the wet area Ω_1^ϵ more than doubled.

6. Idealized Adaptive Discontinuous Galerkin Simulations of Surges

6.1. Abstract

The accurate modeling of the forcing terms is essential for storm surge simulations. We show their discretization in our dynamically adaptive Discontinuous Galerkin (DG) model and develop refinement indicators that are suitable for storm surge scenarios. Numerical tests demonstrate the capability of our model to simulate idealized storm surges, and the improvement of computational efficiency by the dynamically adaptive mesh.

6.2. Introduction

As was mentioned in the introductory chapter 1, the national hydrographic agencies employ operational deterministic storm surge models, that are well established and efficient enough to be used for ensemble forecasts. Besides the current operational models, further state-of-the-art numerical shallow water models are, for example, the finite element model on a stationary unstructured triangular grid (ADCIRC) that has been developed in Westerink et al. (2008) and has up to now been used for various applications as well as the finite volume model GEOCLAW on a nested structured grid which has already been used for the simulation of storm surge events in Mandli (2011). A comparison of the two models has been performed in Mandli and Dawson (2014) with simulations of hurricane Ike.

In contrary to the last two models, ADCIRC and GEOCLAW, our model `StormFlash2d` employs a DG approach and a dynamically adaptive triangular mesh.

This chapter is mainly concerned with the simulation of simplified storm surges with our model `StormFlash2d` that was introduced in the previous chapters. Our main goal is to show the capability of the model to handle the respective source forces and show how the adaptive mesh reduces computational costs and does not significantly affect the quantities of interest.

Model-based adjustments are the focus of section 6.3: We show the discretization of the major source terms in subsection 6.3.1 and discuss a suitable refinement criterion for the dynamically adaptive mesh in subsection 6.3.2. The chapter closes with a set of partially dry test cases, uniform and adaptive, with storm forcings in section 6.4.

6.3. Storm-specific Model Parts

Idealized simulations of storm surge events require the handling of large sets of data as well as an accurate representation of involved source forcings. In this section, we show the discretization of the main source forces of storm surges and develop a composite refinement indicator based on the observations from Mandli and Dawson (2014) for the adaptive mesh described in 3.3.2.

6.3.1. Source Term Modeling

In the following, we elaborate on the discretizations of the source terms of equation (3.2): Coriolis forcing, bottom friction and wind stress.

Coriolis Forcing

Coriolis forcing is a source forcing due to the rotation of the earth and has already been described in (3.10) as

$$\mathbf{f}_c = 2\Omega_{rot}\sin(\psi_L)\begin{pmatrix}\phi v\\ -\phi u\end{pmatrix}.$$

The strength of its influence can be determined using the Rossby number as defined in (2.3). From Holthuijsen (2007), we know that the characteristic length and velocity scales of storm surges are $L \in [5, 300]$ and $U \in [1, 5]$. For example, at the latitude of Lübeck, Germany, $\psi_L = 53°$, the Coriolis scale $f = 2\Omega_{rot}\sin(\psi_L)$ is 786.8 with an assumed earth rotation of $\Omega_{rot} = 463.89\ [ms^{-1}]$. Using this value and the characteristic scales L and U from above, we obtain a Rossby number between 0.0027 and 0.8110 which indicates that the Coriolis effect is important to consider. If, however, we make the assumption that the regime and the waves are linear almost everywhere, we obtain a wave frequency for the shallow water waves of $c = \sqrt{\phi}$ as an approximation for the characteristic velocity per length unit: $c \approx \frac{U}{L}$. This leads to an estimate of the Rossby number of $Ro = c/f$, which can also be found in Mandli (2011). The use of this estimate, however, reveals for $h \approx 3000$, a Rossby number of $Ro = 139.1336$, which suggests a negligible influence of the Coriolis forcing. So, in almost linear regimes the influence of Coriolis forcing is strongly dependent on the depths of the water in a way that in deep water, the influence is negligible.

Bottom Friction

The model for the bottom friction is subject to a quadratic Manning law of the form:

$$\boldsymbol{\tau}_b = g\ n_M^2\ \phi\mathbf{u}\frac{\sqrt{\|\mathbf{u}\|_2}}{\phi^{4/3}}.$$

It is known that the bottom friction coefficient n_M changes with water depth h and depends on the constitution of the material at the bottom and its structures. Therefore,

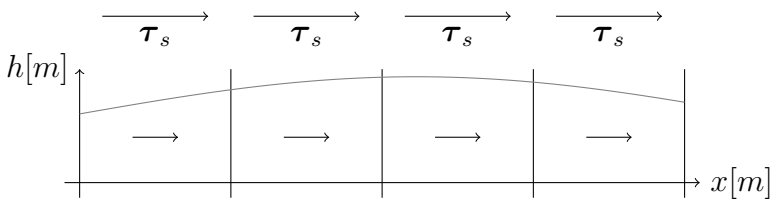

Figure 6.1.: Schematic of wind forcing implementation. Movement of water columns in a one-layer shallow water model.

the bottom friction is usually higher in areas of shallow water and rough bottom topography. The bottom friction model that we use can vary over the domain depending on the water height. Usually the assumptions for Manning's n_M will be $n_M \approx 0.02$ for beaches or sandy ground and $n_M \approx 0.25$ for dry structures as can be found, for example, in Chow (1959). The ansatz, that we chose in our model assumes a homogeneous underground and interpolates linearly between a user-defined n_{max} and n_{min} depending on a user-defined maximum fluid height h_{max} using the dynamic formula:

$$n_M(\mathbf{x}, t) = \frac{h(\mathbf{x}, t) \cdot (n_{min} - n_{max})}{h_{max}} + n_{max}.$$

Another possibility would be to specify polygonal regions of specific bottom friction coefficients. For the idealized applications, that we show, however, the simple linear interpolated bottom friction was sufficient.

An Idealized Storm Model

In our DG model, wind stresses are modeled using the formula:

$$\boldsymbol{\tau}_s = \frac{\gamma_\tau \boldsymbol{\tau}}{\phi \rho},$$

as was already shown in (3.10). The determination of the vector field $\boldsymbol{\tau}$ depends on the model scenario. As we have a one-layer model, the wind forcing will affect the movement of the whole water columns as is outlined in figure 6.1. This simplification is of limited correctness especially in deep water. More realistic would be to have a two-layer model as was developed for example in Mandli (2011) for a finite volume model, where the wind stress is only affecting the relatively small top layer of the fluid.

In general any two-dimensional vector field $\boldsymbol{\tau}$ can be used to describe a wind. As for storms, or hurricanes, we usually have information about the mean radius to maximum winds, central pressure, intensity and track of a storm, which are obtained from measurements and satellite observations, we use these data to obtain a two-dimensional vector field using a common approximation for the winds.

Holland's storm model can be utilized for modeling hurricane profiles as is shown in Holland (1980), where analytical models are especially needed because observations are

too sparse. Using cyclostrophic balances, one obtains winds $\mathbf{w}$ via

$$C = \frac{AB(p_n - p_c)}{\rho_{air}}, \quad \langle \mathbf{w}, \mathbf{n} \rangle = \sqrt{C \cdot \frac{\exp\left(\frac{-A}{r^B}\right)}{r^B} + \frac{r^2\ f^2}{4}} - \frac{r\ f}{2} \tag{6.1}$$

where p_n is the ambient pressure, p_c the central pressure, A and B are model-specific shape parameters, r is the radius of the storm and f the Coriolis forcing. The vector $\mathbf{n}$ is the outward pointing normal vector with respect to the isobars of the storm. The air pressure $\rho_{air} = 1.15$ is assumed to be constant. If the Rossby number Ro indicates that the Coriolis forcing is negligible, we will also neglect it in (6.1), which simplifies the computations. Following Weisberg and Zheng (2006, formula (9)) the wind field $\boldsymbol{\tau}_s$ is then given by

$$\boldsymbol{\tau}_s = C_d\ \rho_{air} \mathbf{w} |\mathbf{w}|$$

A suitable computation of wind drag C_d is chosen as

$$1000.0 \cdot C_d = \begin{cases} 1.2 & |\boldsymbol{\tau}_s| \leq 11 \\ 0.49 + 0.065 \cdot |\boldsymbol{\tau}_s| & 11 \leq |\boldsymbol{\tau}_s| \leq 25 \\ 0.49 + 0.065 \cdot 25 & \text{otherwise.} \end{cases}$$

Details can be found in Weisberg and Zheng (2006). An example of the wind model result for hurricane Ike can be found in figure 6.2.

From the wind model we obtain a wind field at a certain time point. The track of the cyclone is recorded at discrete time points t_k with $\Delta t_{eye,k} = t_k - t_{k-1}$ for $k = 1, \ldots n$. We denote the position of the cyclone at time t with $\text{pos}_{eye}(t)$ and the position of the storm for intermediate time points $t_{k-1} \leq t_l \leq t_k$ will be linearly interpolated as follows:

$$\text{pos}_{eye}(t_l) = \frac{t_l - t_{k-1}}{\Delta t_{eye,k}} \left(\text{pos}_{eye}(t_k) - \text{pos}_{eye}(t_{k-1})\right) + \text{pos}_{eye}(t_{k-1}).$$

The discrete samples also make it possible to compute the velocity $\mathbf{v}_{storm}$ of the storm itself using the great circle distance. Let therefore $\psi_{L,k}, \theta_{L,k}$ be the latitude and longitude of the kth sample and $\Delta\psi_{L,k} = \psi_{L,k} - \psi_{L,k-1}$, $\Delta\theta_{L,k} = \theta_{L,k} - \theta_{L,k-1}$ their absolute differences. Then, the great circle distance is defined as

$$ds = r_{earth} \cdot 2 \arcsin\left(\sqrt{\sin^2(0.5\Delta\psi_{L,k}) + \cos(\theta_{L,k-1})\cos(\theta_{L,k})\sin^2(0.5\Delta\theta_{L,k})}\right),$$

with $r_{earth} = 6371.0\ [km]$ the radius of the earth. The velocity at time t with $t_{k-1} \leq t \leq t_k$ is then computed as

$$\mathbf{v}_{storm}(t) = \frac{ds}{\Delta t_{eye,k}} \text{sign}\left(\text{pos}_{eye}(t_k) - \text{pos}_{eye}(t_{k-1})\right),$$

with sign the component-wise signum function which returns the sign of its arguments.

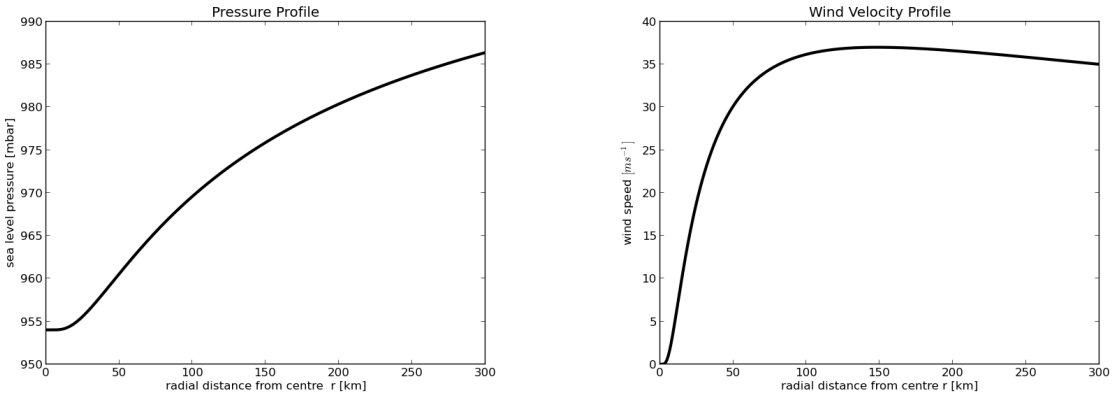

Figure 6.2.: Wind and pressure profile. Numerical approximation for hurricane Ike with parameters $A = 37.5, B = 0.725, p_n = 1013$ [mbar], $p_c = 954$ [mbar].

The model-specific parameters A and B can be computed using the maximum wind speed $\mathbf{w}_{max}$ and the maximum wind radius r_{max}:

$$B = e\ \rho_{air} \frac{\mathbf{w}_{max}^2}{p_n - p_c}, \qquad A = r_{max}^B.$$

6.3.2. Dynamic Mesh Adaptation for Storm Surges

With their large spatial extents and the local fine-scale responses to the storm forcing, storm surge simulations demand for dynamically varying resolution in order to save computing time and represent the main effects. The meshes, that we use are h-adaptive. H-adaptivity comprises the insertion or deletion of (new) nodes and a corresponding increase or decrease of the number of elements of the initial mesh This can be done for individual nodes or in local areas, which is called nesting and means the local insertion of finer meshes. An example for a mesh nesting can be found in GEOCLAW (see e.g. Mandli (2011)). In this thesis, we will exclusively concentrate on h-adaptive meshes that are not necessarily nested and which we create and manipulate with the help of the library `amatos`.

