A High Order Finite Volume Scheme for the 2D Shallow Water Equations Including Topography

Dissertation

zur Erlangung des akademischen Grades einer
Doktorin der Naturwissenschaften (Dr. rer. nat.)

im Fachbereich Mathematik und Naturwissenschaften
der Universität Kassel

vorgelegt von
Bettina Charlotte Messerschmidt
aus Hanau

Gutachter:	Prof. Dr. Andreas Meister, Universität Kassel Prof. Dr. Armin Iske, Universität Hamburg
Tag der Disputation:	12. Juli 2012

Contents

Zusammenfassung

In der vorliegenden Dissertation wird die Konstruktion eines Verfahrens zur numerischen Lösung der zweidimensionalen Flachwassergleichung mit topographieinduzierten Quelltermen beschrieben, welches auch das Fluten trockener Gebiete sowie das Trockenfallen nasser Gebiete beherrscht. Es ist von beliebig hoher Ordnung in Raum und Zeit und erhält bestimmte Arten von stationären Lösungen, sogenannte "lake-at-rest" Zustände. Dies ist möglich, da in diesem Fall die numerischen Flüsse über die Zellgrenzen genau den Integralen der Quellterme entsprechen. Verfahren mit dieser Eigenschaft heißen "well-balanced" .

Basis des Verfahrens ist der Finite-Volumen-Ansatz auf dem sekundären Netz einer Delaunay-Triangulierung. Die Ordnung wird durch eine kombinierte Raum-Zeit Diskretisierung erzielt.

Hierbei wird zunächst, auf Basis der durch das Verfahren zur Verfügung gestellten Zellmittelwerte zum aktuellen Zeitpunkt t^n, für jede Zelle σ_i ein Rekonstruktionspolynom für jede der drei zu betrachtenden Größen $u_k(\mathbf{x}, t)$, $k = 0, 1, 2$, berechnet. Die Polynombasis hierfür besteht aus den Monomen $(\mathbf{x} - \mathbf{b}_i)^{\boldsymbol{\alpha}}$, wobei $\mathbf{b}_i$ den Schwerpunkt der Zelle σ_i bezeichnet. Die Ableitungen der Polynome im Punkt $\mathbf{b}_i$, welche später benötigt werden, können so sehr einfach aus deren Koeffizienten berechnet werden. Durch die Berechnung der Polynome mittels eines WENO-Ansatzes sollen auftretende Oszillationen minimiert werden. Die Berechnung der für die Gewichtung benötigten Polynome wird mit der Methode der kleinsten Quadrate durchgeführt, wobei durch eine Nebenbedingung die Erhaltung des Zellmittelwertes auf σ_i sichergestellt wird. Es wird in der Arbeit bewiesen, dass die Koeffizienten der so berechneten Rekonstruktionspolynome die Ableitungen der zu rekonstruierenden Funktion in $\mathbf{b}_i$ mit einem zur Verfahrensordnung passenden Genauigkeitsgrad approximieren. Die Gewichtung selbst erfolgt, wie in [DuK07] vorgeschlagen, in charakteristischen Variablen und mit dem dort genannten Oszillationsindikator.

In einem weiteren Schritt wird auf jeder Zelle eine Raum-Zeit Taylorentwicklung um den Punkt $(\mathbf{b}_i, t^n)$ für jede Größe u_k berechnet, wobei die

rein räumlichen Ableitungen durch die Ableitungen der Rekonstruktionspolynome ersetzt werden. Alle Koeffizienten der Taylorpolynome die Zeitableitungen enthalten, werden mittels der Cauchy-Kovalewskaja-Prozedur durch sukzessives Ableiten der Differentialgleichung aus Termen mit Zeitableitungen niedrigeren Grades berechnet. Ist *ord* nun die gewünschte Verfahrensordnung, so wird in der Arbeit bewiesen, dass aufgrund der Approximationseigenschaften der Koeffizienten der Rekonstruktionspolynome für die Taylorpolynome

$$U_{i;k}(\mathbf{x},t) - u_k(\mathbf{x},t) = \mathcal{O}(h^{ord}) \text{ für } (\mathbf{x},t) \in \sigma_i \times [t^n, t^n + \Delta t]$$

gilt. Hierbei stellt h ein eindimensionales Maß für die Feinheit des Netzes dar und Δt den von h linear abhängigen Zeitschritt. Dieser Ansatz wurde in [LaW60, HEO87] vorgestellt und in [GLM07, GLM08] für Discontinuous-Galerkin-Verfahren weiter entwickelt. Der Vorteil dieses Vorgehens liegt darin, dass die Quellterme über die Differentialgleichung direkt in die Diskretisierung mit einbezogen werden können, obwohl sie später noch separat integriert werden müssen.

Die Berechnung der numerischen Flüsse über die Zellränder findet an den Gaußpunkten $(\mathbf{x}_{ij}^{k;l}, t^m)$ jeder Raum-Zeitfläche $\mathbf{l}_{ij}^k \times [t^n, t^n + \Delta t]$, $k = 1, 2$, statt. $\mathbf{l}_{ij}^k$ bezeichne hier die zwei Kanten zwischen den Zellen σ_i und σ_j. An diesen Punkten werden die Taylorpolynome beider angrenzender Zellen ausgewertet und die numerischen Flüsse mittels eines Riemannlösers bestimmt.

Nimmt man an dieser Stelle den HLLC Riemannlöser und betrachtet die Erhaltungsgleichung ohne Quellterme, so liegt ein Finite-Volumen-Verfahren beliebig hoher Ordnung, je nach Grad der Rekonstruktionspolynome, vor, welches den zeitlichen Verlauf der Lösung bei gegeben Anfangs- und Randwerten berechnet.

Die Berücksichtigung von Quelltermen topographischen Ursprungs kann nun durch kleinere Anpassungen und die Verwendung eines speziellen Riemannlösers realisiert werden. Voraussetzung hierfür ist, dass die Topographie $top(\mathbf{x})$ nicht zeitabhängig ist. Der Term $\partial_t top = 0$ kann als weitere Gleichung in das System von Differentialgleichungen eingegliedert werden. Die Gleichungen bleiben damit hyperbolisch, die Wellenstruktur verändert sich allerdings in so weit, dass eine weitere Kontaktunstetigkeit mit Geschwindigkeit $S_4 = 0$ hinzukommt, während die restlichen Charakteristiken unverändert bleiben. Betrachtet man demnach das Riemannproblem an der Grenze zwischen zwei Zellen, so liegt die neue Kontaktunstetigkeit genau auf der Zellgrenze.

Zunächst wird ein Rekonstruktionspolynom $u_{i;3}$ zur Approximation von *top* für jede Zelle σ_i sowie eine Größe $K_{ij}^{k;l} = 9.81(u_{i;3} - u_{j;3})(\mathbf{x}_{ij}^{k;l})$ proportional zur Größe der durch die zellweise Rekonstruktion bedingten Unstetigkeit in den Punkten $\mathbf{x}_{ij}^{k;l}$ bestimmt. Da $top(\mathbf{x})$ nicht zeitabhängig ist, muss dies nur einmal, zu Beginn der Rechnung, erfolgen.

In jedem Zeitschritt wird nun nicht mehr die Größe u_0, welche die Wasserhöhe bezeichnet, sondern $u_0 + top$, also die Lage der Wasseroberfläche, rekonstruiert, und bei der Berechnung der Taylorentwicklung werden die Quellterme, wie bereits oben angedeutet, mit einbezogen. Des Weiteren können diese in der rekonstruierten Form, als Produkt von Polynomen auf der Zelle, sehr einfach integriert werden.

Die größten Änderungen finden im Bereich des Riemannlösers statt: Hier wird der Löser aus [ChL99, Seg99] eingesetzt, welcher die durch die Inklusion der Topographie entstehende Kontaktunstetigkeit zum Eigenwert $\lambda_4 = 0$ mit einbezieht. Es wird in der Arbeit bewiesen, dass das resultierende Verfahren, in Kombination mit der geänderten Rekonstruktion und der geänderten Taylorentwicklung, stationäre Zustände mit einer konstanten Höhe der Wasseroberfläche bei stillstehendem Wasser erhält.

Der Riemannlöser ist zudem in der Lage mit "trockenen" Zuständen, sowohl wenn sie als Anfangswert im Riemannproblem gegeben sind, als auch wenn sie erst in dessen Lösung entstehen, umzugehen. Um die Verarbeitung dieser Zustände im Gesamtverfahren zu ermöglichen, wird in Regionen des Rechengebietes, in denen die Wasserhöhe gering im Vergleich zur Zellgröße ist, die Verfahrensordnung sukzessive zuerst auf zwei und, falls nötig, weiter auf eins reduziert.

Obwohl die Techniken für die Rekonstruktion aus [DuK07] stammen, werden sie dort direkt auf die Triangulierung angewendet, was das Verfahren durch die Möglichkeit der Transformation auf ein Referenzdreieck und die mögliche Wahl einer auf diesem Dreieck orthogonalen Basis deutlich vereinfacht. In [DuM07] wird der Quellterm in die Cauchy-Kovalewskaja-Prozedur einbezogen, allerdings ist das resultierende Verfahren nicht well-balanced.

Das von Gallardo, Parés und Castro in [GPC07] vorgestellte Finite-Volumen-Verfahren ist well-balanced, von dritter Ordnung in nassen, glatten Regionen und kann trockene Zustände verarbeiten. Die Autoren nutzen eine Roe-Methode zusammen mit einer hyperbolischen Rekonstruktion. Das Verfahren wird im zweidimensionalen Fall auf Gitter mit viereckigen, nicht notwendigerweise uniformen, Zellen angewendet. In [CGL08] wird eine Godunov-Methode vorgestellt, allerdings für Systeme mit linearer Flussfunktion und in einer Dimension.

Xing und Shu präsentieren in [XiS11] und den dort genannten vorausgehenden Arbeiten ein hochgenaues Finite-Volumen-Verfahren, welches well-balanced ist, die Positivität der Lösung garantiert und eine WENO-Rekonstruktion beinhaltet. Zur Zeitdiskretisierung verwenden die Autoren eine

TVD-Runge-Kutta-Methode dritter Ordnung. Das Verfahren ist in zwei Dimensionen allerdings auf rechteckige Gitter beschränkt, die Positivität der Lösung wird durch die Reduktion der CFL-Zahl auf 0.08 sichergestellt und erfordert darüber hinaus die Bestimmung der Minima der Rekonstruktionspolynome für die Wasserhöhe auf den Zellen.

Weitere Arbeiten, etwa [BoM10, NPP06], gehen entweder nur auf den eindimensionalen Fall ein oder setzen für den zweidimensionalen Fall Netze mit rechteckigen Zellen voraus.

Der Vorteil des hier präsentierten Verfahrens im Vergleich zu oben genannten liegt darin, dass die Struktur des Netzes nur in technischen Abläufen, wie etwa der Bestimmung der Stencils für die Rekonstruktion, ausgenutzt wird. Die Gestalt der Zellen hat aber keinen direkten Einfluss auf die Berechnung der Flüsse oder der Rekonstruktionspolynome. Allerdings sind aufgrund der Gestalt der Zellen, man kann unregelmäßige Zehn- bis 14-ecke erwarten, keine Aussagen über die Positivität der Wasserhöhe nach der Berechnung der Flüsse möglich. Während des Rekonstruktionsprozesses kann durch die Reduktion der Ordnung allerdings sehr wohl Positivität erwartet werden.

In dieser Arbeit werden die Approximationseigenschaften der Koeffizienten der räumlichen Rekonstruktionspolynome bezogen auf die räumlichen Ableitungen der zu rekonstruierenden Funktionen bewiesen. Es wird weiter gezeigt, dass diese Eigenschaften durch die Cauchy-Kovalewskaja Prozedur auf die Koeffizienten der Raum-Zeit Taylorpolynome übertragen werden, welche somit die oben bereits genannte Approximationsordnung besitzen. Die Güte der räumlichen Rekonstruktion wird darüber hinaus numerisch nachgewiesen, ebenso wie die Genauigkeitsordnung des Gesamtverfahrens für die lineare Advektionsgleichung und die zweidimensionale Flachwassergleichung.

Des Weiteren wird bewiesen, dass das Verfahren die "well-balanced"-Eigenschaft für "lake-at-rest" Zustände besitzt, vorausgesetzt die Rekonstruktion und die Berechnung der Taylorpolynome wurden in der oben beschriebenen Weise angepasst. Auch diese Eigenschaft wird numerisch belegt.

Zusammenfassend lässt sich sagen, dass das hier vorgestellte Verfahren, bei beliebig hoher Genauigkeitsordnung in Gebieten mit genügend großer Wasserhöhe, die größtmögliche Allgemeinheit in Bezug auf die zu betrachtende Topographie erlaubt und dabei bestimmte stationäre Zustände erhält.

Introduction

This thesis is concerned with the numerical simulation of the two dimensional shallow water equations. These equations constitute a hyperbolic balance law of the form

$$\partial_t \mathbf{u} + \partial_{x_1} \mathbf{f}_1(\mathbf{u}) + \partial_{x_2} \mathbf{f}_2(\mathbf{u}) = \mathbf{g}(\mathbf{u}).$$

The two dimensional shallow water equations are a model for the flow behavior of water bodies. The model is valid for water bodies whose depth is very small compared to their surface dimensions and it neglects the effects of viscosity.

The vector $\mathbf{u}(\mathbf{x}, t)$ herein contains the quantities whose evolution in time, from a given initial state $\mathbf{u}(\mathbf{x}, t^0) = \mathbf{u}^0(\mathbf{x})$, is sought-after on the bounded domain of integration $\Omega \subset \mathbb{R}^2$. These quantities are, in the case of the two dimensional shallow water equations, the Geo potential Φ, which is the product of the water height and the gravity constant g, and Φv_i, $i = 1, 2$. The latter is called the momentum and is the product of Φ and the velocity v_i in the space direction x_i, $i = 1, 2$. The momentum can be interpreted as the product of the flow rate and g.

The equations generally describe the gravity induced time evolution of water flows with a free surface for given initial conditions. This class contains problems like the behavior of waves on shallow beaches or flood waves in rivers. These examples show that the treatment of bottom topography as well as dry zones is an interesting feature in this context.

Being not able to include topography into the scheme means all geometry that needs to be considered has to be contained in the grid. There, it is embodied as areas that are, in the context of the scheme, bounded by infinitely high impermeable walls normal to the flat horizontal ground. This severely restricts the range of structures that can be considered as it would not be possible to represent any topographical structures like beach slopes or under water trenches and humps.