Using this library, we create a dynamically adaptive and conforming triangular mesh. As we have seen earlier in section 3.3.2 and will see in this chapter, the triangular mesh shows potential to represent difficult coastlines up to polygonal precision and therewith reduce artificial oscillations. An essential point for h-adaptive meshes is the choice of the refinement strategy. According to Behrens (2006) it will not be possible to find a strategy that will work well for all kinds of applications. They will all have advantages and disadvantages. We employ a refinement by bisection with a resolved patch strategy as introduced in Bänsch (1991). This resolved patch strategy ensures that the mesh is kept conforming throughout the computation. Conformity, in this respect, means that the intersection of two distinct triangles is either empty, a common node, or a common edge and that the triangles cover the whole domain and their open interior is disjoint.

Conforming meshes have the computational advantage that fluxes between elements are easier to compute as hanging nodes are avoided. One particular advantage of meshes that are generated with `amatos` is the inherent space-filling curve ordering of elements as shown in Behrens and Bader (2009) that leads to a reduction of cache-misses and, with that, an increase of the computational efficiency.

The quality of adaptive simulations is strongly dependent on the adaptive mesh itself. This, in turn, is influenced by the interpolation routine, the tolerances $\theta_{ref}, \theta_{crs}$ for refinement and coarsening, the grid levels $\lambda_{ref}, \lambda_{crs}$, and the refinement indicators η_{Ω_e}, that are needed to modify a user-defined initial triangulation. The grid represent the maximum and minimum refinement of the initial mesh.

An interpolation routine is needed to determine nodal values of the modified degrees of freedom after mesh adaptation. Within the mesh adaptation framework `amatos`, we therefore developed and tested interpolation routines for a refined and coarsened mesh. In case of refinement, the routine employs the local DG basis functions and evaluates them at the new degrees of freedom, which is mass conserving. For the coarsened mesh, we also employ the local basis functions. In order to conserve mass, we take the mean of the shared node.

The refinement indicator is to be distinguished from the refinement strategy. The strategy is bisection of the longest edge while the refinement indicator determines the elements that are to be limited. The choice of a refinement indicator will be described in the following subsection.

Criteria for Manipulation

As in 3.3.2, we determine suitable refinement indicators for storm surge simulations with the grid generator `amatos`. For practical geophysical applications, the implementations in Behrens (2006) suggest different kinds of indicators: error proxies (e.g. the gradient of the geopotential height), physics-based criteria (e.g. the vorticity for cyclone tracking), or mathematical error estimators (residuals of operators).

In our DG model, we have implemented error proxies in combination with physics-based influences. The development of mathematical rigorous error estimators is an interesting question. However, the expected high computational cost let us not pursue this aspect any further.

The definition of refinement indicators for storm surge simulations has to ensure that all meaningful effects are captured by the adaptive mesh. As recently shown in Mandli and Dawson (2014) for hurricane Ike, it is essential that the mesh captures the wind-induced displacement of the water columns as well as the storm and therefore a composite indicator is recommended that takes care of both effects. The indicators, that we worked with in section 6.4 are of the form

$$\begin{aligned} \eta^1_{\Omega_e}(t) &= \|\boldsymbol{\tau}_s(t)\|_{2,\Omega_e}, \\ \eta^2_{\Omega_e}(t) &= \begin{cases} 1 & \text{for } \max_{\mathbf{x}_i \in \Omega_e} |h_h(\mathbf{x}_i, t) - h(\mathbf{x}_i, 0)| \geq \text{TOL} \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{6.2}$$

with a problem dependent tolerance TOL. They represent the L_2-norm of the wind vector field, that has been described in the previous section and the maximal deviation of the numerical solution h_h from the initial still water state at time $t = 0$. The dependence on the time variable t is explicitly shown to indicate that the mesh is modified in every timestep. The grid manipulation is then carried out as follows:

$$\begin{array}{ll} \text{if} & \eta^1_{\Omega_e} \leq \theta_{crs}\eta^1_{max} \text{ and } \eta^2_{\Omega_e} \leq \theta_{crs}\eta^2_{max} \quad \rightarrow \text{coarsen element } \Omega_e \\ \text{if} & \eta^1_{\Omega_e} \geq \theta_{ref}\eta^1_{max} \text{ or } \eta^2_{\Omega_e} \geq \theta_{ref}\eta^2_{max} \quad \rightarrow \text{refine element } \Omega_e, \end{array} \tag{6.3}$$

with $\eta^k_{max} = \max_{\Omega_e \subset \Omega} \eta^k_{\Omega_e}, k = 1, 2$, the maximum of the element-wise refinement indicators over all elements. An additional indicator is given by the CFL number, which is preferably close to 1, in order to assure, that the maximal possible timestep has been computed. A too short timestep can impose additional diffusion and computational costs. To avoid waves traveling through more than one cell per timestep, we bound the CFL number above by 1. This leads to an additional grid modification of the form:

$$\begin{array}{ll} \text{if} & \text{CFL}_e > 1 \quad \rightarrow \text{coarsen element } \Omega_e \\ \text{if} & \text{CFL}_e < 0.7 \quad \rightarrow \text{refine element } \Omega_e. \end{array} \tag{6.4}$$

The quantity CFL_e is the CFL number for element Ω_e and the lower tolerance of 0.7 has been determined experimentally.

6.4. Simplified Storm Surge Scenarios

We test the implementation of Holland's model and the capability of our DG model to simulate simplified idealized storm surges using two partially dry test problems with storm forcings.

Testcase 9. *Storm in a paraboloid basin*
We consider a domain: $\Omega = [-L, L]^2$ with parameters: $D_0 = 50, L = 430620$, reflecting boundaries, Coriolis forcing, a Manning friction with $n_M = 0.01$ and initial conditions

$$\begin{aligned} D(\mathbf{x}) &= D_0 \left(1 - \frac{x^2 + y^2}{L^2}\right) \\ h(\mathbf{x}, 0) &= \max(D(\mathbf{x}), 0) \\ \mathbf{u}(\mathbf{x}, 0) &= \mathbf{0} \\ b(\mathbf{x}) &= D_0 \frac{x^2 + y^2}{L^2} \end{aligned}$$

In the domain center, we introduce a constant storm with parameters: $p_n = 1005, p_c = 950, B = 23$, and $A = 1.5$ that is smoothly ramped up over the first 6 hours.

We ran the simulation with a timestep of $\Delta t = 0.5$, a second order RK22 method and a uniform spatial resolution of $\Delta \mathbf{x}_{min} = 2378.86$ in combination with the (VB) limiter. The simulation time amounted to $t = 50000s$.

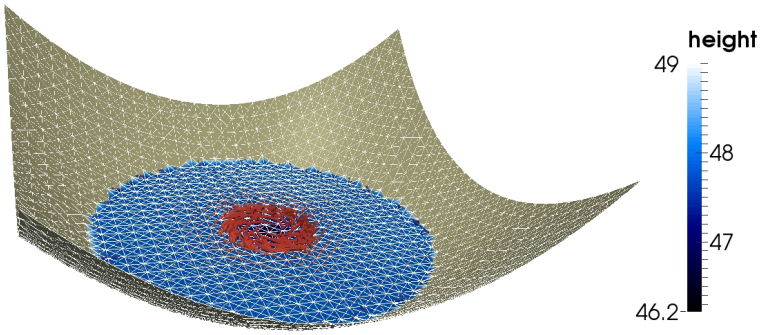

Figure 6.3.: Storm in a paraboloid basin. Uniform simulation. The coloring of the vectors correspond to the strength of the velocity after 282 500 s

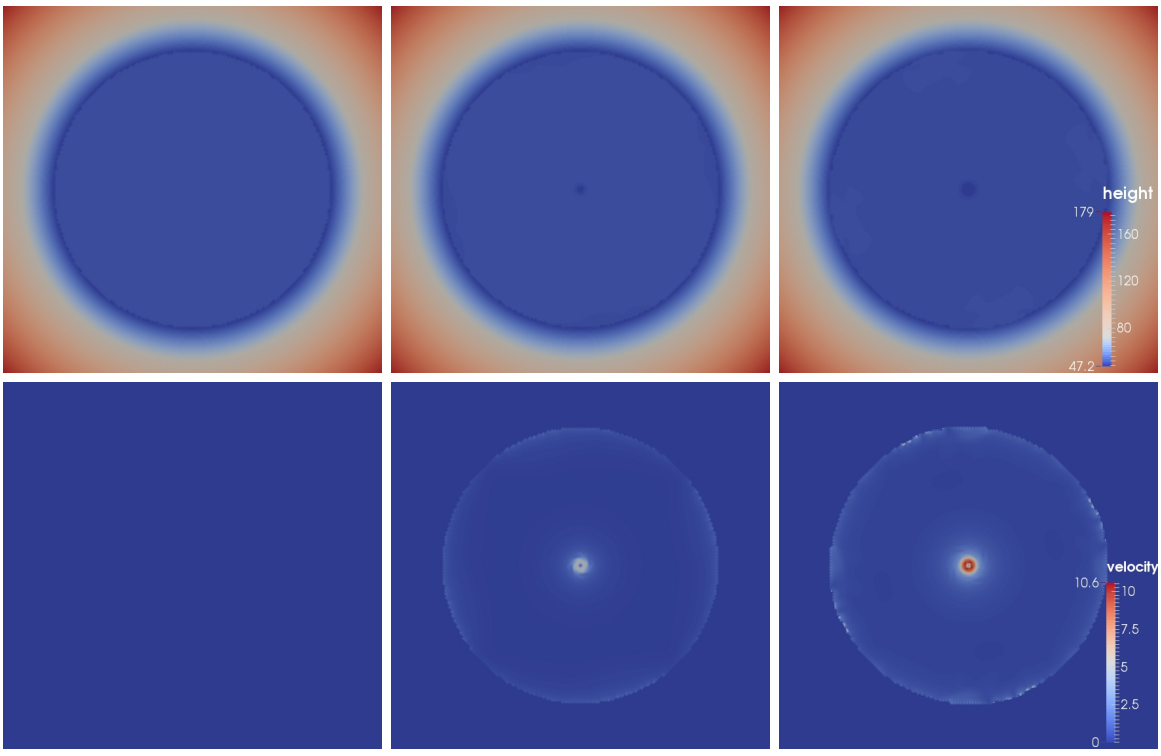

Figure 6.4.: Storm in a paraboloid basin. Plots of total height (top row) and magnitude of velocity (bottom row). Depicted are snapshots every 17500s.

	A	B	C	D	E	F
velocity $[m/s]$	5.0	5.0	5.0	5.0	5.0	25.0
angle [deg]	0	45	-45	90	-90	0
$\text{pos}_{eye}(t=0)$	$(0,0)$	$(200,-100)$	$(200,100)$	$(400,-100)$	$(400,100)$	$(0,0)$

Table 6.1.: Traveling Storm. Storm characteristics

	A	B	C	D	E	F
$\gamma_{\boldsymbol{\tau}}$	1.0	1.0	1.0	1.0	1.0	100.0
# elmt	3986	3512	3501	2274	2310	2178
CPU time	0.4075	0.3923	0.3979	0.2790	0.2747	0.2530

Table 6.2.: Traveling Storm. Numerical results: wind friction parameter, number of elements of adaptive simulation and corresponding normalized CPU time.

Our observations are depicted in figures 6.3 and 6.4. We see that the velocities develop into circular motion in the domain center as expected. After $17500s$ the fluid height responds to the forcing and a small depression zone is forming in the eye of the storm. Furthermore, we observe small velocities at the wet/dry boundary of the domain. This is probably a results of the badly conditioned division of momentum by geopotential height to obtain the velocities at these points, but it did not affect the robustness of our results, nor did it accumulate over time.

Testcase 10. *Traveling Storms*
The model domain is $\Omega = [-200000, 500000]\times[-300000, 300000]$ with a linear bathymetry of the form

$$b(\mathbf{x}) \equiv b(x) = \begin{cases} 0 & \text{for } x \leq 350000 \\ 0.025 \cdot (x - 350000) & \text{otherwise.} \end{cases}$$

and initial conditions

$$\begin{aligned} h(\mathbf{x}, 0) &= \max(0, 3000 - b(x)) \\ \mathbf{u}(\mathbf{x}, 0) &= \mathbf{0} \end{aligned}$$

Six different storms with constant storm parameters $p_n = 1005, p_c = 950, B = 23$, and $A = 1.5$, are initialized at a start position $\text{pos}_{eye}(t=0) \cdot 1000$ and travel with different angles and velocities as depicted in table 10. Wave measurements are taken from four gauges at $\mathbf{G}_1 = (425000, 200000)^\top, \mathbf{G}_2 = (425000, 100000)^\top, \mathbf{G}_3 = (425000, -100000)^\top$ and $\mathbf{G}_4 = (425000, -200000)^\top$ for comparison.

We ran the simulations with the vertex-based limiter, a second order Runge Kutta scheme (RK22) with a timestep of $\Delta t = 5.0$ in combination with a Rusanov Riemann solver and the refinement indicator described in section 6.3.2 for 40h of simulation

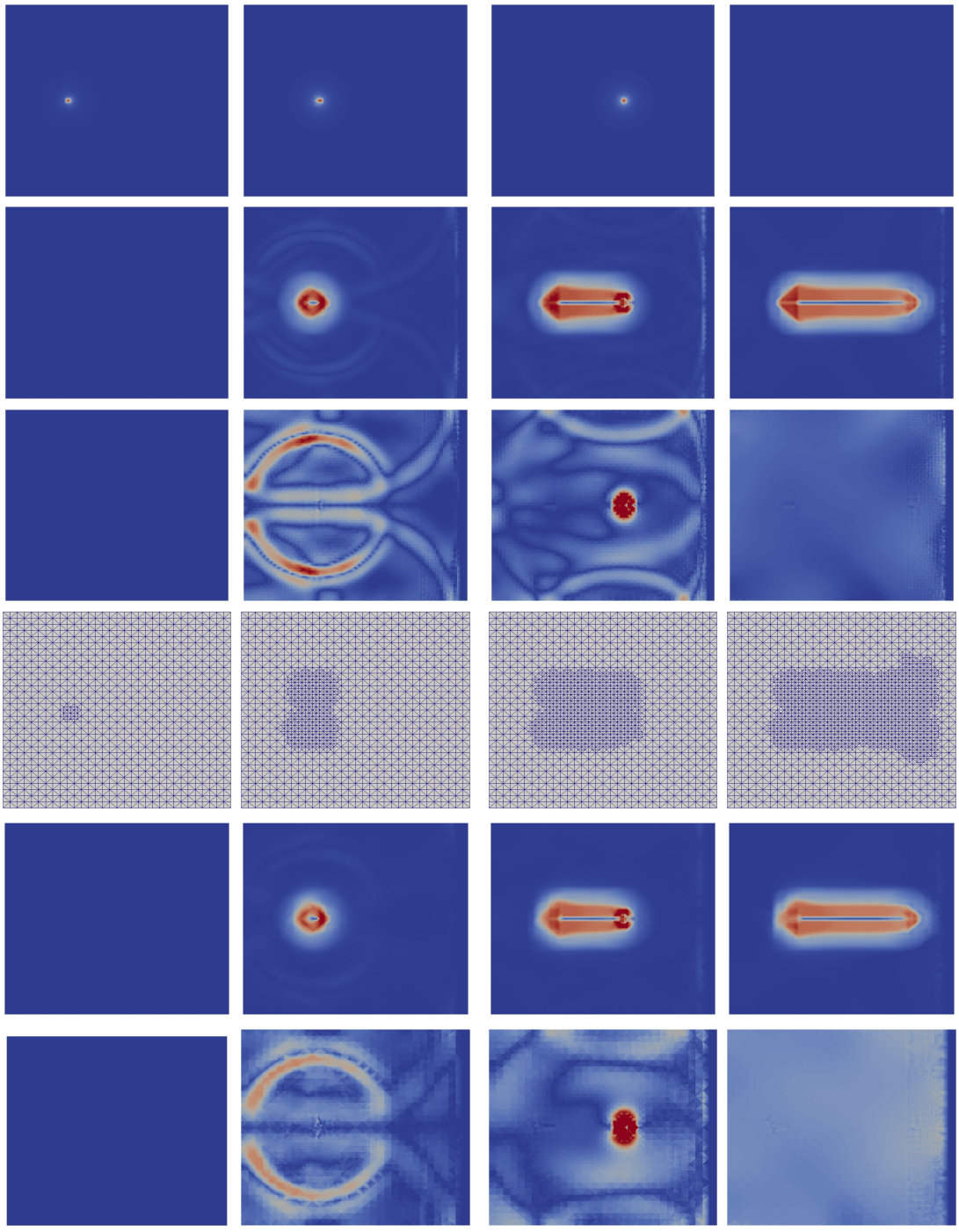

Figure 6.5.: Traveling Storm A: Uniform and adaptive simulation. Depicted are the storm location (top row), magnitude of velocity (second row), water displacement (third row), the adaptive mesh (fourth row), as well as the magnitude of velocity (fifth row) and the water displacement (bottom row) on the adaptive mesh, every 10 h.

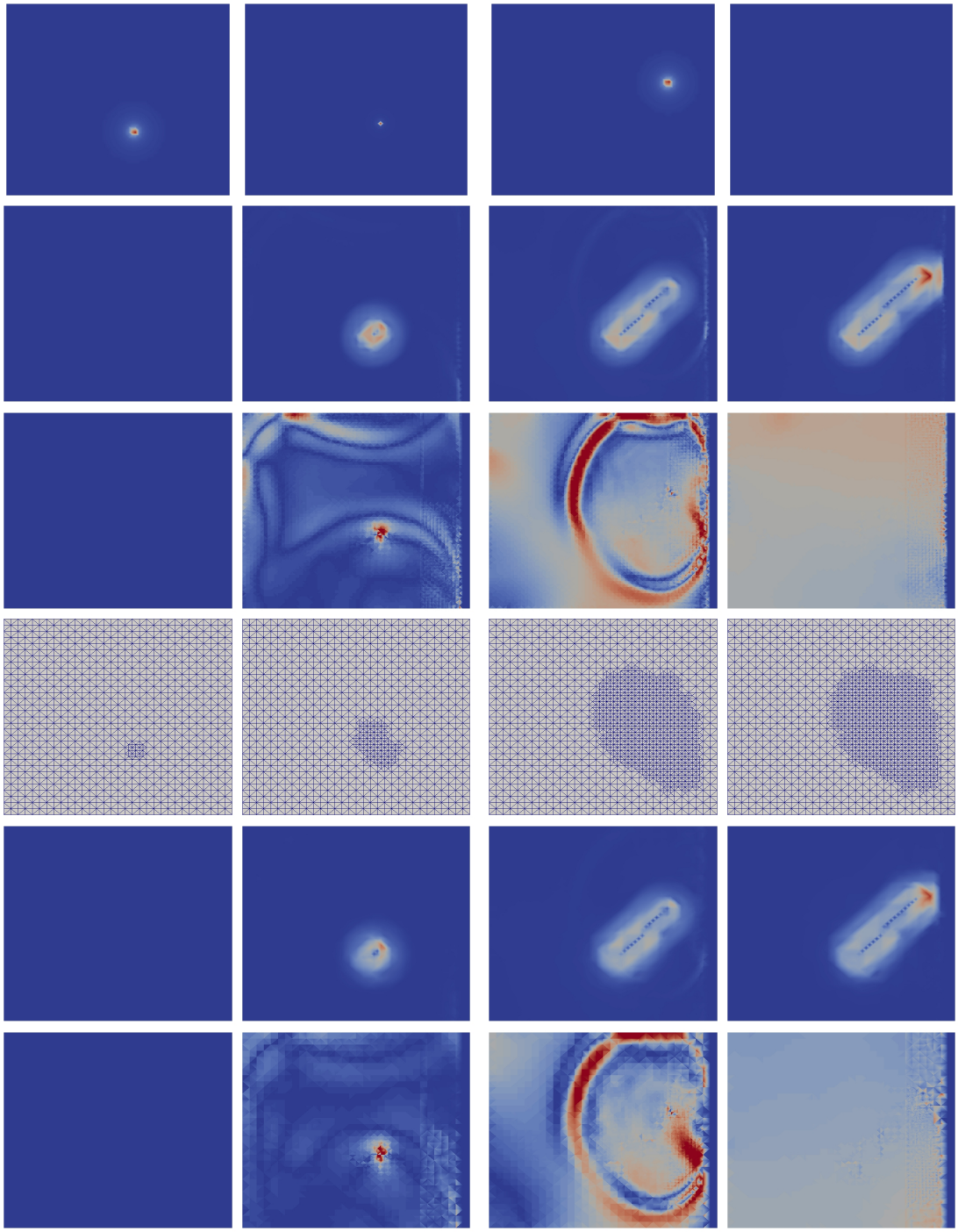

Figure 6.6.: Traveling Storm B: Uniform and adaptive simulation. Depicted are the storm location (top row), magnitude of velocity (second row), water displacement (third row), the adaptive mesh (fourth row), as well as the magnitude of velocity (fifth row) and the water displacement (bottom row) on the adaptive mesh, every 10 h.

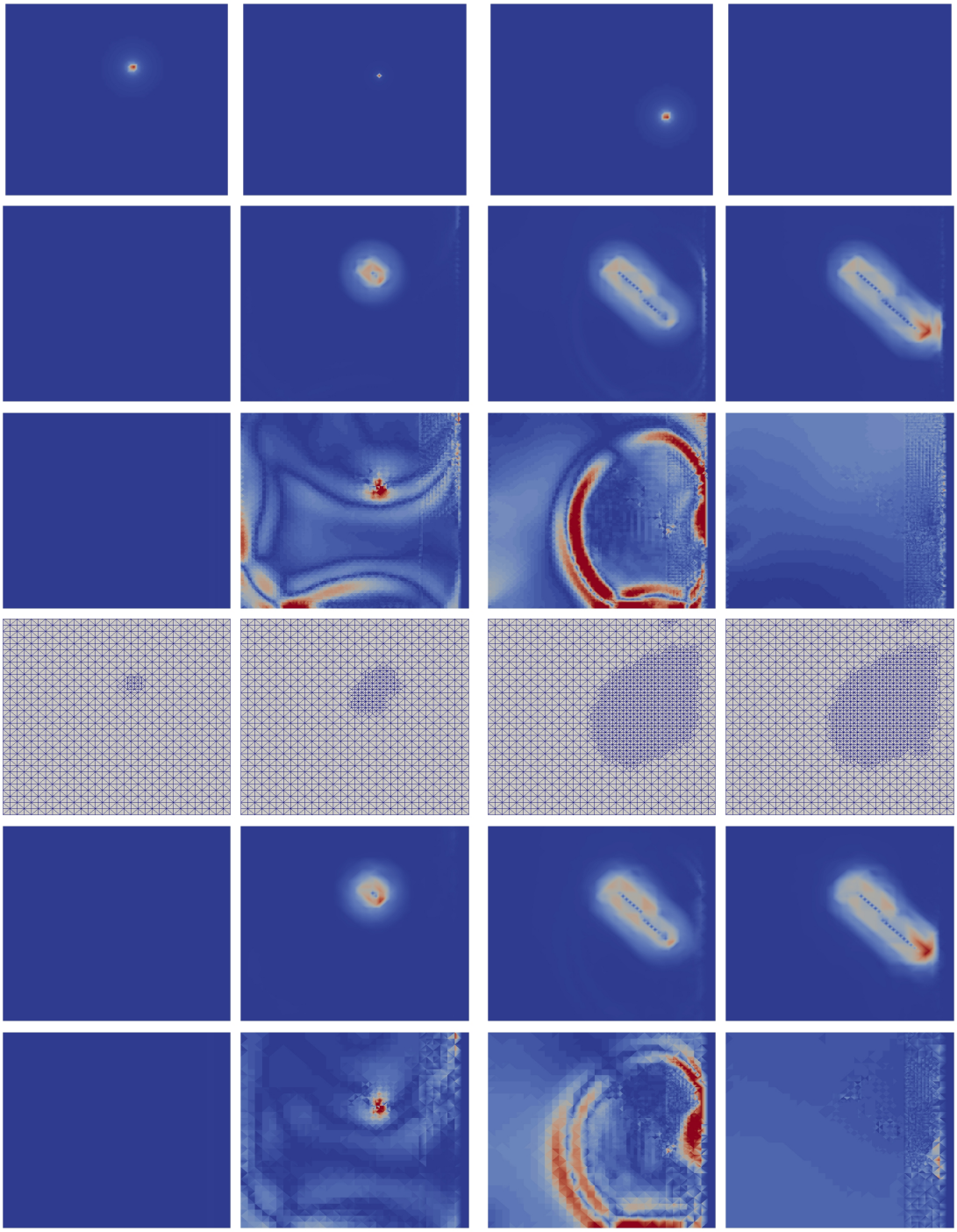

Figure 6.7.: Traveling Storm C: Uniform and adaptive simulation. Depicted are the storm location (top row), magnitude of velocity (second row), water displacement (third row), the adaptive mesh (fourth row), as well as the magnitude of velocity (fifth row) and the water displacement (bottom row) on the adaptive mesh, every 10 h.