The aim of this thesis is to present a numerical scheme whose solution for given initial conditions is of, theoretically, arbitrary high order in smooth and wet regions of the solution and that can cope with source terms due to topography as well as with dry regions.

In recent years, several high-order methods for the shallow water equations have been proposed that can handle these problems, like [BoM10, GPC07, CGL08, NPP06, XiS11]. However, most of them either only treat the one-dimensional case or are of at most second order. Schemes for the two-dimensional shallow water equations that are of at least third order are from Gallardo et al.,[GPC07], and Xing and Shu, [XiS11]. The first is of third order in wet, smooth regions and is restricted to grids with quadrilateral cells. The latter can theoretically be extended to arbitrary high order, but is restricted to grids with rectangular cells. The scheme presented in this work is of theoretically arbitrary high order and the requirement of a grid with triangular cells stems from the organization of the computation, but it is not relevant for the scheme itself.

This thesis is divided into four main chapters. The first contains a compilation of all the theoretical aspects regarding the numerical scheme to be developed in the second chapter. The main source is the book of Godlewski and Raviart, [GoR96], but also results from other works like [MeS02, MRT05, Tor99, Tor01] are cited or applied.

The leading equations are derived with the limitations of the model they describe being named. Afterwards, their properties relevant to the scheme are discussed. The property of rotational invariance is introduced briefly. This property is very useful, as it allows to rotate the equation in a way that the flux only in one direction, instead of two, needs to be considered. The analysis of the equations as well as the numerical scheme can thus be simplified considerably.

The most important property, hyperbolicity, comes next. Hyperbolicity means basically that the system matrix

$$\mathbf{F}(\mathbf{u},\omega) = \mathbf{f}_1'(\mathbf{u})\omega_1 + \mathbf{f}_2'(\mathbf{u})\omega_2 \text{ with } |\boldsymbol{\omega}| = 1$$

can be diagonalized. Hyperbolicity on the one hand is the source of problems that arise in designing a numerical scheme, as for nonlinear systems of partial differential equations having this property even smooth initial data may develop discontinuities in finite time. On the other hand, hyperbolicity provides an ansatz to handle this problem: The solution of Riemann problems, which are special initial value problems, splits into a sequence of constant states separated by waves traveling with a velocity depending on the initial data. Thus, the leading idea of the numerical scheme is to consider the integral form of the two dimensional shallow water equations, which allows discontinuous weak solutions, and solve local Riemann problems. In order to do so, the wave structure of the solution of Riemann problems, that depends on the eigenvalues and eigenvectors of $\mathbf{F}(\mathbf{u},\omega)$, is analyzed in detail.

The last section of this chapter is concerned with the introduction of the basis of the numerical scheme, the finite volume approach. This ansatz is based on the integral form of the equation. It considers the time evolution of the spatial integral mean values of the cells of a given computational grid. The updates for the cell mean values from the time t^n to t^{n+1} are computed by integrating the inter cell fluxes given by $\mathbf{f}_1(\mathbf{u})$ and $\mathbf{f}_2(\mathbf{u})$. The problems arising from the aim to construct a high order numerical scheme based on this approach are discussed at the end of the first chapter and motivate the development of the numerical scheme in the second.

The second chapter contains the derivation of the numerical scheme. It starts with the presentation of the setting used further in this chapter, including the introduction of program constants and the computational grid as well as its properties. The grid is the division of the integration area Ω into cells σ_i.

The next section treats a crucial part of the numerical scheme, the reconstruction. In this step of the scheme, higher order spatial polynomials for each cell are computed from the cell mean values of $\mathbf{u}$. The necessity to do so evokes from the aim of creating a numerical scheme of a high order of exactness, and thus the demand of a high order integration of the fluxes over the cell boundaries, while having only mean values from each cell at disposition. In the simplest possible reconstruction, these mean values can be interpreted as constant polynomials. The difficulties in reconstructing the exact solution by higher order polynomials using the cell mean values come from the fact that near discontinuities these polynomials often oscillate, which was proven by Godunov in [God59]. Many authors have published their analysis and their solution for this problem, see for example [Abg94, DuK07, DKT07, Fri98, Fri99, HaC91, ShO88, Son97] and the works mentioned therein.

The WENO reconstruction, that is used in this work was introduced in [LOC94]. It copes with the problem of oscillation in computing a set of polynomials for each cell σ_i that are each based on a different set of cells, or stencil, containing σ_i. These stencils are situated around σ_i and have the trait that they provide from the mean values of $\mathbf{u}$ of the cells they contain unique polynomials of a preset degree that have the same mean value, at least on σ_i. The reconstruction polynomials are computed with a least squares method. In this work, it is proven that the coefficients of these polynomials approximate the spatial derivatives of $\mathbf{u}$ in the cells' barycenters up to an order that fits with the order of the scheme. To the authors knowledge, this result was not proven before in the case of using a least squares method. As an implication, the reconstruction polynomials approximate $\mathbf{u}$ at the time t^n on the whole cell σ_i with the desired order. From these polynomials a weighted sum is built, with the weights depending on the reverse of a function indicating the measure of oscillation of each polynomial. If a discontinuity is situated in one stencil, the polynomial computed from this stencil will oscillate. Thus, the oscillation indicator will be high and as a consequence the polynomial will have only a small share in the weighted sum.

In the following section, the computing of a space time expansion approximating $\mathbf{u}$ on a space time cell $\sigma_i \times [t^n, t^{n+1}]$ is presented. To obtain highly accurate approximations to the integral of the inter cell fluxes, the time integration needs to be accurate as well. The cell mean values, and as a consequence the reconstruction polynomials for $\mathbf{u}$ as well, are situated at the time level t^n, while for the integration using numerical quadrature rules values at the integration nodes at later time levels are needed. The ansatz proposed in [HEO87, HaO87] and further developed for example in [GLM07, GLM08] makes use of the fact that the shallow water equations give a description of the time derivatives of $\mathbf{u}$ in terms of the spatial derivatives of $\mathbf{f}_i(\mathbf{u})$, $i = 1, 2$. Thus, having approximations for the spatial derivatives of $\mathbf{u}$ provided by the reconstruction polynomials up to a preset order, by deriving the complete differential equation in space as well as in time approximations to higher time derivatives as well as for higher mixed time space derivatives of $\mathbf{u}$ can be computed successively. Using these derivatives, a space time Taylor expansion of the preset degree around the barycenter of σ_i at the time t^n can be determined. This Taylor expansion can be evaluated at the nodes of the quadrature rule. Using the results obtained in the last section concerning the coefficients of the reconstruction polynomial, it is proven in this work that the coefficients of the Taylor expansion approximate the time- and time-space derivatives of $\mathbf{u}$ in the cells' barycenters at the time t^n with an adequate order as well. Again, to the authors knowledge, this is a new result.

The next section is concerned with the solution of the problem that arises form the fact that each integration node is adjacent to two cells and that, due to the cell wise developed Taylor expansion, there exist two approximating values for $\mathbf{u}$ in these points. This configuration is called a Riemann problem, and its analytical solution was discussed in the first chapter. In this section now the numerical solution of Riemann problems is treated in order to obtain an easily computed approximation for the exact solution. Two Riemann solver are presented at this point, namely the HLL and its extension, the HLLC. More exhausting information on the subject can be found for example in [Tor99] and in the references mentioned therein.

The last section of the second chapter finally is dedicated to the treatment of source terms due to topography and to the problems occurring due to wet/dry fronts in the scheme. The modifications being necessary in the reconstruction and in the computation of the space time Taylor expansion for the inclusion of topography are named. Further, another Riemann solver, developed by [ChL99, Seg99, CLS04], is presented that can cope with topography and dry states. It is proven that the resulting scheme, using this Riemann solver, is well-balanced, that means it preserves certain kinds of steady state solutions. The amendments to the scheme necessary to cope with the problems caused by the inclusion of the treatment of dry states in a higher order scheme using polynomial reconstruction are motivated.

In the third chapter, numerical results are presented to confirm the considerations concerning the properties of the scheme developed in chapter two.

Finally, the fourth chapter contains a summary of the results achieved so far and some prospects for the future work, as well as possible applications of the scheme.

Notation and Abbreviations

In general the following notations are used. Vectors are printed in **bold** letters. In most cases, the usual notation $\mathbf{v} = (v_0, ..., v_{n-1})^T$ is used. However, for the exact specification it is sometimes necessary to add multiple indices. To refer to single components of a vector $\mathbf{v}_j^i$, the following notation is used:

$$\mathbf{v}_j^i = \left(v_{j;0}^i, ..., v_{j;n-1}^i\right)^T .$$

The enumeration of vectors in this work starts with the index $i = 0$. The exception are vectors referring to the directions x_1 and x_2 of the coordinate system, such as the vector of velocity $\mathbf{v} = (v_1, v_2)$ or all vectors denoting points in space. Thus, for the vectors $\mathbf{u}$ and $\mathbf{w}$ of the conserved and primitive variables of the 2D shallow water equations, the components with index $i = 1, 2$ contain the directed quantities in direction x_i. The space variable is denoted $\mathbf{x}$, the time variable t.

Sign	Definition	Description
$\mathbf{b}_i = (b_1, b_2)$		barycenter of σ_i
$c = c(\mathbf{x}, t)$	$:= \sqrt{\Phi}$	celerity
deg	$:= ord - 1$	degree of the polynomials used in the scheme
$\mathbb{D}$	$:= \mathbb{R}^+ \times \mathbb{R}^2$	co-domain of the 2D-shallow water equations
$\mathbf{f}_j(\mathbf{u}),\ j = 1, 2$	$:= \begin{pmatrix} u_j \\ \frac{u_1 u_j}{u_0} + \delta_{1j}\left(\frac{1}{2}u_0^2\right) \\ \frac{u_2 u_j}{u_0} + \delta_{2j}\left(\frac{1}{2}u_0^2\right) \end{pmatrix}$	flux functions
g	$:= 9.81\frac{m}{s^2}$	gravity constant

$\mathbf{g}(\mathbf{u})$	$:= \begin{pmatrix} 0 \\ -\mathrm{g}\Phi\partial_{x_1} top \\ -\mathrm{g}\Phi\partial_{x_2} top \end{pmatrix}$	source term
$H = H(\mathbf{x}, t)$		water height
$\lambda_k(\mathbf{u})$, $k = 1, 2, 3(, 4)$		k-th eigenvalue of the x_1-split 2D shallow water equations
$\mathbf{l}^k_{i,j} = (l^k_{i,j;1}, l^k_{i,j;2})^T$		edges between σ_i and σ_j, $k = 1, 2$
$\mathbf{n}^k_{i,j} = (n^k_{i,j;1}, n^k_{i,j;2})^T$	$:= \frac{1}{\lVert l^k_{i,j} \rVert_2}(-l^k_{i,j;2}, l^k_{i,j;1})^T$	outer normal vector to $\mathbf{l}^k_{i,j}$
noc	$:= \binom{deg+2}{2}$	number of coefficients of polynomials $p \in \Pi_{deg}(\mathbb{R}^2; \mathbb{R})$
non_i		number of neighboring cells of σ_i
nos_i		number of stencils $S_{i,j}$ for σ_i
ord		order of the numerical scheme
$\Omega \subset \mathbb{R}^2$		integration domain
$\mathbf{p}^n_{i,j} = \mathbf{p}^n_{i,j}(\mathbf{x})$		vector of basic reconstruction polynomials for σ_i at time t^n, based on stencil $S_{i,j}$
$\Phi = \Phi(\mathbf{x}, t)$	$:= \mathrm{g}H$	Geo potential
$\mathbf{r}_k(\mathbf{u})$, $k = 1, 2, 3(, 4)$		right eigenvector to $\lambda_k(\mathbf{u})$
$\rho^i_k(\mathbf{u})$, $k = 1, 2, 3(, 4)$, $i = 1, 2(, 3)$		i-th Riemann invariant to the k-th eigenvalue

$S_{i,j}$	$:= \{\sigma_k \vert k \in I \subseteq \{ 0, ..., \#\Sigma - 1\}, i \in I\}$	stencil for computation of $\mathbf{p}^n_{i,j}$, contains σ_i
$\sigma_i \subseteq \Omega$		control volume, polygonally bounded
$\vert\sigma_i\vert$	$\int_{\sigma_i} 1 \, d\mathbf{x}$	area of σ_i
$\delta\sigma_i$	$:= \bigcup_{(k)} \delta^k \sigma_i$	boundary of σ_i, polygon
Σ	$:= \{\sigma_i \vert i = 0, ..., \#\Sigma - 1\}$	secondary net resp secondary grid; $\bigcup_i \sigma_i = \Omega$
$\tau_i \subseteq \Omega$		triangle
T	$:= \{\tau_j \vert j = 0, ..., \#T\}$	triangulation; $\bigcup_j \tau_j = \Omega$
$top = top(\mathbf{x})$		topography, bottom elevation
$\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$	$= (\Phi, \Phi v_1, \Phi v_2)^T(\mathbf{x}, t)$	vector of conservative variables
$\overline{\mathbf{u}}^n_i$	$:= \vert\sigma_i\vert^{-1} \int_{\sigma_i} \mathbf{u}(\mathbf{x}, t^n) d\mathbf{x}$	vector of integral cell mean values of $\mathbf{u}$ on σ_i at time t^n
$\mathbf{u}^n_i = \mathbf{u}^n_i(\mathbf{x})$	$:= (u^n_{i;0}, u^n_{i;1}, u^n_{i;2})^T(\mathbf{x})$	vector of weighted reconstruction polynomials for $\mathbf{u}$ on cell σ_i based on the cell mean values at time t^n
$\mathbf{U}^n_i = \mathbf{U}^n_i(\mathbf{x}, t)$	$:= (U^n_{i;0}, U^n_{i;1}, U^n_{i;2})^T(\mathbf{x}, t)$	vector of space time polynomials for $\mathbf{u}$ on cell σ_i at time t^n
$\mathbf{v} = \mathbf{v}(\mathbf{x}, t)$	$= (v_1, v_2)^T(\mathbf{x}, t)$	velocity

$\mathbf{w} = \mathbf{w}(\mathbf{x}, t)$	$= (H, v_1, v_2, top)^T(\mathbf{x}, t)$	vector of primitive variables, including topography
$\omega^n_{i,j;k}$		veight for the polynomial $p^n_{i,j;k}$

Chapter 1

Theoretical Background

In this chapter the theoretical background necessary for developing the numerical scheme in chapter 2 is presented. It is divided into two parts. The first part introduces the 2D shallow water equations, its derivation and its properties. It emphasizes the property of hyperbolicity and introduces a special initial value problem called Riemann problem, whose solution is closely related to hyperbolicity.

The second part contains a brief derivation of an explicit finite volume scheme for the 2D shallow water equations as well as a short motivation for the numerical approach carried out in chapter 2.

1.1 The 2D Shallow Water Equations

The inviscid shallow water equations are a model to describe the temporal evolution of the water height and flow rate, or velocity, of liquids due to gravity under given initial conditions.

This model is considered valid for bodies of water whose surface dimensions are very large compared to their depth, as can be seen from its derivation in section 1.1.1. This is the case for a flat puddle as well as for an ocean. The model as it is derived in this work takes into account influences due to topography, rain and evaporation, but neglects the effects from other environmental conditions like wind or bottom friction. The numerical schemes presented in chapter 2 will only consider the source terms due to topography though.

1.1.1 Derivation of the Equations

The general basis for modeling fluids are the equations for the conservation of mass and momentum modified by additional conditions. On the one hand, these modifications consist of applying conditions for the adaption of the general equations to the special facts describing the properties of the fluid considered. On the other hand, due to the practicability of the model to be developed, some simplifications will be applied. These modifications naturally restrict the validity of the model to cases where it can be assumed that the aforementioned conditions hold.

The basis for deriving the 2D shallow water equations are the equations for conservation of mass and momentum in three space dimensions and time. To adapt those general equations to the case of inviscid liquids, boundary conditions and simplifying physical assumptions are taken into account. The boundary conditions characterize the behavior of the described quantities at the bottom and the surface. The simplifying physical assumptions are firstly that there exists no friction and that, secondly, no production or dissipation of mass occurs. Furthermore, it is assumed that the body forces are restricted to gravity and that the density of the fluid is constant with respect to space and time. Finally, the assumption the model is named after, is that the horizontal extension is so much larger than the vertical length scale that the acceleration of particles in vertical direction can be neglected.

Especially the last aspect restricts the validity of the model to so called shallow flows, that is to bodies of water whose surface is measured on much larger scales than its depth.

A very useful tool for deriving the equations from conservation of mass and momentum is Reynolds transport theorem, which can be found in standard literature to the subject.

Theorem 1.1 (Reynolds' transport theorem)**.** *Let $q(\mathbf{x},t)$ be a quantity per unit volume and $\sigma(t)$ a control volume with surface $\delta\sigma(t)$ that moves along with the flow of velocity $\mathbf{v}(\mathbf{x},t)$. Then the equation*

$$\frac{d}{dt}\int_{\sigma(t)} q(\mathbf{x},t)\ d\mathbf{x} = \int_{\sigma(t)} \left(\frac{\partial}{\partial t}q(\mathbf{x},t) + \nabla\cdot(q\mathbf{v})(\mathbf{x},t)\right)\ d\mathbf{x}$$

holds.

Theorem 1.1 thus gives an expression for the time rate of change of the total amount of $q(\mathbf{x},t)$ contained in the control volume. A proof of this theorem can be found, for example, in [MRT05, ChM98].

1.1.1.1 Conservation of Mass

To obtain the equation for mass conservation, let $\sigma(t)$ be a *control volume* with surface $\delta\sigma(t)$, consisting of a set of particles that moves along with the flow of *velocity* $\mathbf{v} : \mathbb{R}^3 \times \mathbb{R}_0^+ \rightarrow \mathbb{R}^3$. $\mathbf{n}$ is the *outer normal vector* with $|\mathbf{n}| = 1$ that is normal to $\delta\sigma(t)$.

This implies in particular that the mass of $\sigma(t)$ remains constant for all $t \in \mathbb{R}_0^+$, but that the size and shape may change.

Therefore, for the *density* $\rho : \mathbb{R}^3 \times \mathbb{R}_0^+ \rightarrow \mathbb{R}$ it follows that

$$0 = \frac{d}{dt} \int_{\sigma(t)} \rho \; d\mathbf{x},$$

which can be modified to

$$0 = \int_{\sigma(t)} \left(\partial_t \rho + \nabla \cdot (\rho \mathbf{v})\right) \, d\mathbf{x}$$

by using theorem 1.1. As this equation is valid for arbitrary $\sigma(t)$, the equation

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{1.1}$$

holds.

1.1.1.2 Conservation of Momentum

The equation for the conservation of the *momentum* $(\rho\mathbf{v})(\mathbf{x}, t) : \mathbb{R}^3 \times \mathbb{R}_0^+ \rightarrow \mathbb{R}^3$ needed here is obtained from Newtons 2^{nd} Law: *Mutationem motus proportionalem esse vi motrici impressae, et fieri secundum lineam rectam qua vis illa imprimitur.* From this Law follows that for $\sigma(t)$ the temporal change of the momentum equals the sum of all forces that take effect on the mass of the control volume. By applying the assumption of absence of friction, only body forces and *pressure* $p : \mathbb{R}^3 \times \mathbb{R}_0^+ \rightarrow \mathbb{R}$, acting on the boundary of $\sigma(t)$, need to be taken into account. Let $\mathbf{K} : \mathbb{R}^3 \times \mathbb{R}_0^+ \rightarrow \mathbb{R}^3$ represent the *force per unit mass.*

Newtons 2^{nd} Law then reads

$$\frac{d}{dt} \int_{\sigma(t)} \rho \mathbf{v} \; d\mathbf{x} = - \int_{\delta\sigma(t)} p\mathbf{n} \; ds + \int_{\sigma(t)} \rho \mathbf{K} \; d\mathbf{x} \tag{1.2}$$

and transforms into

$$\frac{d}{dt} \int_{\sigma(t)} \rho \mathbf{v} \; d\mathbf{x} = \int_{\sigma(t)} (\rho \mathbf{K} - \nabla p) \; d\mathbf{x}$$

by applying Gauss' integral theorem.

The application of Reynolds' transport theorem to the left hand side of equation (1.2) leads to

$$\begin{aligned}\frac{d}{dt}\int_{\sigma(t)} \rho\mathbf{v}\; d\mathbf{x} &= \frac{d}{dt}\int_{\sigma(t)} \begin{pmatrix} \rho v_1 \\ \rho v_2 \\ \rho v_3 \end{pmatrix} d\mathbf{x} \\ &= \int_{\sigma(t)} \left[\partial_t \begin{pmatrix} \rho v_1 \\ \rho v_2 \\ \rho v_3 \end{pmatrix} + \begin{pmatrix} \nabla \cdot (\rho v_1 \mathbf{v}) \\ \nabla \cdot (\rho v_2 \mathbf{v}) \\ \nabla \cdot (\rho v_3 \mathbf{v}) \end{pmatrix} \right] d\mathbf{x} \\ &= \int_{\sigma(t)} \left(\partial_t(\rho\mathbf{v}) + \sum_{j=1}^{3} \partial_{x_j}(\rho v_j \mathbf{v}) \right) d\mathbf{x}.\end{aligned}$$

Equation (1.2) then reads

$$\int_{\sigma(t)} \left[\partial_t(\rho\mathbf{v}) + \sum_{j=1}^{3} \partial_{x_j}(\rho v_j \mathbf{v}) - \rho\mathbf{K} + \nabla p \right] d\mathbf{x} = \mathbf{0}.$$

Again, as this result is valid for all $\sigma(t)$, the expressions

$$\partial_t(\rho v_i) + \sum_{j=1}^{3} \partial_{x_j}(\rho v_j v_i + \delta_{ij} p) = \rho K_i, \quad i = 1, 2, 3 \tag{1.3}$$

hold.

1.1.1.3 Further Assumptions and Boundary Conditions

The equations derived so far are very general. By introducing boundary conditions they are adapted to the behavior of gravity induced free surface flows. Further assumptions lead to a more specialized model that is easier to handle, at the cost of a restricted validity.

The first assumption is that the density is constant with $\rho(\mathbf{x}, t) = \rho \neq 0$, which yields

$$\partial_t \rho = \partial_{x_1}\rho = \partial_{x_2}\rho = \partial_{x_3}\rho = 0. \tag{1.4}$$

Plugging equation (1.4) into (1.1) gives

$$0 = \underbrace{\partial_t \rho}_{=0} + \underbrace{\mathbf{v} \cdot \nabla\rho}_{=0} + \rho \nabla \cdot \mathbf{v} \Rightarrow \nabla \cdot \mathbf{v} = 0. \tag{1.5}$$

This expresses the incompressibility of the considered fluid, since by theorem 1.1,

$$\frac{d\,|\sigma(t)|}{dt} = \frac{d}{dt}\int_{\sigma(t)} 1 \; d\mathbf{x} = \int_{\sigma(t)} \frac{\partial}{\partial t}1 + \nabla\cdot\mathbf{v} \; d\mathbf{x} = \int_{\sigma(t)} \nabla\cdot\mathbf{v} \; d\mathbf{x} = 0.$$

Secondly, the body forces are given by *gravity* $\mathrm{g} = 9.81\frac{m}{s^2}$ only, such that

$$\mathbf{K} = \begin{pmatrix} K_1 \\ K_2 \\ K_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\mathrm{g} \end{pmatrix}. \tag{1.6}$$

The next assumption is that the horizontal extension of the body of water is substantially larger than the vertical one. As a consequence of this shallow water assumption, the acceleration of particles in vertical direction is assumed to be zero. This yields

$$0 = \frac{Dv_3}{Dt} = \partial_t v_3 + \mathbf{v}\cdot\nabla v_3 = \partial_t v_3 + \sum_{j=1}^{3} v_j \partial_{x_j} v_3. \tag{1.7}$$

Using equation (1.3) with $i = 3$, it follows from the application of equations (1.4) through (1.7) that

$$\begin{aligned} -\rho\mathrm{g} &= \rho K_3 \\ &= \partial_t\left(\rho v_3\right) + \sum_{j=1}^{3} \partial_{x_j}\left(\rho v_j v_3\right) + \partial_{x_3} p \\ &= \rho\left(\partial_t v_3 + \sum_{j=1}^{3} \partial_{x_j}\left(v_j v_3\right)\right) + \partial_{x_3} p \\ &= \rho\left(\underbrace{\partial_t v_3 + \mathbf{v}\cdot\nabla v_3}_{=0} + v_3 \underbrace{\nabla\cdot\mathbf{v}}_{=0}\right) + \partial_{x_3} p \\ &= \partial_{x_3} p \end{aligned} \tag{1.8}$$

holds.

The next step is to look at the special boundary conditions that follow from modeling flowing water. There are two boundaries that have to be taken into account, namely the bottom and the surface. The bottom's elevation is represented by the *topography*

$$top : \mathbb{R}^2 \to \mathbb{R},\; (x_1, x_2)^T \mapsto x_3 = top(x_1, x_2),$$

which is assumed to be invariant with respect to time. The *free surface*

$$s : \mathbb{R}^2 \times \mathbb{R}_0^+ \to \mathbb{R},\ (x_1, x_2, t)^T \mapsto x_3 = s(x_1, x_2, t),$$

however, may depend on time and space. For simplicity of notation

$$\mathbf{x}_{top} := (x_1, x_2, top(x_1, x_2))^T \in \mathbb{R}^3$$

and

$$\mathbf{x}_s(t) := (x_1, x_2, s(x_1, x_2, t))^T \in \mathbb{R}^3 \tag{1.9}$$

denote points on the corresponding boundaries.

The bottom is assumed to be impermeable to water, which gives

$$\mathbf{v} \cdot \mathbf{n} = 0 \quad \text{for} \quad (\mathbf{x}_{top}, t)$$

with *normal vector* $\mathbf{n}(\mathbf{x}_{top}, t) = (\partial_{x_1} top, \partial_{x_2} top, -1)^T \in \mathbb{R}^3$, which implies that

$$v_3 = v_1 \partial_{x_1} top + v_2 \partial_{x_2} top \quad \text{for} \quad (\mathbf{x}_{top}, t) \tag{1.10}$$

holds.

The surface moves due to the field of velocity and may rise or fall by rain or evaporation described as *production term* $q = q(\mathbf{x}_s(t), t)$ which yields

$$(q + v_3)(\mathbf{x}_s(t), t) = \frac{d}{dt} s(x_1, x_2, t),$$

so that

$$v_3 = \partial_t s + v_1 \partial_{x_1} s + v_2 \partial_{x_2} s - q \text{ for } (\mathbf{x}_s(t), t). \tag{1.11}$$

Let p_{atm} be the *atmospheric pressure* at the surface. As the pressure grows linearly to the water depth, equation (1.8) can then be written in the following form:

$$p(\mathbf{x}, t) = -\rho \mathrm{g}(x_3 - s(x_1, x_2, t)) + p_{atm},$$

which gives

$$\partial_{x_i} p = \rho \mathrm{g} \partial_{x_i} s \quad i = 1, 2. \tag{1.12}$$

Inserting the equations (1.12) into the corresponding momentum equations (1.3) and dividing by the constant $\rho \neq 0$ leads to

$$\partial_t v_i + \sum_{j=1}^{3} v_j \partial_{x_j} v_i = -\mathrm{g} \partial_{x_i} s \quad i = 1, 2. \tag{1.13}$$

Integrating equation (1.5) in vertical direction and inserting the equations (1.10) and (1.11) yields

$$\begin{aligned}
0 &= \int_{top(x_1,x_2)}^{s(x_1,x_2,t)} \sum_{j=1}^{3} \partial_{x_j} v_j \; dx_3 \\
&= \sum_{j=1}^{2} \int_{top(x_1,x_2)}^{s(x_1,x_2,t)} \partial_{x_j} v_j \; dx_3 + v_3(\mathbf{x}_s(t),t) - v_3(\mathbf{x}_{top},t) \\
&= \sum_{j=1}^{2} \int_{top(x_1,x_2)}^{s(x_1,x_2,t)} \partial_{x_j} v_j \; dx_3 \\
&\quad + \sum_{j=1}^{2} \big[(v_j(\mathbf{x}_s(t),t)\partial_{x_j} s(x_1,x_2,t)) - (v_j(\mathbf{x}_{top},t)\partial_{x_j} top(x_1,x_2))\big] \\
&\quad + \partial_t s(\mathbf{x}_s(t),t) - q(\mathbf{x}_s(t),t) \\
&= \partial_t s(x_1,x_2,t) + \sum_{j=1}^{2} \partial_{x_j} \int_{top(x_1,x_2)}^{s(x_1,x_2,t)} v_j \; dx_3 - q(\mathbf{x}_s(t),t). \qquad (1.14)
\end{aligned}$$