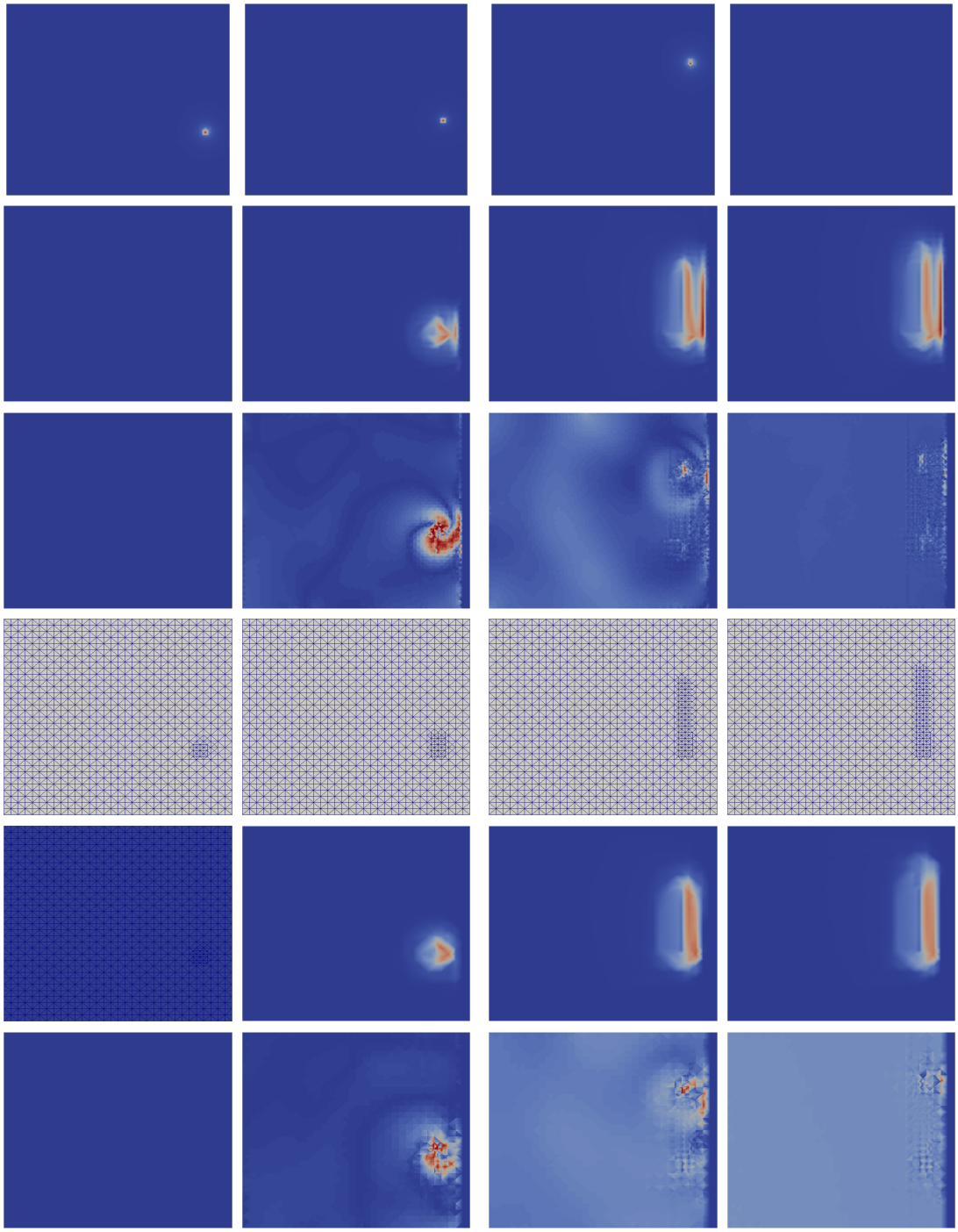

Figure 6.8.: Traveling Storm D: Uniform and adaptive simulation. Depicted are the storm location (top row), magnitude of velocity (second row), water displacement (third row), the adaptive mesh (fourth row), as well as the magnitude of velocity (fifth row) and the water displacement (bottom row) on the adaptive mesh, every 10 h.

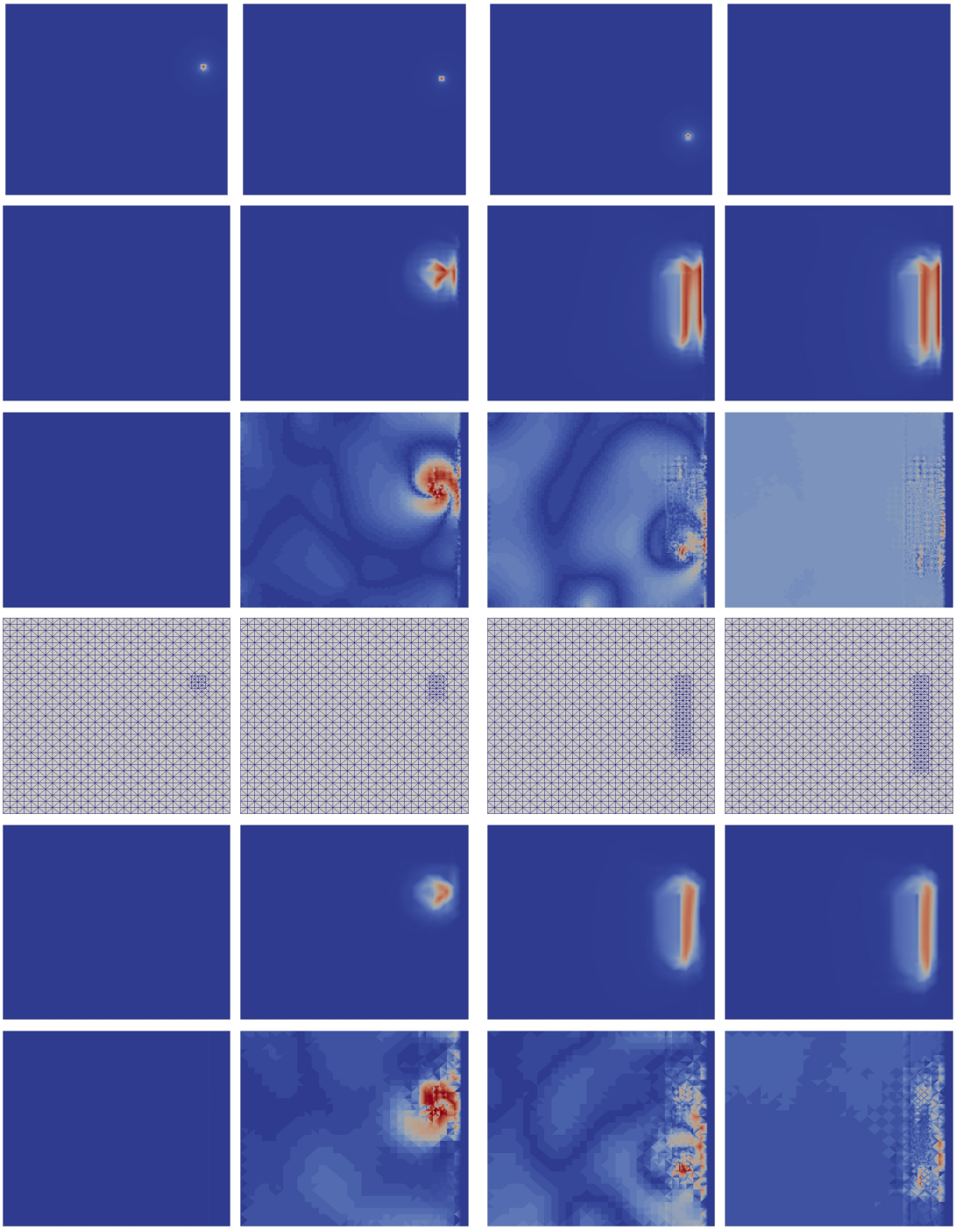

Figure 6.9.: Traveling Storm E: Uniform and adaptive simulation. Depicted are the storm location (top row), magnitude of velocity (second row), water displacement (third row), the adaptive mesh (fourth row), as well as the magnitude of velocity (fifth row) and the water displacement (bottom row) on the adaptive mesh, every 10 h.

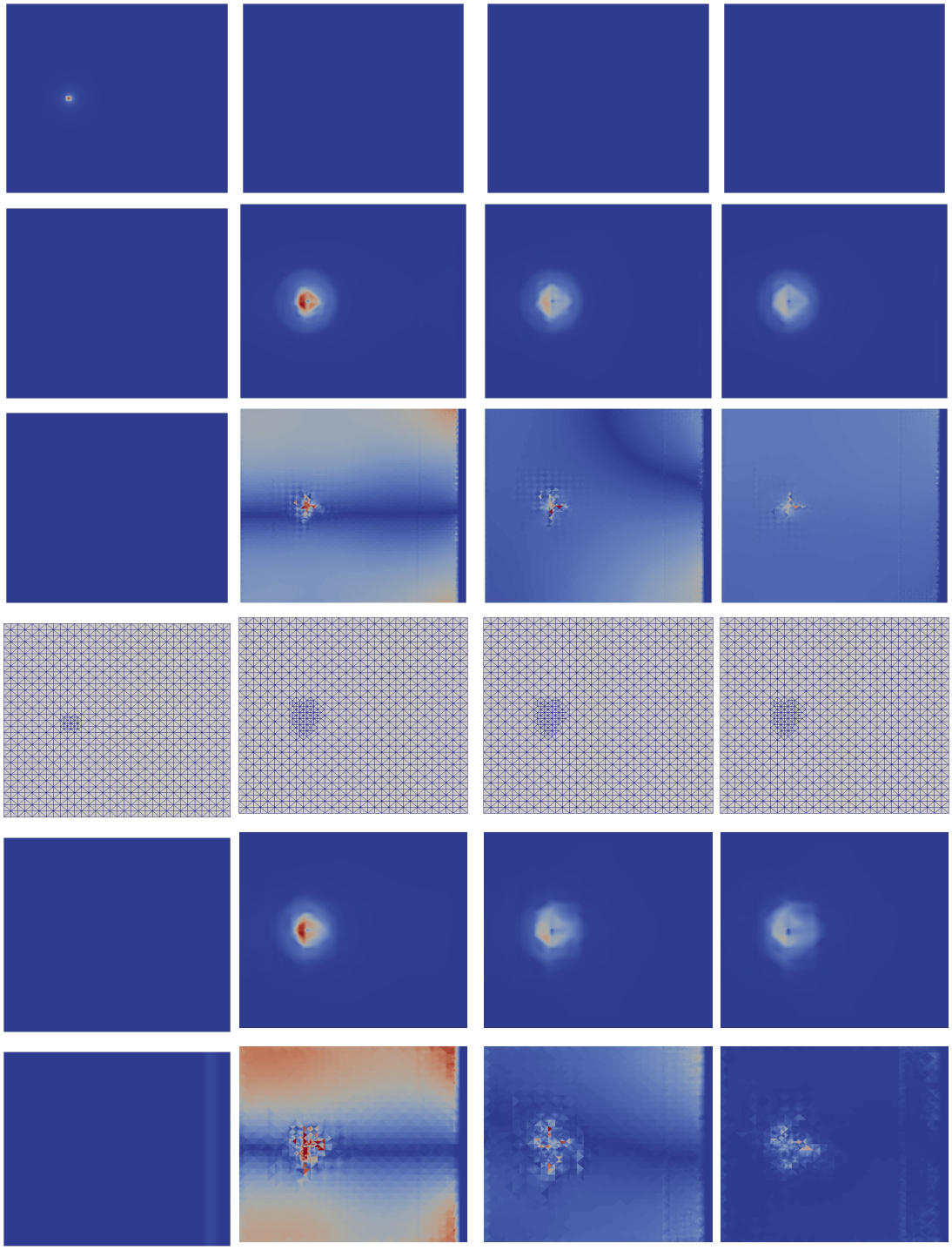

Figure 6.10.: Traveling Storm F: Uniform and adaptive simulation. Depicted are the storm location (top row), magnitude of velocity (second row), water displacement (third row), the adaptive mesh (fourth row), as well as the magnitude of velocity (fifth row) and the water displacement (bottom row) on the adaptive mesh, every 10 h.

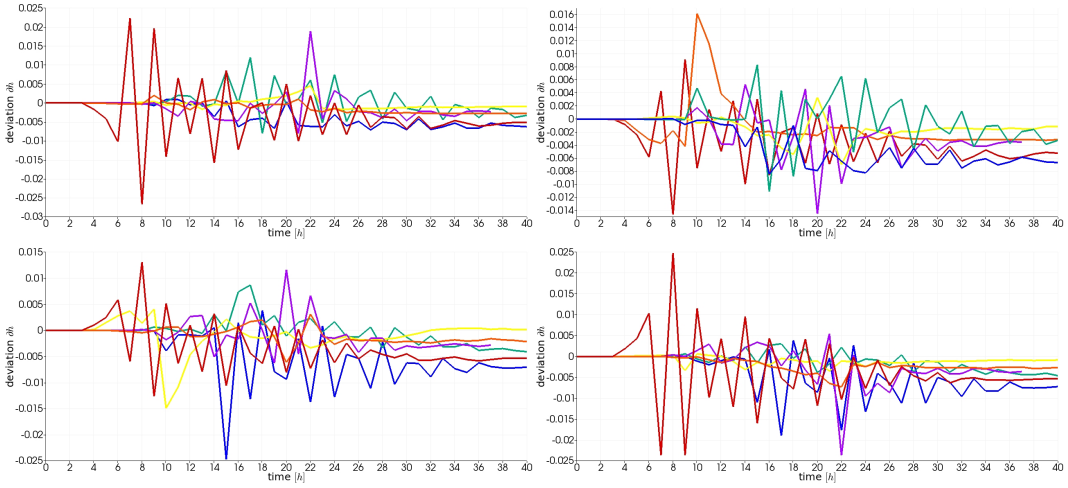

Figure 6.11.: Traveling Storms. Gauge Data recorded every hour for Storm A (violet), B (blue), C (green), D (yellow), E (orange) and F (red). Depicted are the water displacements at $\mathbf{G}_1$ (top left), $\mathbf{G}_2$ (top right), $\mathbf{G}_3$ (bottom left) and $\mathbf{G}_4$ (bottom right).

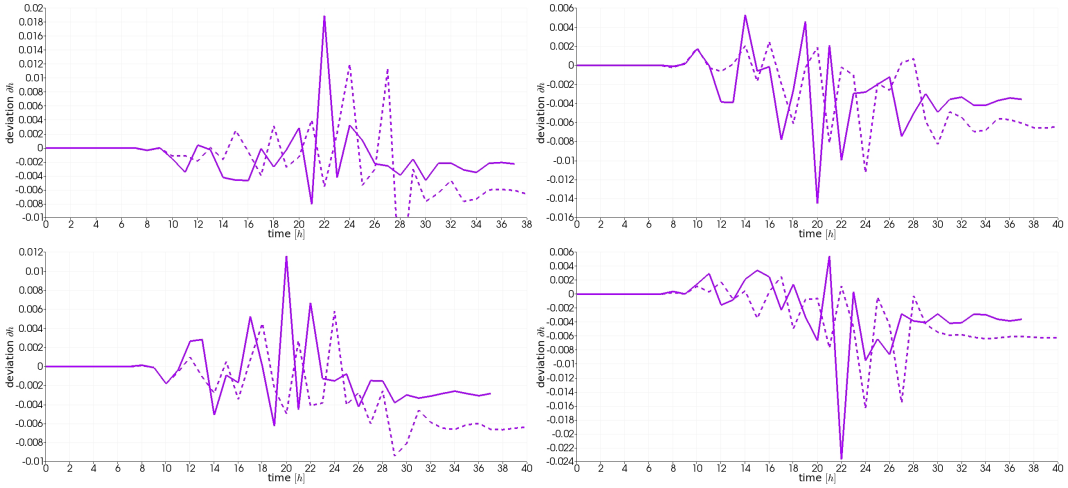

Figure 6.12.: Traveling Storms. Gauge Data for Storm A. Depicted are the deviation ∂h of the uniform (solid) and adaptive (dashed) simulation at $\mathbf{G}_1$ (top left), $\mathbf{G}_2$ (top right), $\mathbf{G}_3$ (bottom left) and $\mathbf{G}_4$ (bottom right).

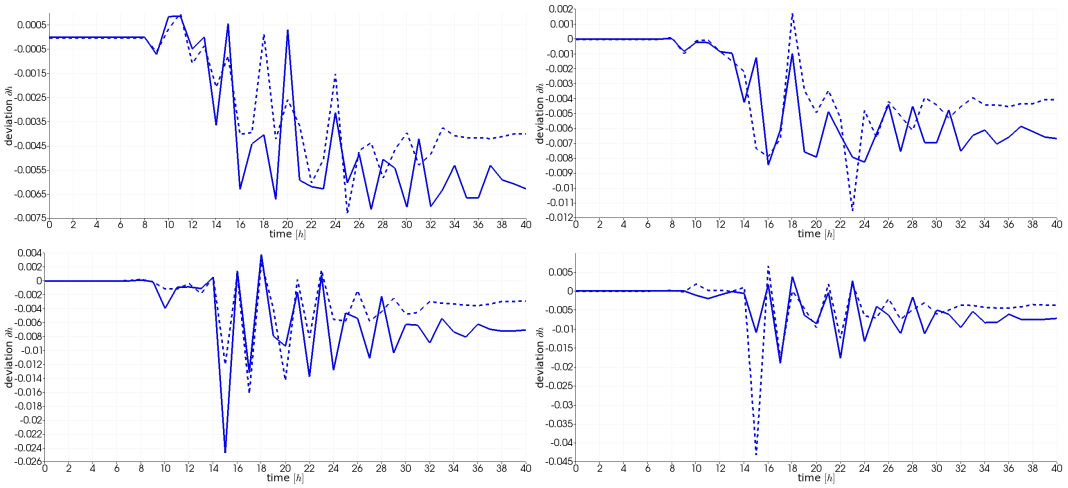

Figure 6.13.: Traveling Storms. Gauge Data for Storm B. Depicted are the deviation ∂h of the uniform (solid) and adaptive (dashed) simulation at $\mathbf{G}_1$ (top left), $\mathbf{G}_2$ (top right), $\mathbf{G}_3$ (bottom left) and $\mathbf{G}_4$ (bottom right).

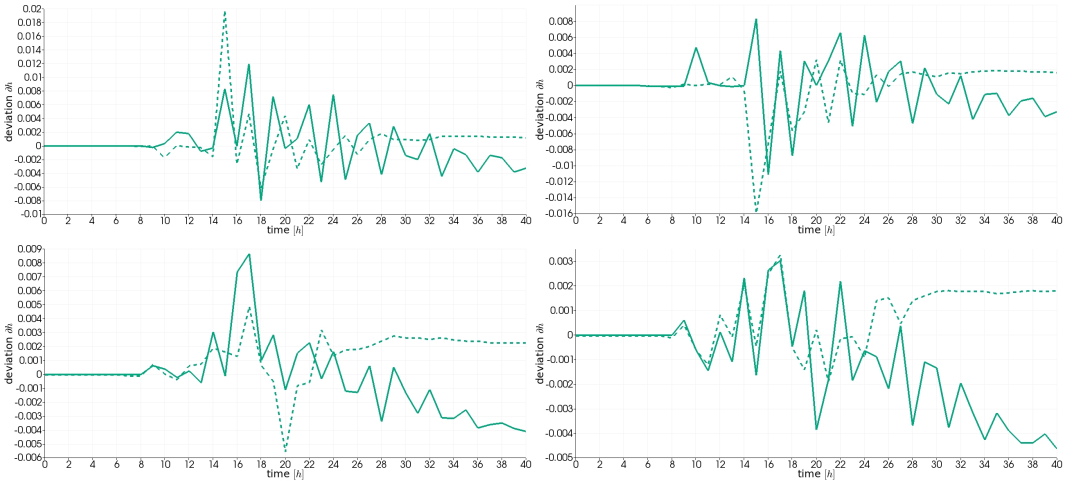

Figure 6.14.: Traveling Storms. Gauge Data for Storm C. Depicted are the deviation ∂h of the uniform (solid) and adaptive (dashed) simulation at $\mathbf{G}_1$ (top left), $\mathbf{G}_2$ (top right), $\mathbf{G}_3$ (bottom left) and $\mathbf{G}_4$ (bottom right).

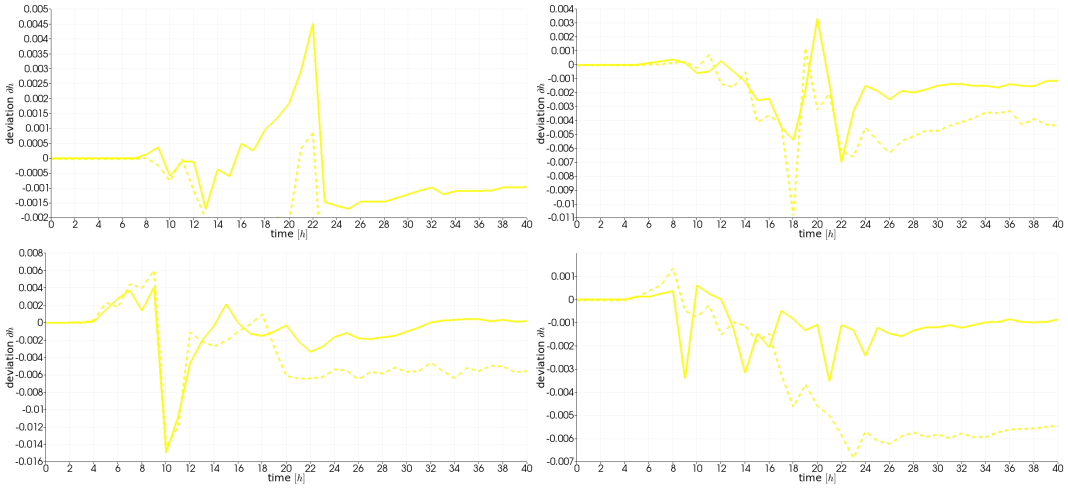

Figure 6.15.: Traveling Storms. Gauge Data for Storm D. Depicted are the deviation ∂h of the uniform (solid) and adaptive (dashed) simulation at $\mathbf{G}_1$ (top left), $\mathbf{G}_2$ (top right), $\mathbf{G}_3$ (bottom left) and $\mathbf{G}_4$ (bottom right).

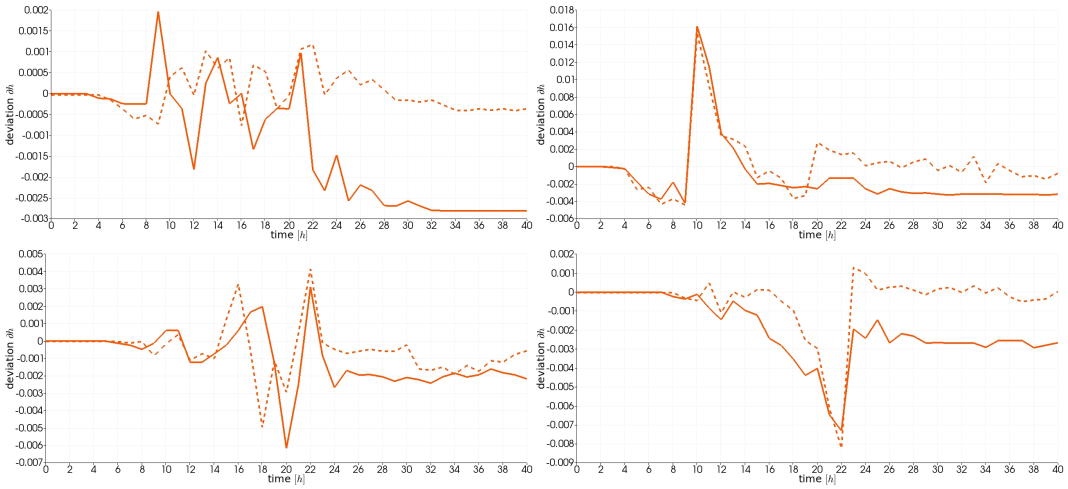

Figure 6.16.: Traveling Storms. Gauge Data for Storm E. Depicted are the deviation ∂h of the uniform (solid) and adaptive (dashed) simulation at $\mathbf{G}_1$ (top left), $\mathbf{G}_2$ (top right), $\mathbf{G}_3$ (bottom left) and $\mathbf{G}_4$ (bottom right).