Let v_1, v_2 be specified at time $t = 0$ via the initial conditions

$$v_i(\mathbf{x},0) = v_i(x_1,x_2,x_3,0) \text{ for } (x_1,x_2,x_3)^T \in \mathbb{R}^3,$$

such that v_i is constant with respect to x_3. This transforms equations (1.13) into

$$-g\partial_{x_i} s = \partial_t v_i + \sum_{j=1}^{2} v_j \partial_{x_j} v_i \quad i = 1,2. \qquad (1.15)$$

Thus, both sides of the equations (1.15) are independent from x_3 and v_3, which implies

$$v_i(x_1,x_2,x_3,t) = v_i(x_1,x_2,t) \quad \forall t \in \mathbb{R}_0^+ \quad i = 1,2.$$

This implies for equation (1.14) in combination with $\partial_t top = 0$ and (1.9), which provides $q(\mathbf{x}_s(t),t) = q(x_1,x_2,t)$,

$$q = \partial_t (s - top) + \sum_{j=1}^{2} \partial_{x_j}((s - top)v_j) \text{ for } (x_1,x_2,t). \qquad (1.16)$$

Defining the *water height* as $H = \max\{s - top, 0\} : \mathbb{R}^2 \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ and multiplying equation (1.16) with g,

$$\partial_t \Phi + \sum_{j=1}^{2} \partial_{x_j}(\Phi v_j) = \mathrm{g} q \text{ for } (\mathbf{x},t) = (x_1,x_2,t), \qquad (1.17)$$

is obtained where $\Phi = \mathrm{g}H$ is called the *Geo potential.*

Finally, by using equations (1.15) and (1.17) for $i = 1, 2$,

$$\begin{aligned} g(qv_i - \Phi\partial_{x_i} top) &= v_i \left(\partial_t \Phi + \sum_{j=1}^{2} \partial_{x_j} (\Phi v_j) \right) \\ &\quad + \Phi \left(\partial_t v_i + \sum_{j=1}^{2} v_j \partial_{x_j} v_i + g\partial_{x_i} s \right) - \mathrm{g}\Phi\partial_{x_i} top \\ &= \partial_t (\Phi v_i) + \sum_{j=1}^{2} \partial_{x_j} (\Phi v_i v_j) + \Phi\partial_{x_i} (\mathrm{g}(s - top)) \\ &= \partial_t (\Phi v_i) + \sum_{j=1}^{2} \partial_{x_j} \left[(\Phi v_i v_j) + \delta_{ij} \left(\frac{1}{2} \Phi^2 \right) \right] \end{aligned}$$

can be obtained. These equations describe the momentum balance and complete the 2D shallow water equations.

Thus, the complete 2D shallow water equations read

$$\partial_t \mathbf{u} + \sum_{j=1}^{2} \partial_{x_j} \mathbf{f}_j(\mathbf{u}) = \mathbf{g}(\mathbf{u}) \tag{1.18}$$

with the vector of conservative variables

$$\mathbf{u} = \begin{pmatrix} \Phi \\ \Phi v_1 \\ \Phi v_2 \end{pmatrix} = \begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix} \in \mathbb{D} := \mathbb{R}^+ \times \mathbb{R}^2,$$

the two flux functions

$$\mathbf{f}_j(\mathbf{u}) = \begin{pmatrix} \Phi v_j \\ \Phi v_1 v_j + \delta_{1j} \left(\frac{1}{2} \Phi^2 \right) \\ \Phi v_2 v_j + \delta_{2j} \left(\frac{1}{2} \Phi^2 \right) \end{pmatrix} = \begin{pmatrix} u_j \\ \frac{u_1 u_j}{u_0} + \delta_{1j} \left(\frac{1}{2} u_0^2 \right) \\ \frac{u_2 u_j}{u_0} + \delta_{2j} \left(\frac{1}{2} u_0^2 \right) \end{pmatrix} \in \mathbb{R}^3, \quad j = 1, 2,$$

and the vector of source terms

$$\mathbf{g}(\mathbf{u}) = \begin{pmatrix} \mathrm{g}q \\ \mathrm{g}\left(qv_1 - \Phi\partial_{x_1} top\right) \\ \mathrm{g}\left(qv_2 - \Phi\partial_{x_2} top\right) \end{pmatrix} \in \mathbb{R}^3.$$

Rain and evaporation will not be considered in this work, so it is assumed that $q = 0$ which yields

$$\mathbf{g}(\mathbf{u}) = \begin{pmatrix} 0 \\ -\mathrm{g}\Phi\partial_{x_1} top \\ -\mathrm{g}\Phi\partial_{x_2} top \end{pmatrix}.$$

Integrating equation (1.18) over an arbitrary control volume σ with boundary $\delta\sigma$ and outer normal vector $\mathbf{n}$ with $|\mathbf{n}| = 1$, and applying the transport theorem and Gauss' integral theorem yields the *integral form*

$$\frac{d}{dt}\int_\sigma \mathbf{u} d\mathbf{x} = -\int_{\delta\sigma}\sum_{j=1}^{2} \mathbf{f}_j(\mathbf{u}) n_j ds + \int_\sigma \mathbf{g}(\mathbf{u}) d\mathbf{x}. \tag{1.19}$$

This form has the advantage of allowing discontinuous solutions for $\mathbf{u}$. That is important especially for hyperbolic partial differential equations, like the 2D shallow water equations, as their solutions may develop discontinuities in finite time, [Lax73], even for smooth initial conditions. The integral form of the 2D shallow water equations forms the base for the finite volume approach used in this work.

Sometimes it is more convenient to consider the 2D shallow water equations formulated in *primitive variables*. To obtain this formulation, firstly the product rule is applied to all derivatives from (1.18). Secondly, the whole system is divided by g. Afterwards the expression for $\partial_t H$ obtained from the equation for mass conservation is plugged into the equations for the conservation of momentum. Finally, these expressions are divided by H. The equations resulting from all this operations then read

$$\begin{aligned} \partial_t H + v_1\partial_{x_1} H + H\partial_{x_1} v_1 + v_2\partial_{x_2} H + H\partial_{x_2} v_2 &= 0 \\ \partial_t v_1 + v_1\partial_{x_1} v_1 + \mathrm{g}\partial_{x_1} H + v_2\partial_{x_2} v_1 &= -\mathrm{g}\partial_{x_1} top \\ \partial_t v_2 + v_1\partial_{x_1} v_2 + v_2\partial_{x_2} v_2 + \mathrm{g}\partial_{x_2} H &= -\mathrm{g}\partial_{x_2} top. \end{aligned} \tag{1.20}$$

The formulation (1.20), that is valid only for $H > 0$, is used for deriving the properties of the equations in the following section 1.2.

1.2 Properties of the Equations

The 2D shallow water equations have some properties that will be exploited in developing the numerical scheme.

The most important property is hyperbolicity. Hyperbolicity in the context of systems of partial differential equations is a property that is defined through the eigenvalues of the Jacobian matrices of the flux functions $\mathbf{f}_j,\ j = 1, 2$. For hyperbolic systems of partial differential equations, the solution for special initial conditions, the so-called Riemann problems, splits into separate waves and can thus be determined analytically. The mathematical theory related to the solution of Riemann problems is discussed in sections 1.2.3, 1.2.4, 1.2.5. Solving local Riemann problems will be the basis of the numerical scheme.

Nevertheless, the first property to be discussed here is the rotational invariance of the 2D shallow water equations as all the following considerations can be simplified by its application.

1.2.1 Rotational Invariance

This property will be used later on to derive the numerical scheme. Its consequence is that it is possible to rotate the coordinate system from the (x_1, x_2) into the (x_n, x_t) direction, that is normal and tangential, in relation to a given line segment. Therefore, the treatment of the full 2D-shallow water equations can be restricted to the x_1-split case in the computation of the fluxes over the cell boundaries later in the finite volume scheme. In this case, only the first flux function and the derivatives in x_n-direction, or x_1-direction, respectively, need to be taken into account.

Theorem 1.2 (Rotational Invariance)**.** *For the flux functions $\mathbf{f}_j,\ j = 1, 2$, as in equation* (1.19)*, the equation*

$$\sum_{j=1}^{2} \mathbf{f}_j(\mathbf{u}) n_j = \mathbf{T}^{-1}(\mathbf{n}) \mathbf{f}_1(\mathbf{T}(\mathbf{n})\mathbf{u})$$

with

$$\mathbf{T}(\mathbf{n}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & n_1 & n_2 \\ 0 & -n_2 & n_1 \end{pmatrix}$$

holds.

Proof. Let $v_n := v_1 n_1 + v_2 n_2$ be the velocity in normal and $v_t := -v_1 n_2 + v_2 n_1$

the velocity in tangential direction. Moreover, let $|\mathbf{n}| = 1$. Then

$$\sum_{j=1}^{2} \mathbf{f}_j(\mathbf{u}) n_j = \sum_{j=1}^{2} \begin{pmatrix} \Phi v_j \\ \Phi v_1 v_j + \delta_{1j}\Phi^2 \\ \Phi v_2 v_j + \delta_{2j}\Phi^2 \end{pmatrix} n_j = \begin{pmatrix} \Phi v_n \\ \Phi v_1 v_n + \frac{1}{2}\Phi^2 n_1 \\ \Phi v_2 v_n + \frac{1}{2}\Phi^2 n_2 \end{pmatrix}$$

follows directly. Since $\mathbf{T}(\mathbf{n})\mathbf{u} = (\Phi, \Phi v_n, \Phi v_t)^T$, the equation

$$\mathbf{f}_1(\mathbf{T}(\mathbf{n})\mathbf{u}) = \begin{pmatrix} \Phi v_n \\ \Phi v_n^2 + \Phi^2 \\ \Phi v_n v_t \end{pmatrix},$$

holds, which yields

$$\begin{aligned} \mathbf{T}^{-1}(\mathbf{n})\mathbf{f}_1(\mathbf{T}(\mathbf{n})\mathbf{u}) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & n_1 & -n_2 \\ 0 & n_2 & n_1 \end{pmatrix} \mathbf{f}_1(\mathbf{T}(\mathbf{n})\mathbf{u}) \\ &= \begin{pmatrix} \Phi v_n \\ \Phi v_n^2 n_1 + \frac{1}{2}\Phi^2 n_1 - \Phi v_n v_t n_2 \\ \Phi v_n^2 n_2 + \frac{1}{2}\Phi^2 n_2 + \Phi v_n v_t n_1 \end{pmatrix} \\ &= \begin{pmatrix} \Phi v_n \\ \Phi v_n (v_n n_1 - v_t n_2) + \frac{1}{2}\Phi^2 n_1 \\ \Phi v_n (v_n n_2 + v_t n_1) + \frac{1}{2}\Phi^2 n_2 \end{pmatrix}. \end{aligned}$$

With

$$v_n n_1 - v_t n_2 = (v_1 n_1 + v_2 n_2) n_1 - (v_2 n_1 - v_1 n_2) n_2 = v_1 \underbrace{(n_1^2 + n_2^2)}_{=1} = v_1$$

and

$$v_n n_2 + v_t n_1 = (v_1 n_1 + v_2 n_2) n_2 + (v_2 n_1 - v_1 n_2) n_1 = v_2 \underbrace{(n_2^2 + n_1^2)}_{=1} = v_2,$$

the proof is complete. □

1.2.2 Hyperbolicity

Hyperbolicity is the property that represents the central point concerning the aim of solving the 2D shallow water equations numerically. On the one hand, it is known that hyperbolic systems of conservation laws can develop discontinuities in the solution after finite time even for smooth initial conditions, and that thus solutions can only be understood in the weak sense, see [Lax73]. On the other hand, hyperbolicity allows to determine the solution of a special discontinuous initial value problem, the Riemann problem (1.27), that is introduced on page 38, analytically. In the course of this chapter this solution is developed, mostly following [Lax57, GoR96].

To define the property of hyperbolicity, it is necessary to take a look at the quasi linear form of (1.18) first. In this form, the partial derivatives of the flux functions are expanded via the chain rule into

$$\frac{\partial \mathbf{f}_i(\mathbf{u}(\mathbf{x},t))}{\partial x_i} = \mathbf{f}_i'(\mathbf{u})\frac{\partial \mathbf{u}(\mathbf{x},t)}{\partial x_i}.$$

For the shallow water equations, this leads to

$$\partial_t \mathbf{u} + \sum_{j=1}^{2} \mathbf{f}_j'(\mathbf{u})\partial_{x_j}\mathbf{u} = \mathbf{g}(\mathbf{u}) \tag{1.21}$$

with the Jacobian matrices

$$\mathbf{f}_1'(\mathbf{u}) = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{u_1^2}{u_0^2} + u_0 & 2\frac{u_1}{u_0} & 0 \\ -\frac{u_1 u_2}{u_0^2} & \frac{u_2}{u_0} & \frac{u_1}{u_0} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ c^2 - v_1^2 & 2v_1 & 0 \\ -v_1 v_2 & v_2 & v_1 \end{pmatrix},$$

$$\mathbf{f}_2'(\mathbf{u}) = \begin{pmatrix} 0 & 0 & 1 \\ -\frac{u_1 u_2}{u_0^2} & \frac{u_2}{u_0} & \frac{u_1}{u_0} \\ -\frac{u_2^2}{u_0^2} + u_0 & 0 & 2\frac{u_2}{u_0} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ -v_1 v_2 & v_2 & v_1 \\ c^2 - v_2^2 & 0 & 2v_2 \end{pmatrix}.$$

Here and later on, $c := \sqrt{\Phi}$ denotes the *celerity*. The celerity is the gravity induced velocity of information propagation in water. An easily observable example is the spreading of waves after having thrown a stone into a lake. As the water height is always greater or equal to zero, $c \in \mathbb{R}_0^+$.

The property of hyperbolicity for systems of conservation and balance laws respectively is defined as follows:

Definition 1.3 (Hyperbolic System)**.** A system (1.18) of p conservation laws with Jacobian matrices $\mathbf{f}_j'(\mathbf{u}) \in \mathbb{R}^{p\times p}, j = 1, ..., d$, is called *hyperbolic* if the matrix

$$\mathbf{F}(\mathbf{u},\boldsymbol{\omega}) := \sum_{j=1}^{d} \omega_j \mathbf{f}_j'(\mathbf{u})$$

has p real eigenvalues

$$\lambda_1(\mathbf{u},\boldsymbol{\omega}) \leq ... \leq \lambda_p(\mathbf{u},\boldsymbol{\omega})$$

and a corresponding set of linearly independent (right) eigenvectors $\mathbf{r}_k(\mathbf{u},\boldsymbol{\omega})$, $k = 1, ..., p$, for any vector $\mathbf{u} \in \mathbb{D} \subset \mathbb{R}^p$ of conservative variables and any direction $\boldsymbol{\omega} \in \mathbb{R}^d$, $|\boldsymbol{\omega}| = 1$. The system is called *strictly hyperbolic* if the eigenvalues are all distinct.