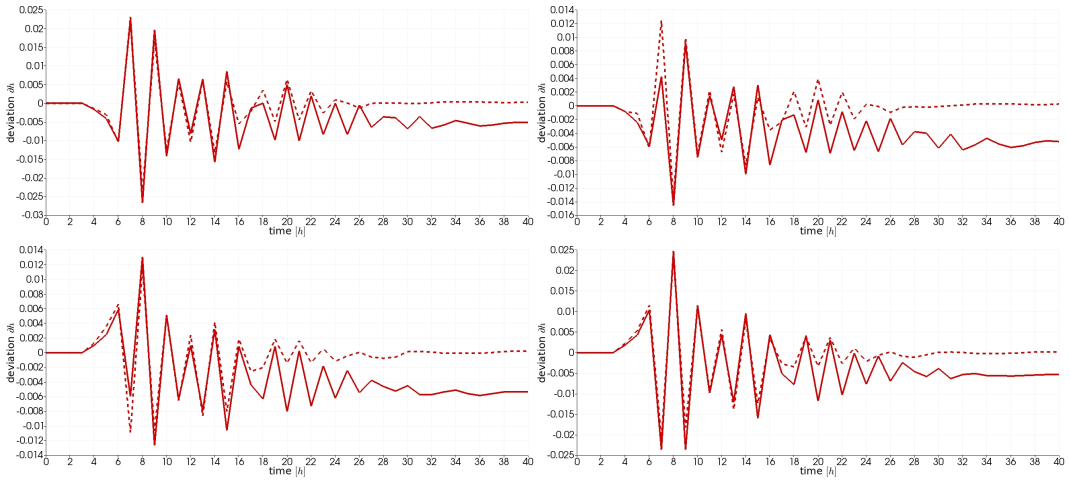

Figure 6.17.: Traveling Storms. Gauge Data for Storm F. Depicted are the deviation ∂h of the uniform (solid) and adaptive (dashed) simulation at $\mathbf{G}_1$ (top left), $\mathbf{G}_2$ (top right), $\mathbf{G}_3$ (bottom left) and $\mathbf{G}_4$ (bottom right).

time. The spatial resolution of the uniform simulations was $\Delta\mathbf{x}_{min} = 9375.0$ and for the adaptive simulations $\Delta\mathbf{x}_{min} = 9375.0$ and $\Delta\mathbf{x}_{max} = 21875.0$. The maximal number of elements for the adaptive simulations are stated in table 6.4. The uniform simulation contained 8192 elements.

We ramped up the wind forcing over a period of $8h$ in order to create a smooth velocity field. The ramping, though, was also the cause for the slowly decreasing velocity imprint at the start position $\text{pos}_{eye}(t = 0)$, that we observe and which fades with time due to bottom friction forcing. The test results of our model are illustrated in figures 6.5, 6.6, 6.7, 6.8, 6.9 and 6.10 for the six different storms. Depicted are the position of the storm, the magnitude of the velocities and the water displacement for the adaptive and uniform simulations as well as the corresponding adaptive mesh. The water displacement ∂h is computed as the deviation from the still water state: $\partial h(\mathbf{x}, t) = h(\mathbf{x}, t) - \max(0, 3000 - b(\mathbf{x}))$. For the Storms A-E, the mesh is dynamically refined in a neighborhood of the track of the storm, with a slightly larger extent perpendicular to the track, because this is the main direction of the emitted waves. From table 6.4, we see, that the maximum number of elements of the adaptive simulation did non exceed half the number of elements of the uniform simulation resulting in normalized CPU times between 0.2530 and 0.4075 depending on the scenario.

To assess the quality of the adaptive solution, we compared the gauge data from the adaptive with the uniform simulation. The corresponding figures are 6.12, 6.13, 6.14, 6.15, 6.16 and 6.17. We measured the deviation from the still water state every hour over the first 40h of simulation time. We observe, that the arrival time of the waves nearly match although the following waves do not always show a good agreement. In general a better match within the first 20 hours is noticeable. For the storms D and E traveling in y-direction, we notice, that the largest gap between adaptive and uniform simulation results occurs after approximately 20 hours at the gauge that is opposite to

the direction of the storm, which we attribute to the adaptive mesh, that follows the direction of the storm.

Problems with the adaptive mesh are visible in figure 6.10 for the fast storm F. The adaptive refinement is restricted to the initial state of the storm. This is due to the high speed of the storm. The transfer of movement from the wind stress onto the water column did not take place despite the large wind friction parameter $\gamma_{\boldsymbol{\tau}}$ (see table 6.4). We think, that this malfunction is due to the one-layer character of the model.

As mentioned in Mandli (2011), there were discussions after hurricane Ike in 2008, about the strong link between surge pattern and the angle with which the storm approached the coast. Figure 6.11 shows the gauge measurements for all uniform simulated storms. The different colors indicate the different storms. We observe, that they all show different wave patterns and that, for example, the waves generated by the fast storm F arrive at the gauges after around 3 hours, while the waves generated with the other storms usually take between 4 and 8 hours to first arrive at the gauges.

6.5. Concluding Remarks

We have presented the discretization of the essential source terms of storm surges: bottom friction, wind stress and Coriolis forcing. Furthermore, we defined a suitable composite refinement indicator for the dynamically adaptive mesh, that takes into account the location of the storm as well as the originating waves and the CFL stability condition. With two subsequent idealized storm tests, we validated the implementation of the source forcings and confirmed the robustness of our scheme on adaptive grids as well as the capability to handle quasi-realistic length scales. The observed storm surge behavior at four distinct measuring points in an idealized test case with different storm settings was plausible. We confirmed that the wave height depends on the angle with which the storm approaches land. Finally, we showed adaptive simulations of the same scenarios and were able to obtain almost the same arrival time of the first wave as well as similar wave behavior.

7. Discussion and Conclusions

The goal of this thesis was the methodological improvement of Discontinuous Galerkin (DG) methods for storm surge simulations with a focus on computational complexity and numerical accuracy. Therefore, we developed and implemented a nodal DG model, `StormFlash2d`, to solve the two-dimensional, one-layer and non-linear shallow water equations into the software framework `amatos`, which is a Fortran 90 library, that provides routines for grid creation and manipulation.

We performed idealized numerical experiments for flooding and drying scenarios to assess the robustness of the model as well as the gain of computational efficiency due to the usage of a dynamically adaptive mesh. To improve the accuracy of the simulations, we formulated a theoretical characterization of slope limiters and determined strengths and weaknesses of handling wetting and drying of the limiters, that we implemented in our model. Furthermore, we analyzed the viability of third-order basis functions for wetting and drying and developed different approaches, whose features we investigated in a numerical study. Finally, we implemented discretizations of source terms, that are relevant for storm surge modeling, such as the wind forcing. With an idealized set of storm test cases we determined a sensible refinement indicator for the adaptive mesh and showed the capability of our model to simulate storm surges.

In the following we sum up and discuss our major findings and their projections and close this final chapter with proposed research questions for future investigation.

7.1. Improvement of Computational Complexity and Numerical Accuracy

The main aim of methodological improvement of DG methods for storm surge simulations, which led us to the development and validation of `StormFlash2d`, was mainly guided by three independent research questions, that we formulated in section 1.2 and that we repeat here for reasons of clarity:

- To which extent does varying spatial resolution in a DG model reduce the computational complexity and, in turn, how does it affect the overall accuracy and robustness of the method, especially for flooding and drying scenarios?
- Does a theoretical formulation of general slope limiters for the reduction of numerical oscillations exist; how does it help to determine their strengths and weaknesses in handling flooding and drying and can the results be confirmed numerically?
- Are third-order basis functions viable for the simulation of wetting and drying?

In chapter 3, we combined the DG model with a triangular mesh, created with `amatos`, that is dynamically refined by bisection according to a user-defined refinement indicator. To study the effects on robustness, accuracy and computational complexity, we performed a set of numerical tests for flooding and drying, that we showed in section 3.4. The tests revealed that the method is robust on adaptive grids, which is probably due to the only local communication of elements over edges. Furthermore, we demonstrated that with the adaptive mesh the computational complexity could be reduced to up to 80% for the tidal beach tests with a loss of accuracy of only 1% due to the quasi one-dimensional nature of the problem. A truly two-dimensional paraboloid basin testcase revealed, that a decrease in computing time of $\approx 45\%$ lead to a loss of accuracy of around 25%. In all, our tests confirmed that the model is well-balanced and that the adaptive mesh led to an improvement of computational complexity.

Chapter 4 dealt with the numerical approximations in the DG model and their impact on accuracy and computational efficiency. Our study of existing slope limiters comprised a general theoretical characterization that showed the link between the different approaches and the determination of strengths and weaknesses of commonly used edge-and vertex-based limiters. In a subsequent numerical study we tested our implementation for robustness, convergence and conservation properties. The edge-based limiters were computationally the most expensive as opposed to simple L_2-limiter, which showed the least accurate performance, so that we conclude that the vertex-based limiter is preferable in combination with our DG model. Furthermore, we discussed Riemann solvers to compute fluxes over the element boundaries. According to our numerical study all the approximate Riemann solvers, that we tested, led to equivalent results. Finally, we presented a computationally inexpensive technique, that mainly depends on the use of the strong form of the underlying equations, to ensure well-balancedness of the model. Numerical tests confirmed that the implementation is robust and able to preserve partially dry steady states without introducing significant oscillations.

We investigated the viability of higher than second order basis functions in chapter 5 and demonstrated that the nodal Lagrange approach destabilizes the model for problems of flooding and drying. This led us to the development of a new strategy using Bernstein polynomials as test- and ansatz-functions. The monotonicity of Bernstein functions helped to circumvent problems with negative water height that occur for the Lagrange ansatz in the vicinity of wet/dry interfaces. The study of convergence properties comprised two wet/dry test cases with analytical solutions. The errors were measured in standard L_2- and L_1-norms both, for the global domain and for the partial smooth domains. We could show, that the use of higher third-order basis functions led to a global improvement of the convergence rate of 0.5 and a local improvement of approximately 1. Throughout the numerical error resulting from the Bernstein approach was smaller than the error of the Lagrange approach. The Bernstein polynomials showed better conservation properties than the modified and the second-order Lagrange approach.

We have presented the discretizations of the major source forcings of storm surges in chapter 6. Among others, this included an idealized storm model. Since the quality of the dynamically adaptive simulation strongly depends on the quality of the underlying mesh, we developed a composite refinement indicator, that takes the major aspects of

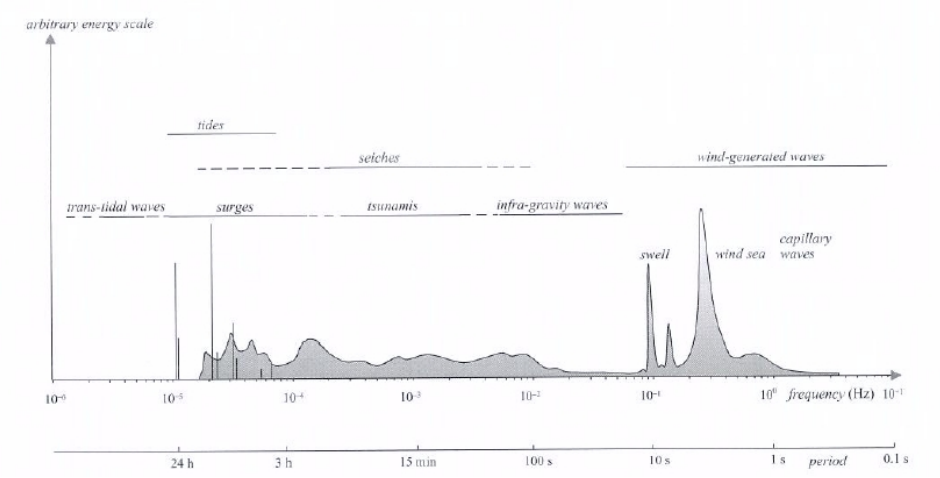

Figure 7.1.: Wave spectrum according to Holland (1980)

storm surges into account. With two idealized storm tests, we validated our implementation and were able to obtain realistic numerical results. The adaptive mesh was able to reduce the computational cost by a factor of ≈ 2 and the corresponding simulation results still showed good agreement, especially for the arrival time of the first wave at discrete points, with the uniform simulation.

7.2. Outlook and Future Work

The implementations and investigations of this thesis leave natural directions for future research and further work to be done.

The natural continuation of this work is the application to real data and the comparison of the results to real measurements. The acquisition of measurement data can, however, be difficult.