Due to hyperbolicity $\mathbf{F}(\mathbf{u},\boldsymbol{\omega})$ can be diagonalized. This trait can, unfortunately, not be used to uncouple systems of nonlinear hyperbolic equations. This is because the eigenvectors depend on space and time as can be seen in section 1.2.3. Thus, the multiplication with $\mathbf{R} = (\mathbf{r}_k(\mathbf{x}, t))_{k=1,...,p}$ can not be interchanged with the partial derivations of $\mathbf{u}$ in space and time.

Theorem 1.4. *The 2D shallow water equations are strictly hyperbolic for* $\Phi > 0$.

Proof. The matrix $\mathbf{F}(\mathbf{u},\boldsymbol{\omega})$ is given by

$$\mathbf{F}(\mathbf{u},\boldsymbol{\omega}) = \begin{pmatrix} 0 & \omega_1 & \omega_2 \\ (c^2 - v_1^2)\omega_1 - v_1 v_2 \omega_2 & 2v_1\omega_1 + v_2\omega_2 & v_1\omega_2 \\ -v_1 v_2 \omega_1 + (c^2 - v_2^2)\omega_2 & v_2\omega_1 & v_1\omega_1 + 2v_2\omega_2 \end{pmatrix}.$$

Its characteristic polynomial is

$$p(\lambda) = (\lambda - (v_1\omega_1 + v_2\omega_2 - c\,|\boldsymbol{\omega}|))(\lambda - (v_1\omega_1 + v_2\omega_2))(\lambda - (v_1\omega_1 + v_2\omega_2 + c\,|\boldsymbol{\omega}|)).$$

Hence, the eigenvalues of $\mathbf{F}(\mathbf{u},\boldsymbol{\omega})$ are

$$\begin{aligned} \lambda_1(\mathbf{u},\boldsymbol{\omega}) &= v_1\omega_1 + v_2\omega_2 - c\,|\boldsymbol{\omega}| && \in \mathbb{R} \\ \lambda_2(\mathbf{u},\boldsymbol{\omega}) &= v_1\omega_1 + v_2\omega_2 && \in \mathbb{R} \\ \lambda_3(\mathbf{u},\boldsymbol{\omega}) &= v_1\omega_1 + v_2\omega_2 + c\,|\boldsymbol{\omega}| && \in \mathbb{R}. \end{aligned}$$

The eigenvalues are distinct for $c = \sqrt{\Phi} \neq 0$ and, thus, the 2D shallow water equations are strictly hyperbolic for $\Phi = \mathrm{g}H > 0$. □

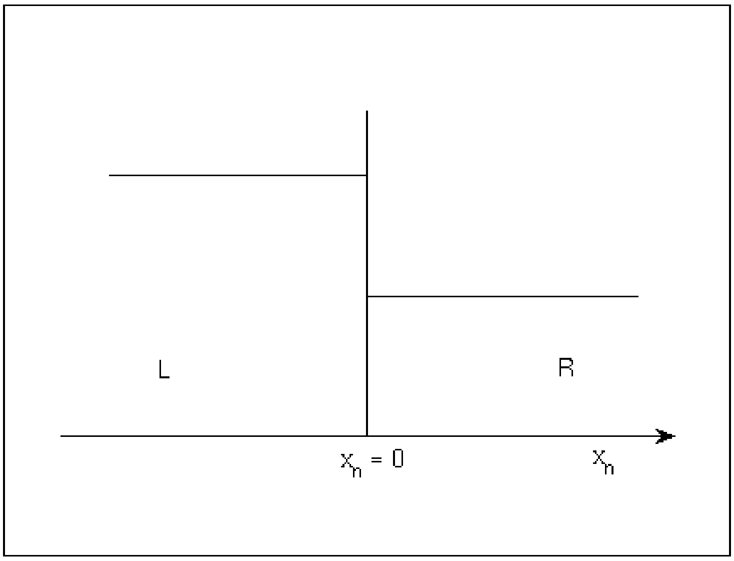

Figure 1.1: The Riemann problem.

1.2.3 Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors are important for analyzing and understanding the nature of hyperbolic systems of conservation laws.

The aim of the following analysis is to develop a solution for a so-called Riemann problem. Here, constant, but different, initial data are given on each side of a set $\{x_n = 0\}$, as depicted in figure 1.1, and their progress in time with respect to the differential equation is studied. Solving Riemann problems is one of the central aspects of the numerical scheme presented in chapter 2. Thus, the theory of the solution of these problems is treated thoroughly. [GoR96] states that it is at this sufficient to analyze the properties of the equation in the direction of the vector $\mathbf{n} = (n_1, n_2)^T$, where $\mathbf{x} \cdot \mathbf{n} = x_n$, due to the rotational invariance, see figure 1.2.

The ansatz for this projection consists of a function $\mathbf{h} : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{D}$ with

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{h}(\mathbf{x} \cdot \mathbf{n}, t) = \mathbf{h}(x_n, t). \tag{1.22}$$

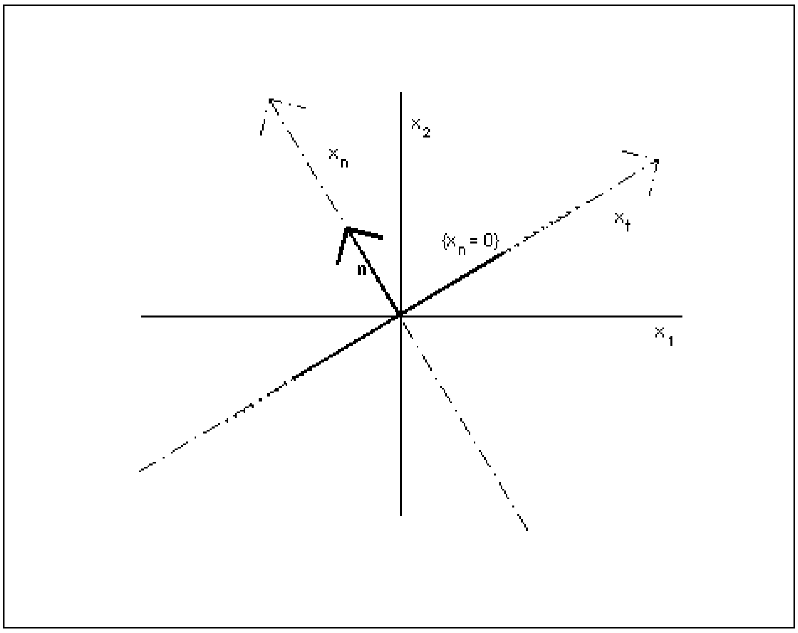

Figure 1.2: Coordinate system, rotated with respect to the normal vector **n**.

Differentiating this equation with respect to x_1 and x_2 leads to

$$\begin{aligned}\frac{\partial}{\partial x_1}\mathbf{u} &= \frac{\partial}{\partial x_n}\mathbf{h}\frac{\partial}{\partial x_1}x_n = \frac{\partial}{\partial x_n}\mathbf{h}n_1\\ \frac{\partial}{\partial x_2}\mathbf{u} &= \frac{\partial}{\partial x_n}\mathbf{h}\frac{\partial}{\partial x_2}x_n = \frac{\partial}{\partial x_n}\mathbf{h}n_2,\end{aligned}$$

and thus

$$\begin{aligned}\frac{\partial}{\partial x_1}\mathbf{f}_1(\mathbf{u}) + \frac{\partial}{\partial x_2}\mathbf{f}_2(\mathbf{u}) &= \mathbf{f}_1'(\mathbf{u})\frac{\partial}{\partial x_1}\mathbf{u} + \mathbf{f}_2'(\mathbf{u})\frac{\partial}{\partial x_2}\mathbf{u}\\ &= \mathbf{f}_1'(\mathbf{h})\frac{\partial}{\partial x_n}\mathbf{h}n_1 + \mathbf{f}_2'(\mathbf{h})\frac{\partial}{\partial x_n}\mathbf{h}n_2\\ &= \frac{\partial}{\partial x_n}\mathbf{f}_1(\mathbf{h})n_1 + \frac{\partial}{\partial x_n}\mathbf{f}_2(\mathbf{h})n_2\\ &= \frac{\partial}{\partial x_n}\sum_{j=1}^{2}\mathbf{f}_j(\mathbf{h})n_j.\end{aligned}$$

The 'projected equations' then read, with equation (1.22),

$$\frac{\partial}{\partial t}\mathbf{u} + \frac{\partial}{\partial x_n}\sum_{j=1}^{2}\mathbf{f}_j(\mathbf{u})n_j = \mathbf{0}. \tag{1.23}$$

The multiplication by the matrix $\mathbf{T}(\mathbf{n})$ as defined in theorem 1.2 rotates the vector $\mathbf{u}$ in a way that the direction of the momentum changes from the (x_1, x_2)-direction into the (normal, tangential)-direction in relation to $\{x_n = 0\}$. Multiplying the equation by $\mathbf{T}(\mathbf{n})$ and using the fact that the shallow water equations are rotationally invariant transforms the equations (1.23) into

$$\begin{aligned}
&\mathbf{T}(\mathbf{n})\frac{\partial \mathbf{u}}{\partial t} + \mathbf{T}(\mathbf{n})\frac{\partial}{\partial x_n}\left(\sum_{j=1}^{2} \mathbf{f}_j(\mathbf{u})n_j\right)\\
=&\mathbf{T}(\mathbf{n})\frac{\partial \mathbf{u}}{\partial t} + \mathbf{T}(\mathbf{n})\frac{\partial}{\partial x_n}\left(\mathbf{T}^{-1}(\mathbf{n})\mathbf{f}_1(\mathbf{T}(\mathbf{n})\mathbf{u})\right)\\
=&\frac{\partial(\mathbf{T}(\mathbf{n})\mathbf{u})}{\partial t} + \frac{\partial}{\partial x_n}\mathbf{f}_1(\mathbf{T}(\mathbf{n})\mathbf{u}),
\end{aligned}$$

as $\mathbf{T}(\mathbf{n})$ is a constant matrix. In the following, to keep the notation simple, $\mathbf{T}(\mathbf{n})\mathbf{u}$ and x_n will be set again to $\mathbf{u}$ and x_1. The equations obtained by the previous considerations are the *x_1-split 2D shallow water equations*

$$\partial_t \mathbf{u} + \partial_{x_1}\mathbf{f}_1(\mathbf{u}) = \mathbf{0} \tag{1.24}$$

or, in quasilinear form,

$$\partial_t \mathbf{u} + \mathbf{f}_1'(\mathbf{u})\partial_{x_1}\mathbf{u} = \mathbf{0}$$

with

$$\mathbf{f}_1'(\mathbf{u}) = \begin{pmatrix} 0 & 1 & 0 \\ u_0 - \frac{u_1^2}{u_0^2} & 2\frac{u_1}{u_0} & 0 \\ -\frac{u_1u_2}{u_0^2} & \frac{u_2}{u_0} & \frac{u_1}{u_0} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ c^2 - v_1^2 & 2v_1 & 0 \\ -v_1v_2 & v_2 & v_1 \end{pmatrix}.$$

The eigenvalues $\lambda_k(\mathbf{u})$ of $\mathbf{f}_1'$ and their associated right eigenvectors $\mathbf{r}_k(\mathbf{u})$, $k = 1, 2, 3$, are

$$\begin{aligned}
\lambda_1(\mathbf{u}) &= \frac{u_1}{u_0} - \sqrt{u_0} && = v_1 - c,\\
\mathbf{r}_1(\mathbf{u}) &= \alpha_1 \begin{pmatrix} 1 \\ \frac{u_1}{u_0} - \sqrt{u_0} \\ \frac{u_2}{u_0} \end{pmatrix} && = \alpha_1 \begin{pmatrix} 1 \\ v_1 - c \\ v_2 \end{pmatrix},\\
\lambda_2(\mathbf{u}) &= \frac{u_1}{u_0} && = v_1,
\end{aligned} \tag{1.25}$$

$$\mathbf{r}_2(\mathbf{u}) \qquad = \alpha_2 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

$$\lambda_3(\mathbf{u}) = \frac{u_1}{u_0} + \sqrt{u_0} \qquad = v_1 + c,$$

$$\mathbf{r}_3(\mathbf{u}) = \alpha_3 \begin{pmatrix} 1 \\ \frac{u_1}{u_0} + \sqrt{u_0} \\ \frac{u_2}{u_0} \end{pmatrix} \qquad = \alpha_3 \begin{pmatrix} 1 \\ v_1 + c \\ v_2 \end{pmatrix}.$$

The pair of eigenvalue, or characteristic speed $\lambda(\mathbf{u})$, and its corresponding right eigenvector $\mathbf{r}(\mathbf{u})$ is called a characteristic field. The k-th eigenpair is called the k-field. The nature of this field is determined by the scalar product of the gradient of λ_k with $\mathbf{r}_k$:

Definition 1.5 (Characteristic field)**.** An k-field is called *linearly degenerate* if

$$\nabla\lambda_k(\mathbf{u}) \cdot \mathbf{r}_k(\mathbf{u}) = 0 \;\forall \mathbf{u} \in \mathbb{D}.$$

An k-field is called *genuinely non-linear* if

$$\nabla\lambda_k(\mathbf{u}) \cdot \mathbf{r}_k(\mathbf{u}) \neq 0 \;\forall \mathbf{u} \in \mathbb{D}.$$

Supposing $\mathbf{u}$ were an integral curve of $\mathbf{r}_k$, that is

$$\mathbf{u}' = \mathbf{r}_k(\mathbf{u}).$$

Then, from definition 1.5 it follows that if the k-field is linearly degenerate, for all points $\mathbf{u}_L, \mathbf{u}_R \in \mathbf{u}$, $\lambda_k(\mathbf{u}_L) = \lambda_k(\mathbf{u}_R)$ holds, as the level curves of $\lambda_k(\mathbf{u})$ are parallel to the streamlines of the vector field $\mathbf{r}_k$. On the other hand, if the k-field is genuinely non-linear, the streamlines of the vector field $\mathbf{r}_k$ always intersect the level curves of $\lambda_k(\mathbf{u})$, and thus for $\mathbf{u}_L, \mathbf{u}_R \in \mathbf{u}$ being distinct points, it always holds that $\lambda_k(\mathbf{u}_L) \neq \lambda_k(\mathbf{u}_R)$.

Theorem 1.6. *For the x_1-split 2D shallow water equations, the 2-field is linearly degenerate while the 1-field and the 3-field are genuinely non-linear.*

Proof. The 2-field is associated with the eigenvalue $\lambda_2(\mathbf{u}) = v_1 = \frac{u_1}{u_0}$. Thus, $\nabla\lambda_2(\mathbf{u}) = (-\frac{u_1}{u_0^2}, \frac{1}{u_0}, 0)^T$ and it follows directly that

$$\nabla\lambda_2(\mathbf{u}) \cdot \mathbf{r}_2(\mathbf{u}) = -\frac{u_1}{u_0^2} \cdot 0 + \frac{1}{u_0} \cdot 0 + 0 \cdot \frac{u_1}{u_0} = 0 \qquad \forall \mathbf{u} \in \mathbb{D}.$$

The 1-field is associated with the eigenvalue $\lambda_1(\mathbf{u}) = v_1 - c = \frac{u_1}{u_0} - \sqrt{u_0}$. Its gradient is $\nabla\lambda_1(\mathbf{u}) = (-\frac{u_1}{u_0^2} - \frac{1}{2\sqrt{u_0}}, \frac{1}{u_0}, 0)^T$ and it follows that

$$\begin{aligned}\nabla\lambda_1(\mathbf{u}) \cdot \mathbf{r}_1(\mathbf{u}) &= (-\frac{u_1}{u_0^2} - \frac{1}{2\sqrt{u_0}}) \cdot 1 + \frac{1}{u_0} \cdot (\frac{u_1}{u_0} - \sqrt{u_0}) + 0 \cdot \frac{u_2}{u_0} \\ &= -\frac{3}{2\sqrt{u_0}} = -\frac{3}{2c} \neq 0 \qquad \forall \mathbf{u} \in \mathbb{D}.\end{aligned}$$

For the 3-field,

$$\nabla\lambda_3(\mathbf{u}) \cdot \mathbf{r}_3(\mathbf{u}) = \frac{3}{2c} \neq 0 \qquad \forall \mathbf{u} \in \mathbb{D}$$

is obtained analogously. □

Generally, it cannot be stated that the k-field of partial differential equations is always either linearly degenerate or genuinely non-linear, as can be seen on the equation

$$\partial_t \mathbf{u} + \partial_{x_1} \mathbf{f}(\mathbf{u}) = \partial_t \mathbf{u} + \partial_{x_1} \begin{pmatrix} u_1^2 + u_1^3 \\ u_2^2 - u_2^3 \end{pmatrix} = 0.$$

In this example, $\mathbf{f}'(\mathbf{u}) = diag(2u_1 + 3u_1^2, 2u_2 - 3u_2^2)$ holds, and thus

$$\begin{aligned} \lambda_1(\mathbf{u}) &= 2u_1 + 3u_1^2 & \mathbf{r}_1(\mathbf{u}) &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \lambda_2(\mathbf{u}) &= 2u_2 - 3u_2^2 & \mathbf{r}_2(\mathbf{u}) &= \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \end{aligned}$$

which shows that this is system of partial differential equations is hyperbolic for all $\mathbf{u} \in \mathbb{R}^2$. On the other hand, it holds that

$$\begin{aligned}\nabla\lambda_1(\mathbf{u}) \cdot \mathbf{r}_1(\mathbf{u}) &= 2 + 6u_1 \text{ and} \\ \nabla\lambda_2(\mathbf{u}) \cdot \mathbf{r}_2(\mathbf{u}) &= 2 - 6u_2,\end{aligned}$$

and thus the nature of the k-field, $k = 1, 2$, depends on $\mathbf{u}$.

It is feasible, as [GoR96] proves, to use the equations in primitive variables for the analysis of eigenvectors and eigenvalues. In this case, the given notation allows to integrate the topography term *top* into the equation. This is quite important for the later analysis of the wave related to the eigenvalue that stems from the topography. In terms of primitive variables, the x_1-split 2D shallow water equations with the additional term $\partial_t top = 0$ for the bottom topography read

$$\partial_t H + H\partial_{x_1} v_1 + v_1 \partial_{x_1} H = 0$$

$$\begin{aligned}\partial_t v_1 + v_1\partial_{x_1} v_1 + g\partial_{x_1}(H + top) =&0\\ \partial_t v_2 + v_1\partial_{x_1} v_2 =&0\\ \partial_t top =&0,\end{aligned}$$

or

$$\partial_t \mathbf{w} + \mathbf{A}(\mathbf{w})\partial_{x_1}\mathbf{w} = \mathbf{0}$$

with

$$\mathbf{w} = \begin{pmatrix} H \\ v_1 \\ v_2 \\ top \end{pmatrix} \text{ and } \mathbf{A}(\mathbf{w}) = \begin{pmatrix} v_1 & H & 0 & 0 \\ g & v_1 & 0 & g \\ 0 & 0 & v_1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The characteristic polynomial for $\mathbf{A}(\mathbf{w})$ is

$$\begin{aligned} p_{\text{char}}(\lambda) &= \begin{vmatrix} \lambda - v_1 & -H & 0 & 0 \\ -g & \lambda - v_1 & 0 & -g \\ 0 & 0 & \lambda - v_1 & 0 \\ 0 & 0 & 0 & \lambda \end{vmatrix} \\ &= (\lambda - v_1)\lambda\left[(\lambda - v_1)^2 - gH\right] \\ &= (\lambda - v_1)\lambda(\lambda - (v_1 - c))(\lambda - (v_1 + c)), \end{aligned}$$

thus the eigenvalues of $\mathbf{A}(\mathbf{w})$ again are

$$\lambda_1(\mathbf{w}) = v_1 - c,\ \lambda_2(\mathbf{w}) = v_1,\ \lambda_3(\mathbf{w}) = v_1 + c \text{ and } \lambda_4(\mathbf{w}) = 0. \tag{1.26}$$

The associated eigenvectors are

$$\mathbf{r}_1(\mathbf{w}) = \alpha_1 \begin{pmatrix} H \\ -c \\ 0 \\ 0 \end{pmatrix},\ \mathbf{r}_2(\mathbf{w}) = \alpha_2 \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix},$$

$$\mathbf{r}_3(\mathbf{w}) = \alpha_3 \begin{pmatrix} H \\ c \\ 0 \\ 0 \end{pmatrix} \text{ and } \mathbf{r}_4(\mathbf{w}) = \alpha_4 \begin{pmatrix} -H \\ v_1 \\ 0 \\ H - \frac{v_1^2}{g} \end{pmatrix}.$$

If v_1 becomes zero, the system is no longer strictly hyperbolic, but even then stays hyperbolic as long as $\mathbf{w} \in \mathbb{D} \times \mathbb{R}$.

Obviously, the 2-field is linearly degenerate as before. For the 1-field (and analogously for the 3-field) the equation

$$\nabla\lambda_1(\mathbf{w}) \cdot \mathbf{r}_1(\mathbf{w}) = -\sqrt{\frac{g}{H}}H + 1(-c) + 0 + 0 = -2c \neq 0 \; \forall \mathbf{w} \in \mathbb{D} \times \mathbb{R}$$

holds, so the 1-field and the 3-field are genuinely non-linear. The 4-field again is linearly degenerate, which follows directly from $\lambda_4(\mathbf{w}) = 0$.

1.2.4 Generalized Riemann Invariants

The generalized Riemann invariants are a set of relations that remain constant across the wave structure for rarefaction waves and contact discontinuities. This trait, together with the Rankine-Hugoniot conditions discussed in section 1.2.5, will be exploited in the design of Riemann solvers in the sections 2.4.1, 2.4.2 and 2.5.3.

The Riemann invariants were first obtained by Riemann in [Rie60]. In that work, the author pursues the question of the evolution of differences in pressure at gases. Initial value problems of that type, that (more generally) consist of two constant states separated by a discontinuity, are called Riemann problems nowadays.

Definition 1.7 (Riemann problem)**.** Let $\mathbf{u}_L$ and $\mathbf{u}_R$ be two states of $\mathbb{D} \subset \mathbb{R}^p$. The initial value problem

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + \frac{\partial}{\partial x}\mathbf{f}(\mathbf{u}) = \mathbf{0}, \\ \mathbf{u}(x,0) = \begin{cases} \mathbf{u}_L, & x \leq 0 \\ \mathbf{u}_R, & x > 0 \end{cases} \end{cases} \tag{1.27}$$

is called *Riemann problem.*

Considering piecewise smooth continuous functions $\mathbf{u}\colon (x,t) \mapsto \mathbf{u}(x,t)$ that solve (1.27), [GoR96] shows that for *self-similar solutions* of (1.27) of the form

$$\mathbf{u}(x,t) = \mathbf{s}\left(\frac{x}{t}\right)$$

either

$$\mathbf{s}'\left(\frac{x}{t}\right) = \mathbf{0}$$

holds or that there exists an index $k \in \{1, ..., p\}$ such that

$$\mathbf{s}'\left(\frac{x}{t}\right) = \alpha\left(\frac{x}{t}\right)\mathbf{r}_k\left(\mathbf{s}\left(\frac{x}{t}\right)\right), \qquad \lambda_k\left(\mathbf{s}\left(\frac{x}{t}\right)\right) = \frac{x}{t}. \tag{1.28}$$

By differentiating the second equation with respect to $\frac{x}{t}$ and inserting the first equation, the expression

$$\alpha\left(\frac{x}{t}\right)\nabla\lambda_k\left(\mathbf{s}\right)\cdot\mathbf{r}_k\left(\mathbf{s}\left(\frac{x}{t}\right)\right)=1$$

is obtained. This equation can only be solved if the k-th field is genuinely non-linear. In this case, it is also possible to normalize the equation with $\alpha\left(\frac{x}{t}\right)=1$. Then, $\mathbf{s}$ is an integral curve of the vector field $\mathbf{r}_k(\mathbf{s})$, as $\mathbf{s}'=\mathbf{r}_k\left(\mathbf{s}\right)$ holds.

Thus, assuming $\mathbf{u}_L$ and $\mathbf{u}_R$ are on the same integral curve of $\mathbf{r}_k$ and that λ_k increases from $\mathbf{u}_L$ to $\mathbf{u}_R$ along this curve, the function

$$\mathbf{u}(x_1,t)=\begin{cases}\mathbf{u}_L, & \frac{x_1}{t}\leq\lambda_k(\mathbf{u}_L)\\ \mathbf{s}\left(\frac{x_1}{t}\right), & \lambda_k(\mathbf{u}_L)<\frac{x_1}{t}\leq\lambda_k(\mathbf{u}_R)\\ \mathbf{u}_R, & \lambda_k(\mathbf{u}_R)<\frac{x_1}{t}\end{cases}\tag{1.29}$$

with $\mathbf{s}\left(\frac{x_1}{t}\right)$ being an integral curve of $\mathbf{r}_k$ and

$$\mathbf{s}(\lambda_k(\mathbf{u}_L))=\mathbf{u}_L,\qquad\mathbf{s}(\lambda_k(\mathbf{u}_R))=\mathbf{u}_R$$

is a continuous self-similar weak solution of (1.27).

Definition 1.8. Such a self-similar weak solution (1.29) of (1.27) is called a k-centered simple wave or a k-rarefaction wave connecting the states $\mathbf{u}_L$ and $\mathbf{u}_R$.

Apart from the k-field being linearly degenerated or genuinely non-linear, Riemann invariants are an important tool for determining the full solution of a Riemann problem. They provide relations between states under certain conditions.

Definition 1.9. A smooth function $\rho_k\colon\mathbb{D}\to\mathbb{R}$ is called a *k-Riemann invariant* if it satisfies

$$\nabla\rho_k(\mathbf{u})\cdot\mathbf{r}_k(\mathbf{u})=0\;\forall\mathbf{u}\in\mathbb{D}.\tag{1.30}$$

Remark 1.10. *If the k-field is linearly degenerated, λ_k is a Riemann invariant.*

Theorem 1.11. *On a k-rarefaction wave, all k-Riemann invariants ρ_k are constant. In particular,*

$$\rho_k(\mathbf{u}_L)=\rho_k(\mathbf{u}_R)$$

holds.

Proof. Let $\mathbf{u}$ be a k-rarefaction wave of the form (1.29) and let $\mathbf{s}\left(\frac{x}{t}\right)$ be the integral curve of $\mathbf{r}_k$ that connects the states $\mathbf{u}_L$ and $\mathbf{u}_R$.

Obviously, for $\frac{x}{t} \leq \lambda_k(\mathbf{u}_L)$ and for $\frac{x}{t} \geq \lambda_k(\mathbf{u}_R)$, $\rho_k(\mathbf{u}(\frac{x}{t}))$ is constant. For the derivative of ρ_k along $\mathbf{s}\left(\frac{x}{t}\right)$ the equation (1.28) yields

$$\frac{d}{d\frac{x}{t}}\rho_k\left(\mathbf{s}\left(\frac{x}{t}\right)\right) = \nabla\rho_k\left(\mathbf{s}\right)\cdot\mathbf{s}'\left(\frac{x}{t}\right) = \nabla\rho_k\left(\mathbf{s}\right)\cdot\mathbf{r}_k\left(\mathbf{s}\left(\frac{x}{t}\right)\right) = 0.$$

This means that ρ_k is constant along the trajectories of the vector field $\mathbf{r}_k$. As $\mathbf{u}$ is continuous and ρ_k smooth, ρ_k is thus constant on the k-rarefaction wave and thus also for the states $\mathbf{u}_L$ and $\mathbf{u}_R$. □

[GoR96] proves that locally there exist $(p-1)$ k-Riemann invariants, whose gradients are linearly independent if $\mathbf{u} \in \mathbb{D} \subset \mathbb{R}^p$.