Also, an extension of the third-order Bernstein polynomials to higher order could be interesting. So far, we were content with the third-order polynomials for two reasons. As the code is not fully parallel, we will probably not be able to use the potential, that higher order functions offer and our primary goal is to develop a model for realistic applications, so that the extension to higher than third-oder basis functions was not prioritized. Further research could be conducted on the potential of Bernstein polynomials to suppress artificial oscillations in p-adaptive schemes. This was not possible to explore as currently the model `StormFlash2d` does not offer p-adaptivity.

The testcases, that we presented in chapters 3—6 are a good basis for a comparison with other existing models, that employ different numerical techniques or are implemented in a different software framework. This would help to gain insight into how for example continuous and discontinuous methods compare and identify further weaknesses of our model. One weakness that we already recognized is the run time of our model. The numerical tests suggested that especially for high-resolution applications,

we need to improve the run time of the code. A parallelization with openMP has already been initiated. However, more effort needs to be invested in order to obtain a practically feasible numerical code.

A multilayer model would have several advantages. The wind stress would then only affect the upper layer rather than the whole water column which would lead to more realistic results. Moreover the assumption of a constant density of the fluid would be relaxed in that case as at least for each layer one could easily assume different densities. However, the coupling of the layers is an interesting problem. For shallow water equations this has been done in Mandli (2011) for a finite volume scheme.

Furthermore, a non-hydrostatic correction would make it possible to relax the assumption made on hydrostasy in chapter 2 at the expense of solving a quadratic linear system of equations in every times step.

In our model we exclusively used an explicit time stepping. Reasons for their application are the computational simplicity and that they do not modify the speed of the gravity waves of the system which is important for computing the arrival times of waves onto the coast. However, in some regimes the restriction of the time step is rather severe. Therefore methods need to be explored that allow for a relaxation of the restriction. There are works on semi-implicit methods such as Giraldo and Restelli (2009) where a local linearization is performed to come up with a semi-implicit system. This allows for a larger timestep when we approve of solving a linear system in every time step with for example a GMRES method.

As depicted in figure (7.1) the waves of the system that we are solving can be roughly clustered into two categories: long and short, wind-generated waves which are both present in for example the modeling of hurricane storm surges. Since especially for the waves approaching the coast, the shallow water equations are not valid any more, it would make sense to work with a different Green-Naghdi equations based model to model the physical processes there.

Bibliography

Aizinger, V., 2011: A geometry independent slope limiter for the discontinuous galerkin method. *Computational Science and High Performance Computing IV*, E. Krause, Y. Shokin, M. Resch, D. Kröner, and N. Shokina, Eds., Springer, Notes on Numerical Fluid Mechanics and Multidisciplinary Design, 207–217.

Balzano, A., 1998: Evaluation of methods for numerical simulation of wetting and drying in shallow water flow models. *Coast. Eng.*, **34 (1-2)**, 83–107, doi: 16/S0378-3839(98)00015-5.

Bänsch, E., 1991: Local mesh refinement in 2 and 3 dimensions. *ICSE*, **3 (3)**, 181–191, doi: 10.1016/0899-8248(91)90006-G.

Barth, T. and D. Jespersen, 1989: The design and application of upwind schemes on unstructured meshes. *AIAA Paper*, **(89-0366)**.

Bastian, P., et al., 2012: Adaptive modelling of coupled hydrological processes with application in water management. *Mathematics in Industry*, **17**, progress in Industrial Mathematics at ECMI 2010.

Behrens, J., 2006: *Adaptive Atmospheric Modeling*. Springer.

Behrens, J. and M. Bader, 2009: Efficiency considerations in triangular adaptive mesh refinement. *Philos. T. Roy. Soc. A.*, **367 (1907)**, 4577 –4589, doi: 10.1098/rsta.2009.0175.

Behrens, J., N. Rakowsky, W. Hiller, D. Handorf, M. Läuter, J. Päpke, and K. Dethloff, 2005: amatos: Parallel adaptive mesh generator for atmospheric and oceanic simulation. *Ocean Modelling*, **10 (1-2)**, 171–183, doi: 10.1016/j.ocemod.2004.06.003.

Bell, J. B., C. N. Dawson, and G. R. Shubin, 1988: An unsplit, higher order godunov method for scalar conservation laws in multiple dimensions. *J. Comput. Phys.*, **74**, 1–24, issue 1.

Berg, R., 2010: Tropical cyclone report hurricane ike (al092008). Tech. rep., National Hurricane Center.

Blyth, M. G. and C. Pozrikidis, 2006: A lobatto interpolation grid over the triangle. *IMA Journal of Applied Mathematics*, **71 (1)**, 153 –169, doi: 10.1093/imamat/hxh077.

Bunya, S., E. Kubatko, J. Westerink, and C. Dawson, 2009: A wetting and drying treatment for the runge-kutta discontinuous galerkin solution of the shallow water equations. *Comput. Methods Appl. Mech. Engrg.*, **198**, 1548–1562.

Chen, C., J. Qi, C. Li, R. C. Beardsley, R. Lin, H. Walker, and K. Gates, 2008: Complexity of the flooding/drying process in an estuarine tidal-creek salt-marsh system: an application of FVCOM. *J. Geophys. Res.*, **113**, doi: 10.1029/2007jc004328.

Chow, V. T., 1959: *Open-channel hydraulics*. McGraw-Hill Book Co. New York.

Cockburn, B., 1998: An introduction to the discontinuous galerkin method for convection-dominated problems. *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, A. Quarteroni, Ed., Springer Berlin Heidelberg, Lecture Notes in Mathematics, Vol. 1697, 151–268, doi: 10.1007/BFb0096353, URL `http://dx.doi.org/10.1007/BFb0096353`.

Cockburn, B., G. E. Karniadakis, and C.-W. Shu, (Eds.) , 1999: *Discontinuous Galerkin Methods*. Springer.

Cockburn, B. and C. Shu, 1998: The Runge-Kutta Discontinuous Galerkin Method for Conservation Laws V: Multidimensional Systems. *J. Comput. Phys.*, **141 (2)**, 199–224, doi: 10.1006/jcph.1998.5892.

Dedner, A., D. Kröner, and N. Shokina, 2011: Adaptive modelling of two-dimensional shallow water flows with wetting and drying. *Computational Science and High Performance Computing IV*, E. Krause, Y. Shokin, M. Resch, D. Kröner, and N. Shokina, Eds., Springer Berlin Heidelberg, Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Vol. 115, 1–15.

Dumbser, M., 2011: A simple two-phase method for the simulation of complex free surface flows. *Comput. Method. Appl. M.*, **200**, 1204–1219.

Farin, G., 1986: Triangular bernstein-bezier patches. *Comput. Aided Geom. D.*, **3**, 83–127.

Fuchs, A., 2013: Effiziente parallele Verfahren zur Lösung verteilter, dünnbesetzter Gleichungssysteme eines nichthydrostatischen Tsunamimodells. Ph.D. thesis, University of Bremen.

Giraldo, F. X., J. S. Hesthaven, and T. Warburton, 2002: Nodal High-Order Discontinuous Galerkin Methods for the Spherical Shallow Water Equations. *J. Comp. Phys.*, **181**, 499–525, doi: 10.1006/jcph.2002.7139.

Giraldo, F. X. and M. Restelli, 2009: High-Order Semi-Implicit Time-Integration of a Triangular Discontinuous Galerkin Oceanic Shallow Water Model. *J. Num. Meths. Fluids*, **63**, 1077–1102.

Godunov, S., 1959: A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics. *Math. Sb.*, **47**, 357–393.

Gopalakrishnan, S. and F. X. Giraldo, 2014: A Wetting and Drying Algorithm for Coastal Ocean Modeling using Triangular Discontinuos Galerkin Methods. In preparation.

Gottlieb, S., 2005: On high order strong stability preserving rungekutta and multi step time discretizations. *Journal of Scientific Computing*, **25**, 105–128, doi: 10.1007/s10915-004-4635-5.

Heniche, M., Y. Secretan, P. Buodreau, and M. Leclerc, 2000: A two-dimensional finite element drying-wetting shallow water model for rivers and estuaries. *Advances in Water Resources*, **23**, 359–372.

Hermann, V., M. Käser, and C. E. Castro, 2011: Non-conforming hybrid meshes for efficient 2d wave propagation using the discontinuous galerkin method. *Geophys. J. Int.*, **184 (2)**, 746–758.

Hesthaven, J. S., 1998: From Electrostatics to Almost Optimal Nodal Sets for Polynomial Interpolation in a Simplex. *SIAM J. Numer. Anal.*, **35**, 655–676, doi: 10.1137/S003614299630587X.

Hesthaven, J. S. and T. Warburton, 2008: *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer.

Hindenlang, F., G. Gassner, T. Bolemann, and C.-D. Munz, 2010: Unstructured high order grids and their application in discontinuous galerkin methods. *Preprint Series, Stuttgart Research Centre for Simulation Technology (SRC SimTech)*, issue No. 2010-26.

Holland, G. J., 1980: An analytic model of the wind and pressure profiles in hurricanes. *Mon. Weather Rev.*, **108**, 1212–1218.

Holthuijsen, L. H., 2007: *Waves in Oceanic and Coastal Waters*. Cambridge.

Iskandarani, M., D. B. Haidvogel, and J. C. Levin, 2003: A three-dimensional spectral element model for the solution of the hydrostatic primitive equations. *Journal of Computational Physics*, **186 (2)**, 397–425, doi: 10.1016/S0021-9991(03)00025-1, ACM ID: 782531.

Kärnä, T., B. de Brye, O. Gourgue, J. Lambrechts, R. Comblen, V. Legat, and E. Deleersnijder, 2011: A fully implicit wetting-drying method for DG-FEM shallow water models, with an application to the scheldt estuary. *Comput. Method. Appl. M.*, **200 (5-8)**, 509–524, doi: 10.1016/j.cma.2010.07.001.

Kennedy, A. B., et al., 2012: Tropical cyclone inundation potential on the hawaiian islands oahu and kauai. *Ocean Modelling*, **52-53**, 54–68.

Kerr, P., et al., 2013: Surge generation mechanisms in the lower mississippi river and discharge dependency. *J. Waterw. Port. C-ASCE*, **139 (4)**, 326–335.

Kopera, M. A. and F. X. Giraldo, 2013: Analysis of adaptive mesh refinement for imex discontinuous galerkin solutions of the compressible euler equations with application to atmospheric simulations. *J. Comp. Phys.*, submitted.

Kuzmin, D., 2010: A vertex-based hierarchical slope limiter for p-adaptive discontinuous galerkin methods. *J. Comput. Appl. Math.*, **233**, 3077–3085.

Kuzmin, D., 2013a: Hierarchical slope limiting in explicit and implicit discontinuous galerkin methods. *J. Comp. Phys.*, submitted.

Kuzmin, D., 2013b: Slope limiting for discontinuous galerkin approximations with a possibly nonorthogonal taylor basis. *Int. J. Numer. Methods Fluids*, **71**, 1178–1190.

Leclerc, M., J.-F. Bellemare, G. Dumas, and G. Dhatt, 1990: A finite element model of estuarian and river flows. *Adv. Water Resour.*, **4**, 158–168.

LeVeque, R. J., 2002: *Finite-Volume Methods for Hyperbolic Problems*. Cambridge University Press.

Loubére, R., J. Ovadia, and R. Abgrall, 2004: A Lagrangian Discontinuous Galerkin-type method on unstructured meshes to solve hydrodynamics problems. *Int. J. Numer. Meth. Fluids*, **44**, 645–663.

Lynch, D. and W. Gray, 1978: Finite element simulation of shallow water problems with moving boundaries. *2nd Int. Conf. on Finite Elements in Water Resources*.

Mandli, K. T., 2011: Finite volume methods for the multilayer shallow water equations with applications to storm surges. Ph.D. thesis, University of Washington.

Mandli, K. T. and C. N. Dawson, 2014: Adaptive mesh refinement for storm surge. *Ocean Modelling*, in press.

McWilliams, J. C., 2006: *Fundamentals of Geophysical Fluid Dynamics*. Cambridge.

National Hurricane Center, N., 2012: The saffir-simpson hurricane wind scale. URL `http://www.nhc.noaa.gov/pdf/sshws.pdf`, last visited on Mar. 17, 2014.

NOAA, 2013: Storm surge overview. URL `http://www.nhc.noaa.gov/surge/images/stormsurgevsstormtide.jpg`, last visited on Feb. 28, 2014.

Thacker, W. C., 1981: Some exact solutions of the nonlinear shallow-water wave equations. *J. Fluid Mech.*, **107**, 499–508.

Toro, E. F., 2001: *Shock Capturing Methods for Free-Surface Shallow Flows*. Wiley.