In the case of the x_1-split 2D shallow water equations with added topography, this means that for every eigenvalue λ_k, $k = 1, ..., 4$, there exist three k-Riemann invariants ρ_k^i, $i = 1, 2, 3$. The 1-Riemann invariants ρ_1^i, $i = 1, 2, 3$, have to satisfy the equation

$$(\nabla\rho_1^i)\cdot\mathbf{r}_1 = H\cdot\partial_H\rho_1^i - c\cdot\partial_{v_1}\rho_1^i + 0\cdot\partial_{v_2}\rho_1^i + 0\cdot\partial_b\rho_1^i = 0,\ i = 1, 2, 3,$$

which yields

$$\begin{aligned}\rho_1^1 &= v_2,\\ \rho_1^2 &= v_1 + 2c,\\ \rho_1^3 &= b.\end{aligned}\tag{1.31}$$

Analogously, it is obtained for $k = 2$ that

$$(\nabla\rho_2^i)\cdot\mathbf{r}_2 = 0\cdot\partial_H\rho_2^i + 0\cdot\partial_{v_1}\rho_2^i + 1\cdot\partial_{v_2}\rho_2^i + 0\cdot\partial_b\rho_2^i = 0,\ i = 1, 2, 3,$$

and thus

$$\begin{aligned}\rho_2^1 &= H,\\ \rho_2^2 &= v_1,\\ \rho_2^3 &= b,\end{aligned}\tag{1.32}$$

for $k = 3$ that

$$(\nabla\rho_3^i)\cdot\mathbf{r}_3 = H\cdot\partial_H\rho_3^i + c\cdot\partial_{v_1}\rho_3^i + 0\cdot\partial_{v_2}\rho_3^i + 0\cdot\partial_b\rho_3^i = 0,\ i = 1, 2, 3,$$

which leads to

$$\begin{aligned}\rho_3^1 &= v_2,\\ \rho_3^2 &= v_1 - 2c,\\ \rho_3^3 &= b,\end{aligned}\tag{1.33}$$

and for $k = 4$ that

$$(\nabla\rho_4^i)\cdot\mathbf{r}_4 = -H\cdot\partial_H\rho_4^i+v_1\cdot\partial_{v_1}\rho_4^i+0\cdot\partial_{v_2}\rho_4^i+\left(H-\frac{v_1^2}{g}\right)\cdot\partial_b\rho_4^i = 0,\ i=1,2,3,$$

which at last gives

$$\begin{aligned}\rho_4^1 &= Hv_1,\\ \rho_4^2 &= v_2,\\ \rho_4^3 &= gH+\frac{v_1^2}{2}+gb.\end{aligned}\tag{1.34}$$

1.2.5 Rankine-Hugoniot Conditions

Analogously to the Riemann invariants, the Rankine-Hugoniot conditions are a set of equations that describe relations across the wave structure, though across shock waves and contact discontinuities. More precisely, a piecewise continuous function $\mathbf{u}$ is a weak solution of the partial differential equation (1.24), if it satisfies the Rankine-Hugoniot conditions along its lines of discontinuity [GoR96]. They were developed independently by Rankine in [Ran70] and Hugoniot in [Hug87, Hug89]. By contrast to the Riemann invariants, the Rankine-Hugoniot conditions must not be derived from the equations in primitive variables, as that would yield wrong results for the shock speed in the case of a discontinuous solution. This was demonstrated by Toro in [Tor01].

The proposition of the Rankine-Hugoniot conditions is that for two constant states $\mathbf{u}_L$ and $\mathbf{u}_R$ that are to the left and to the right of a line of discontinuity, the relation

$$\mathbf{f}_1(\mathbf{u}_R)-\mathbf{f}_1(\mathbf{u}_L) = S(\mathbf{u}_L,\mathbf{u}_R)\,(\mathbf{u}_R-\mathbf{u}_L)\tag{1.35}$$

holds. $S(\mathbf{u}_L,\mathbf{u}_R)$ is the velocity of propagation of the discontinuity.

A discontinuous weak solution of (1.27) therefore has the form

$$\mathbf{u}\,(x,t) = \begin{cases}\mathbf{u}_L, & x/t < S(\mathbf{u}_L,\mathbf{u}_R),\\ \mathbf{u}_R, & S(\mathbf{u}_L,\mathbf{u}_R) < x/t.\end{cases}\tag{1.36}$$

The set of all states $\mathbf{u}\in\mathbb{D}$ that can be connected to a given state $\mathbf{u}_0\in\mathbb{D}$ via a discontinuity is called the Rankine-Hugoniot set, [GoR96].

Definition 1.12 (Rankine-Hugoniot Set)**.** The *Rankine-Hugoniot set* of a state $\mathbf{u}_0\in\mathbb{D}$ is the set of all states $\mathbf{u}\in\mathbb{D}$ such that there exists $S(\mathbf{u}_0,\mathbf{u})\in\mathbb{R}$ with

$$S(\mathbf{u}_0,\mathbf{u})\,(\mathbf{u}-\mathbf{u}_0) = \mathbf{f}\,(\mathbf{u})-\mathbf{f}\,(\mathbf{u}_0)\,.\tag{1.37}$$

Theorem 1.13. *Let $\mathbf{u}_0$ be in $\mathbb{D} \subset \mathbb{R}^p$. The Rankine-Hugoniot set of $\mathbf{u}_0$ is locally made of p smooth curves $\mathcal{H}_k(\mathbf{u}_0)$, $1 \leq k \leq p$. Moreover, for all k, there exists a parametrization of $\mathcal{H}_k(\mathbf{u}_0) : \varepsilon \to \Psi_k(\varepsilon)$ defined for $|\varepsilon| \leq \varepsilon_1$, ε_1 small enough, such that*

$$\Psi_k(\varepsilon) = \mathbf{u}_0 + \varepsilon \mathbf{r}_k(\mathbf{u}_0) + \frac{\varepsilon^2}{2} \nabla \mathbf{r}_k(\mathbf{u}_0) \cdot \mathbf{r}_k(\mathbf{u}_0) + \mathcal{O}(\varepsilon^3) \tag{1.38}$$

and

$$S\left(\mathbf{u}_0, \Psi_k(\varepsilon)\right) = \lambda_k(\mathbf{u}_0) + \frac{\varepsilon}{2} \nabla \lambda_k(\mathbf{u}_0) \cdot \mathbf{r}_k(\mathbf{u}_0) + \mathcal{O}(\varepsilon^2).$$

Proof. [GoR96] □

Another important property that is proven in [GoR96] concerns the relations between states of a special kind of discontinuity, the contact discontinuity.

Theorem 1.14. *If the k-field is linearly degenerated and $\mathbf{u} \in \mathcal{H}_k(\mathbf{u}_0)$, then*

$$S\left(\mathbf{u}_0, \mathbf{u}\right) = \lambda_k(\mathbf{u}) = \lambda_k(\mathbf{u}_0) \tag{1.39}$$

and $\rho_k(\mathbf{u}_0) = \rho_k(\mathbf{u})$ holds for any k-Riemann invariant ρ_k.

Following this theorem, there will not arise any problems from the fact that the function *top* for the bottom is not included in the set of conservative variables. The only place in the model where the bottom is not continuous will per definition be the jump of topography between two computational cells. This is the case for the contact discontinuity with $\lambda_4 = 0$ which is linearly degenerated, so that the Riemann invariants hold in this case.

The following theorem from [Smo83] links the considerations made so far with the aim to generate the solution of the Riemann problem (1.27).

Theorem 1.15. *Let $\mathbf{u}_L \in \mathbb{D} \subset \mathbb{R}^p$ and suppose that the system* (1.24) *is hyperbolic and that each characteristic field is either genuinely nonlinear or linearly degenerate in $\mathbb{D}$. Then there is a neighborhood $\tilde{\mathbb{D}} \subset \mathbb{D}$ of $\mathbf{u}_L$ such that if $\mathbf{u}_R \subset \tilde{\mathbb{D}}$, the Riemann problem* (1.27) *has a solution. This solution consists of at most $(p+1)$ constant states separated by shocks, centered simple waves or contact discontinuities. There is precisely one physically relevant solution of this kind in $\mathbb{D}$.*

So far, continuous and discontinuous weak solutions for waves connecting states $\mathbf{u}$ to a given state $\mathbf{u}_0$ were discussed, but no condition to determine which is the right one in the physical sense was given yet. This feature becomes important especially in the context of theorem 1.15, where the existence of a unique sequence of states $\mathbf{u}_i$, $i = 1, ..., p-1$, is stated that connect two initial states $\mathbf{u}_L$ and $\mathbf{u}_R$.

For example, consider a conservation law with $p = 2$ that has the treat that each characteristic field is either genuinely nonlinear or linearly degenerate. Apart from the case $\mathbf{u}_L$ and $\mathbf{u}_R$ being identical, there exist four possibilities for the connecting state $\mathbf{u}_1$:

$$\begin{tikzcd}
\mathbf{u}_L \arrow[r, "1-shock"] & \mathbf{u}_1 & \mathbf{u}_R \arrow[l, "2-shock"'] \\
\mathbf{u}_L \arrow[r, "1-shock"] & \mathbf{u}_1 & \mathbf{u}_R \arrow[l, "2-rarefaction"'] \\
\mathbf{u}_L \arrow[r, "1-rarefaction"] & \mathbf{u}_1 & \mathbf{u}_R \arrow[l, "2-shock"'] \\
\mathbf{u}_L \arrow[r, "1-rarefaction"] & \mathbf{u}_1 & \mathbf{u}_R, \arrow[l, "2-rarefaction"']
\end{tikzcd}$$

and in each of these cases $\mathbf{u}_1$ can be expected to be different. Thus, a criterion to sort out which is the correct sequence is needed.

In a physically relevant solution of the Riemann problem (1.27), the particles of the fluid cross a shock from its front to its back, [Lax57]. The entropy of these particles has to increase in crossing a shock [CoF44]. The criterion that allows to determine whether the k-waves, $k = 1, ..., p$, are shock waves or not, and thus to find the physically relevant weak solution of (1.27), is the *Lax entropy condition.*

Theorem 1.16 (Lax entropy condition, [Lax57])**.** *A jump discontinuity in a weak solution is called a shock if the total number of characteristics drawn in this fashion is $p-1$. The k-shock wave* (1.36) *satisfies the entropy condition if the inequalities*

$$\lambda_{k-1}(\mathbf{u}_L) < S < \lambda_k(\mathbf{u}_L) \tag{1.40a}$$

$$\lambda_k(\mathbf{u}_R) < S < \lambda_{k+1}(\mathbf{u}_R) \tag{1.40b}$$

hold with S being the shock speed given by (1.35)*.*

As a direct conclusion, the following remark can be obtained.

Remark 1.17. *For a k-shock with velocity S,*

$$\lambda_k(\mathbf{u}_L) > S > \lambda_k(\mathbf{u}_R)$$

holds.

For the shallow water equations, the application of the Rankine-Hugoniot conditions (1.37) leads to the equations

$$\begin{aligned} u_1 - u_{0;1} &= Su_0 - Su_{0;0} \\ \frac{u_1^2}{u_0} + \frac{1}{2}u_0^2 - \frac{u_{0;1}^2}{u_{0;0}} - \frac{1}{2}u_{0;0}^2 &= Su_1 - Su_{0;1} \\ \frac{u_1u_2}{u_0} - \frac{u_{0;1}u_{0;2}}{u_{0;0}} &= Su_2 - Su_{0;2} \end{aligned}$$

or, when inserting the conservative variables, to

$$\Phi v_1 - \Phi_0 v_{0;1} = S\left(\Phi - \Phi_0\right) \tag{1.41a}$$

$$\Phi v_1^2 + \frac{1}{2}\Phi^2 - \Phi_0 v_{0;1}^2 - \frac{1}{2}\Phi_0^2 = S\left(\Phi v_1 - \Phi_0 v_{0;1}\right) \tag{1.41b}$$

$$\Phi v_1 v_2 - \Phi_0 v_{0;1} v_{0;2} = S\left(\Phi v_2 - \Phi_0 v_{0;2}\right). \tag{1.41c}$$

Sorting these equations by $\mathbf{u}/\mathbf{u}_0$-terms leads to

$$\Phi(v_1 - S) = \Phi_0(v_{0;1} - S) \tag{1.42a}$$

$$\Phi v_1(v_1 - S) + \frac{1}{2}\Phi^2 = \Phi_0 v_{0;1}(v_{0;1} - S) + \frac{1}{2}\Phi_0^2 \tag{1.42b}$$

$$\Phi v_2(v_1 - S) = \Phi_0 v_{0,2}(v_{0;1} - S). \tag{1.42c}$$

As the 2-field is linearly degenerated, theorem 1.14 and equation (1.25) imply that

$$S_2 = v_{0;1} = v_1$$

for the 2-wave. Equation (1.42) then reduces to another 2-Riemann invariant, $\Phi = \Phi_0$, as $\Phi \in \mathbb{R}_0^+$.

For the 1- and 3-waves, it follows from theorem 1.16 that $(v_{0;1} - S) \neq 0$ and $(v_1 - S) \neq 0$. Thus, from the equations (1.42a) and (1.42c) it follows directly that

$$v_2 = v_{0;2}, \tag{1.43}$$

and from equation (1.41a) by eliminating S using equation (1.41b) that

$$v_1 - v_{0;1} = \pm(\Phi - \Phi_0)\sqrt{\frac{\Phi + \Phi_0}{2\Phi\Phi_0}}. \tag{1.44}$$

Depending on the algebraic sign, equation (1.44) results in two curves $\mathcal{H}_k(\mathbf{u}_0)$, $k \in \{1, 3\}$ of the Rankine-Hugoniot set of states connected to $\mathbf{u}_0$ via a shock. From theorem 1.13 it is known that $\mathcal{H}_k(\mathbf{u}_0)$ is tangent to the eigenvector $\mathbf{r}_k(\mathbf{u}_0)$ at $\mathbf{u}_0$. Thus, it is possible to determine which curve belongs to a 1-

and which to a 3-shock. By multiplying (1.44) with Φ and replacing Φ with $\Phi_0 + \varepsilon$ on the right hand side of the equation, the parametrization

$$\Phi v_1 = \Phi_0 v_{0;1} + \varepsilon \left(v_{0;1} \pm \sqrt{\Phi_0 + \varepsilon \left(1 + \frac{\Phi_0 + \varepsilon}{2\Phi_0}\right)} \right)$$

is obtained. Analogous manipulations of (1.43) lead to

$$\mathbf{u} = \mathbf{u}_0 + \varepsilon \begin{pmatrix} 1 \\ v_{0;1} \pm \sqrt{\Phi_0 + \mathcal{O}(\varepsilon)} \\ v_{0;2} \end{pmatrix} \quad \text{as } \varepsilon \to 0.$$

Since $\mathbf{r}_1(\mathbf{u}_0) = (1, v_{0;1} - \sqrt{\Phi_0}, v_{0;2})^T$, the equation

$$v_1 = v_{0;1} - (\Phi - \Phi_0)\sqrt{\frac{\Phi + \Phi_0}{2\Phi\Phi_0}} \tag{1.45}$$

holds for the states connected by a 1-shock. For the 3-shock

$$v_1 = v_{0;1} + (\Phi - \Phi_0)\sqrt{\frac{\Phi + \Phi_0}{2\Phi\Phi_0}}. \tag{1.46}$$

holds by similar considerations.