Wahl, T., C. Mudersbach, and J. J., 2012a: Assessing the hydrodynamic boundary conditions for risk analyses in coastal areas: a multivariate statistical approach based on copula functions. *Nat. Hazard. Earth Sys.*, **12**, 495–510.

Wahl, T., C. Mudersbach, and J. J., 2012b: Assessing the hydrodynamic boundary conditions for risk analyses in coastal areas: a stochastic storm surge model. *Nat. Hazard. Earth Sys.*, **11**, 2925–2939.

Walters, R. A. and G. F. Carey, 1983: Analysis of Spurious Oscillation Modes for the Shallow Water and Navier Stokes equations. *Computer and Fluids*, **11**, 51–68.

Weisberg, R. H. and L. Zheng, 2006: Hurricane Storm Surge Simulations for Tampa Bay. *Estuar. Coast.*, **26 (6A)**, 889 – 913.

Westerink, J. J., et al., 2008: A Basin- to Channel-Scale Unstructured Grid Hurricane Storm Surge Model Applied to Southern Louisiana. *Mon. Weather Rev.*, **136**, 833–864, doi: 10.1175/2007MWR1946.1.

Xing, Y. and X. Zhang, 2013: Positivity-Preserving Well-Balanced Discontinuous Galerkin Methods for the Shallow Water Equations on Unstructured Triangular Meshes. *J. Sci. Comput.*, **57**, 19–41.

Xing, Y., X. Zhang, and C. Shu, 2010: Positivity-preserving high order well-balanced discontinuous galerkin methods for the shallow water equations. *Adv. Water Resour.*, **33 (12)**, 1476–1493, doi: 10.1016/j.advwatres.2010.08.005.

Zhang, X. and C. Shu, 2011: Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments. *Proc. R. Soc. A*, **467 (2134)**, 2752–2776, doi: 10.1098/rspa.2011.0153.

Zhang, Y. and A. A. Baptista, 2008: Selfe: A semi-implicit eulerian-lagrangian finite-element model for cross-scale ocean circulation. *Ocean Modelling*, **21**, 71–96.

Zhang, Y., A. M. Baptista, and E. P. Myers III, 2004: A cross-scale model for 3D baroclinic circulation in estuary-plume-shelf systems: I. Formulation and skill assessment. *Cont. Shelf Res.*, **24**, 2187–2214.

A. Appendix: Hurricane Wind Scale

The Hurricane Scale according to Saffir Simpson is a classification for storms that exceed wind speeds of $33 ms^{-1}$ and was officially released by the National Hurricane Center, NOAA in 1972. It is based on the sustained winds and helps to estimate the expected damage. It is defined as follows.

Category	Sustained winds $[ms^{-1}]$	Normal central pressure $[mbar]$	Example
1	33-42	980-994	
2	43-49	965-979	Hurricane Frances (2004)
3	50-58	945-964	Hurricane Ivan (2004)
4	58-70	920-944	Hurricane Charley (2004)
5	$\geq$70	< 920	Hurricane Andrew (1992)

Hurricanes of category 3 and higher are considered as major hurricanes because they are usually accompanied by major loss of life and property. The damages rise by approximately a factor of 4 between the categories. The categories, however, do not give any information about the to be expected storm surge. The latter is also strongly influenced by the size of the hurricane, bathymetry and topography of the area, forward speed of the hurricane and the angle with which the storm approaches the coast. A detailed description of the scale and corresponding expected extent of damage can be found in National Hurricane Center (2012)

B. Appendix: Implementation Issues

B.1. Derivation of Shallow Water Equations

We derive the shallow water equations from the Navier Stokes equations that are defined in McWilliams (2006) and are used to model the general movement of fluids:

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0 \tag{B.1a}$$

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \nabla \cdot \mathbf{\Phi} + \mathbf{F} \tag{B.1b}$$

where ρ is the density of the fluid, $\mathbf{u} = (u, v, w)^\top$ is the three-dimensional velocity, $\mathbf{x} = (x, y, z)^\top$ are the spatial coordinates, p is the pressure, ∇ is the gradient operator with respect to $\mathbf{x}$ and $\frac{D}{Dt}$ is the total derivative defined as $\frac{D}{Dt} := \frac{\partial}{\partial t} + \nabla \cdot \mathbf{u}$. The vector $\mathbf{F}$ is a vector of conservative body forces and $\mathbf{\Phi}$ contains all stress terms such as bottom friction or wind stress.

If we assume constant density $\rho = \rho_0 \in \mathbb{R}$, the first equation simplifies to

$$\nabla \cdot \mathbf{u} = 0 \tag{B.2}$$

We integrate (B.2) over the fluid height $h(\mathbf{x}, t) = H(\mathbf{x}, t) - b(\mathbf{x})$ as depicted in the sketch B.1 and assume no slip boundary conditions at the bottom boundary, i.e.

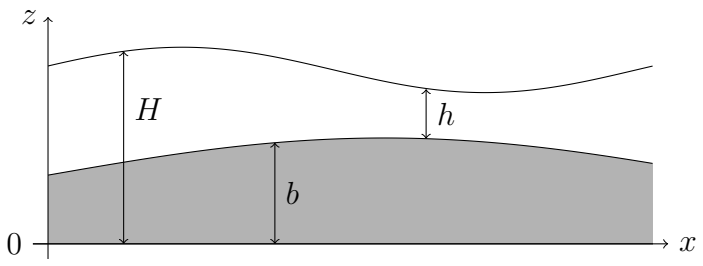

Figure B.1.: Shallow Water Equations. Sketch of bathymetry b, total height H and fluid height h.

$u = v = 0$ for $z = b$. Neglecting the vertical velocities yields

$$\begin{aligned} 0 &= \int_{b(\mathbf{x})}^{H(\mathbf{x},t)} \mathbf{u} \cdot \nabla \; dz \\ &= \frac{\partial}{\partial x} \int_{b(\mathbf{x})}^{H(\mathbf{x},t)} u \; dz - \left(u|_{z=H(\mathbf{x},t)} \frac{\partial h}{\partial x} \right) + \frac{\partial}{\partial y} \int_{b(\mathbf{x})}^{H(\mathbf{x},t)} v \; dz - \left(v|_{z=H(\mathbf{x},t)} \frac{\partial h}{\partial y} \right) \end{aligned}$$

Introducing the depth-averaged velocities $\overline{u}$ and $\overline{v}$:

$$\overline{u}(\mathbf{x},t) = \frac{1}{h(\mathbf{x},t)} \int_{b(\mathbf{x})}^{H(\mathbf{x},t)} u \; dz, \qquad \overline{v}(\mathbf{x},t) = \frac{1}{h(\mathbf{x},t)} \int_{b(\mathbf{x})}^{H(\mathbf{x},t)} v \; dz$$

and assuming no relative normal flow, i.e. $\frac{Dh}{Dt} = 0$ at $z = H$, yields the depth-averaged continuity equations

$$\frac{\partial h}{\partial t} + \frac{\partial (h\overline{u})}{dx} + \frac{\partial (h\overline{v})}{dy} = 0 \tag{B.3}$$

Now, the components of the momentum balances are

$$\frac{\partial u}{\partial t} + \frac{\partial (u^2)}{\partial x} + \frac{\partial (uv)}{\partial y} + \frac{\partial (uw)}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \frac{\partial \Phi}{\partial x} + \mathbf{F}_1 \tag{B.4a}$$

$$\frac{\partial v}{\partial t} + \frac{\partial (uv)}{\partial x} + \frac{\partial (v^2)}{\partial y} + \frac{\partial (vw)}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \frac{\partial \Phi}{\partial y} + \mathbf{F}_2 \tag{B.4b}$$

$$\frac{\partial w}{\partial t} + \frac{\partial (uw)}{\partial x} + \frac{\partial (vw)}{\partial y} + \frac{\partial (w^2)}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} + \frac{\partial \Phi}{\partial z} + \mathbf{F}_3 \tag{B.4c}$$

where $\mathbf{F}$ contains the conservative body forces such as gravitational acceleration g and Coriolis forcing $\mathbf{f}_c$, which has a zero third entry, and $\mathbf{\Phi}$ contains all the stress terms. If we assume small vertical velocities and the absence of stress terms in the vertical component, (B.4c) is reduced to

$$\frac{\partial p}{\partial z} = \rho g \quad \Rightarrow \quad p = \rho g (h + b - z),$$

which simplifies the pressure terms in the other two equations (B.4a) and (B.4b) to

$$\frac{\partial p}{\partial x} = \rho g \frac{\partial (h+b)}{\partial x}, \qquad \frac{\partial p}{\partial y} = \rho g \frac{\partial (h+b)}{\partial y}$$

which is the hydrostatic pressure distribution, that we will assume throughout. We integrate (B.4a) and (B.4b) over the fluid height, assume again that the vertical velocity is negligible , and obtain the depth-averaged momentum equations. We demonstrate

this exemplarily for (B.4a).

$$\int_{b(\mathbf{x})}^{H(\mathbf{x},t)} \left(\frac{\partial u}{\partial t} + \frac{\partial (u^2)}{\partial x} + \frac{\partial (uv)}{\partial y} \right) dz = \int_{b(\mathbf{x})}^{H(\mathbf{x},t)} \left(-\frac{1}{\rho} \frac{\partial p}{\partial x} + \frac{\partial \Phi}{\partial x} + \mathbf{F}_1 \right) dz$$
$$= \int_{b(\mathbf{x})}^{H(\mathbf{x},t)} \left(-g \frac{\partial (h+b)}{\partial x} + \frac{\partial \mathbf{\Phi}}{\partial x} + \mathbf{F}_1 \right) dz$$
$$\frac{\partial h\overline{u}}{\partial t} + \frac{\partial (h\overline{u}^2)}{\partial x} + \frac{\partial (h\overline{u}\overline{v})}{\partial y} = -gh \frac{\partial (h+b)}{\partial x} + h \frac{\partial \mathbf{\Phi}}{\partial x} + h\mathbf{F}_1$$

where we have used the definition of the depth-integrated velocities. The corresponding equation for the y-momentum then takes the form

$$\frac{\partial h\overline{u}}{\partial t} + \frac{\partial (h\overline{u}\overline{v})}{\partial x} + \frac{\partial (h\overline{v}^2)}{\partial y} = -gh \frac{\partial (h+b)}{\partial y} + h \frac{\partial \mathbf{\Phi}}{\partial y} + h\mathbf{F}_2$$

B.2. Convergence Study with Fully Wet Testcase

The convergence studies shown in this thesis solely comprised partially dry test problems for which the theoretical convergence order of $n+1$ for polynomials of degree n can not be achieved due to regularity problems. To complement the study, we tested convergence for a smooth and fully wet problem.

Testcase 11. *Linearized Sloshing Basin*
We consider a domain $\Omega = [0,1]^2$ with constant bathymetry, the linearized version of the shallow water equations and reflecting boundary conditions. The initial conditions are defined for $t = 0$ as follows:

$$h(\mathbf{x},t) = \cos(\pi x)\cos(\pi y)\cos(\sqrt{2}\pi t)$$
$$u(\mathbf{x},t) = \frac{1}{\sqrt{2}}\sin(\pi x)\cos(\pi y)\sin(\sqrt{2}\pi t)$$
$$v(\mathbf{x},t) = \frac{1}{\sqrt{2}}\cos(\pi x)\sin(\pi y)\sin(\sqrt{2}\pi t)$$

An analytical solution can be found in Iskandarani et al. (2003). We ran the test with a second order Runge Kutta scheme (RK22) and decreased the spatial resolution by bisection from $\Delta\mathbf{x}_{min} = 0.125$ to $\Delta\mathbf{x}_{min} = 0.03125$. The results are depicted in figure B.2, where we employ a double logarithmic scale to increase the readability, and show the expected theoretical convergence rate.

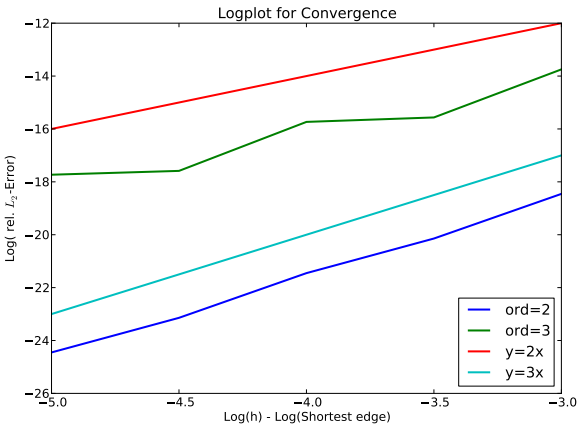

Figure B.2.: Convergence of DG Method