After having assigned the equations to the corresponding shocks, it is necessary to analyze the restrictions concerning their physical admissibility due to theorem 1.16.

For the 1-shock, the shock velocity

$$S_1 = v_{0;1} - \sqrt{\frac{\Phi}{\Phi_0}}\sqrt{\frac{\Phi + \Phi_0}{2}} = v_1 - \sqrt{\frac{\Phi_0}{\Phi}}\sqrt{\frac{\Phi + \Phi_0}{2}}$$

can be derived by inserting (1.45) into (1.41a) and solving the resulting equation with respect to $S = S_1$. Comparing S_1 to (1.40a) and (1.40b), the condition

$$S_1 < \lambda_2(\mathbf{u}) = v_1$$

is obviously fulfilled, while the remaining two conditions of theorem 1.16 imply

$$S_1 < \lambda_1(\mathbf{u}_0) \qquad \Leftrightarrow \sqrt{\frac{\Phi}{\Phi_0}}\sqrt{\frac{\Phi + \Phi_0}{2}} > \sqrt{\Phi_0} \qquad \Leftrightarrow \Phi > \Phi_0$$

$$S_1 > \lambda_1(\mathbf{u}) \qquad \Leftrightarrow \sqrt{\frac{\Phi_0}{\Phi}}\sqrt{\frac{\Phi+\Phi_0}{2}} < \sqrt{\Phi} \qquad \Leftrightarrow \Phi > \Phi_0.$$

Similar manipulations of (1.46) for the 3-wave lead again to the restriction $\Phi > \Phi_0$ for the connection of the states $\mathbf{u}_0$ and $\mathbf{u}$ by a physically admissible 3-shock.

In all other cases, that is if $\Phi \leq \Phi_0$, the states $\mathbf{u}_0$ and $\mathbf{u}$ are connected by a rarefaction wave. Then the Riemann invariants (1.31) and (1.33), respectively, determine the according relation.

Thus, for the set of (primitive) states $\mathbf{w}$ that can be connected by a 1-wave to a given initial state $\mathbf{w}_L$ the relations

$$\begin{aligned} v_1 &= \begin{cases} v_{L;1} - 2(c - c_L) & \text{if } H \leq H_L \\ v_{L;1} - (H - H_L)\sqrt{\frac{g(H+H_L)}{2HH_L}} & \text{if } H > H_L \end{cases} \\ v_2 &= v_{L;2} \end{aligned} \tag{1.47}$$

hold. This set will be referred to as RL.

Analogously, for the states $\mathbf{w}$ of the set RR that can be connected by a 3-wave to a given initial state $\mathbf{w}_R$

$$\begin{aligned} v_1 &= \begin{cases} v_{R;1} + 2(c - c_R) & \text{if } H \leq H_R \\ v_{R;1} + (H - H_R)\sqrt{\frac{g(H+H_R)}{2HH_R}} & \text{if } H > H_R \end{cases} \\ v_2 &= v_{R;2} \end{aligned} \tag{1.48}$$

hold.

Regarding theorem 1.15, it is now possible to determine the sequence of states $\mathbf{u}_1$ and $\mathbf{u}_2$ solving the Riemann problem with initial values $\mathbf{u}_L$ and $\mathbf{u}_R$ in the following way:

For the primitive representation $\mathbf{w}_1$ and $\mathbf{w}_2$ of this sequence, with equation (1.32) it holds that

$$w_{1;0} = w_{2;0} := w_0 \text{ and } w_{1;1} = w_{2;1} := w_1,$$

while with (1.43), (1.31) and (1.33)

$$w_{L;2} = w_{1;2} \text{ and } w_{2;2} = w_{R;2}$$

holds. Thus, computing w_0 and w_1 by intersecting RL with RR completes the solution.

The considerations made above, though valid for all Riemann problems with positive water height throughout the whole solution, do not include the influence of the 4-wave that is induced by topography. In contrast to the 1-, 2- and 3-wave, that have varying velocities but a fixed order, the 4-wave always has the velocity $S_4 = \lambda_4 = 0$ but its position in the system of waves varies. The 4-wave and its inclusion into the solution of the Riemann problem will be discussed later in section 2.5.3. The treatment of Riemann problems containing states with a water height of zero, so-called dry-bed states, in either the initial values or the solution is covered in section 2.5.4.

1.3 Finite Volume Schemes

The aim of this work is to present a high order finite volume scheme for the numerical solution of the 2D shallow water equations with topography.

The principle of finite volume schemes is based directly on the integral form (1.19) of the 2D shallow water equations. This is an important advantage especially for hyperbolic systems of partial differential equations, as it is known for these equations that even smooth initial conditions may lead to discontinuous solutions in finite time. In that case, the integral form of the equations still holds.

The idea of finite volume schemes is to divide the domain of integration $\Omega \subset \mathbb{R}^2$ into a finite number of smaller cells or control volumes σ_i and compute the evolution of the integral cell mean values $\overline{\mathbf{u}}_i$ of the set of variables during the integration period. The solution then consists of cell mean values. Thus, the spatial resolution of the solution depends on the grade of refinement of this division. Obviously, the most basic form of gaining a piecewise continuous approximation of $\mathbf{u}$ from $\overline{\mathbf{u}}_i$ is interpreting the solution of a finite volume scheme as a piecewise constant function on the domain of integration.

Let σ_i be an arbitrary control volume that is polygonally bounded by

$$\delta\sigma_i = \sum_{k=1}^{N_i} \delta^{(k)}\sigma_i.$$

N_i is the number of edges of the polygon and $\delta^{(k)}\sigma_i$ denotes such an edge with outer normal vector $\mathbf{n}^{(k)}$.

For the equation in integral form

$$\frac{d}{dt}\int_{\sigma_i} \mathbf{u}\ d\mathbf{x} + \int_{\delta\sigma_i} \sum_{j=1}^{2} \mathbf{f}_j(\mathbf{u}) n_j\ ds = \int_{\sigma_i} \mathbf{g}(\mathbf{u})\ d\mathbf{x},$$

the property 1.2 of rotational invariance implies

$$\frac{d}{dt}\int_{\sigma_i} \mathbf{u}\ d\mathbf{x} + \int_{\delta\sigma_i} \mathbf{T}(\mathbf{n})^{-1}\mathbf{f}_1(\mathbf{T}(\mathbf{n})\mathbf{u})\ ds = \int_{\sigma_i} \mathbf{g}(\mathbf{u})\ d\mathbf{x}.$$

Writing the contour integral edgewise, integrating the whole equation in time from t^n to t^{n+1} and multiplying by the inverse of the area of σ_i, $|\sigma_i|^{-1}$, yields

$$\begin{aligned} & |\sigma_i|^{-1} \int_{t^n}^{t^{n+1}} \frac{d}{dt} \int_{\sigma_i} \mathbf{u}\ d\mathbf{x}\ dt \\ & + |\sigma_i|^{-1} \int_{t^n}^{t^{n+1}} \sum_{k=1}^{N_i} \mathbf{T}(\mathbf{n}^{(k)})^{-1} \int_{\delta^{(k)}\sigma_i} \mathbf{f}_1(\mathbf{T}(\mathbf{n}^{(k)})\mathbf{u})\ ds\ dt \\ = \ & |\sigma_i|^{-1} \int_{t^n}^{t^{n+1}} \int_{\sigma_i} \mathbf{g}(\mathbf{u})\ d\mathbf{x}\ dt. \end{aligned}$$

Define

$$\overline{\mathbf{u}}_i^n := |\sigma_i|^{-1} \int_{\sigma_i} \mathbf{u}(\mathbf{x}, t^n)\, d\mathbf{x}$$

as the *integral cell mean value* of $\mathbf{u}$ at time level t^n. The equation then transforms into

$$\begin{aligned} \overline{\mathbf{u}}_i^{n+1} - \overline{\mathbf{u}}_i^n \ = \ & -|\sigma_i|^{-1} \int_{t^n}^{t^{n+1}} \sum_{k=1}^{N_i} \mathbf{T}(\mathbf{n}^{(k)})^{-1} \int_{\delta^{(k)}\sigma_i} \mathbf{f}_1(\mathbf{T}(\mathbf{n}^{(k)})\mathbf{u})\ ds\ dt \\ & + |\sigma_i|^{-1} \int_{t^n}^{t^{n+1}} \int_{\sigma_i} \mathbf{g}(\mathbf{u})\ d\mathbf{x}\ dt. \end{aligned}$$

Finally, let N_x and N_t be the number of nodes in space and time of the Gauss quadrature rules, α_j the weights and $\tilde{\mathbf{x}}^{k,l}$ on $\delta^{(k)}\sigma_i$, $l = 1, ..., N_x$, and $t^{n,m} \in [t^n, t^{n+1}]$, $m = 1, ..., N_t$, the associated nodes in space and time respectively. Approximating the integrals by these rules results in

$$\overline{\mathbf{u}}_i^{n+1} \approx \overline{\mathbf{u}}_i^n - |\sigma_i|^{-1}\ \mathit{update} \tag{1.49}$$

with

$$\begin{aligned} \mathit{update} = & \sum_{k=1}^{N_k} \mathbf{T}(\mathbf{n}^{(k)})^{-1} \left\{ \left|\delta^{(k)}\sigma_i\right| \sum_{l=1}^{N_x} \alpha_l \left[\Delta t \sum_{m=1}^{N_t} \alpha_m \mathbf{f}_1(\mathbf{T}(\mathbf{n}^{(k)})\mathbf{u}(\tilde{\mathbf{x}}^{k,l}, t^{n,m})) \right] \right\} \\ & + \int_{t^n}^{t^{n+1}} \int_{\sigma_i} \mathbf{g}(\mathbf{u})\ d\mathbf{x}\ dt. \end{aligned} \tag{1.50}$$

The i-th component of *update* consists of the quantity of u_i that 'flowed' across the cell boundaries, that is the numerical flux, and the amount that was produced by source terms. *update* represents the net amount of change of the quantities considered in $\mathbf{u}$ in the cell σ_i between time t^n and t^{n+1}. Dividing *update* by $|\sigma_i|$ the cell mean value of change is obtained. Thus, the result $\overline{\mathbf{u}}_i^{n+1}$ of equation (1.49) is the cell mean value of the quantities $\mathbf{u}$ at time t^{n+1}.

The aim of this work is to present a high order finite volume scheme for the 2D shallow water equations. The expectation is of course to obtain a more accurate numerical solution. It makes sense in this context to distinguish between accuracy in the sense of resolution and accuracy in the sense of computing more exact updates. As this work is about finite volume schemes, whose solution consists generally of cell mean values, accuracy is to be understood in the latter sense.

Having a close look at equation (1.50), there arise some questions from that formulation, especially when accounting for the aim to develop a scheme of higher order than one:

- Aiming for the computation of more accurate updates calls for the use of higher order quadrature rules.
- A higher order quadrature rule only makes sense when the function to be integrated is not just constant, in space as well as in time.
- There exist cell mean values for all variables for t^n, but for an appropriate scheme the values $\mathbf{u}(\tilde{\mathbf{x}}^{k,l}, t^{n,m})$ are needed, which not only afford a spatial approximation of $\mathbf{u}$ but also a prediction of its time dependent course as the integration points belong to later time levels than t^n.
- Assuming higher order approximations for the quantities contained in $\mathbf{u}$ are available for each cell, for each integration point at the cell boundary there could be two different values $\mathbf{u}(\tilde{\mathbf{x}}^{k,l}, t^{n,m})$, as there are two adjacent cells.

These problems are treated in the numerical realization presented in chapter 2. Approximating the variables in each cell by a polynomial of an order appropriate to the desired order of the scheme allows to obtain approximations for higher derivatives of the functions in $\mathbf{u}$ at a given time level. The process of computing such a polynomial is called reconstruction and is treated in section 2.2.

Using these approximations for the higher derivatives and the partial differential equation itself, a Taylor expansion in space and time approximating the functions in $\mathbf{u}$ can be computed, as discussed in section 2.3. This allows a prediction of the evolution of the function $\mathbf{u}$ for each space-time integration point and each cell.

The Riemann problem, that stems from the fact that there are two predictions for each integration point, can be solved numerically using the methods presented in section 2.4. The methods presented there require a continuous topography and a water height greater than zero throughout the whole solution.

The treatment of Riemann problems that include discontinuities in topography or whose solution or initial conditions contain dry states is finally discussed in section 2.5.

Chapter 2

Numerical Realization

In this chapter, the numerical methods are developed which are used to cope with the problems that arise from the formulation (1.49), (1.50) and the intention to create a high order scheme.

However, at first a brief introduction of the computational setting, concerning the definition of some computational constants and a description of the computational grid including its properties and denotations, are given.

2.1 Parameters of the Scheme

To avoid the use of large amounts of indices, the usual multi index notation in two (and three) dimensions

$$\begin{aligned}
\boldsymbol{\alpha} &= (\alpha_1, ..., \alpha_n), \quad \alpha_i \geq 0 \text{ for } i = 1, ..., n, \\
\mathbf{x}^{\boldsymbol{\alpha}} &= \prod_{i=1}^{n} x_i^{\alpha_i}, \\
a_{\boldsymbol{\alpha}} &= a_{\alpha_1,...,\alpha_n}, \\
|\boldsymbol{\alpha}| &= \sum_{i=1}^{n} \alpha_i, \\
\boldsymbol{\alpha}! &= \prod_{i=1}^{n} \alpha_i!,
\end{aligned}$$

will be used.

Depending on the desired order of the scheme, the size of many parameters varies. These parameters remain constant during one run of the program, though. For a required order ord of the solution, the degree of the polynomials that is necessary to obtain this order will be $deg = ord - 1$. A polynomial in two dimensions

$$p(\mathbf{x}) = \sum_{|\boldsymbol{\alpha}|=0}^{deg} a_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}},$$

as it is used for the spatial reconstruction, then has

$$noc = \sum_{k=0}^{deg} (k+1) = \frac{(deg+1)(deg+2)}{2}$$

coefficients. As usual for polynomials these are arranged according to the degree of their dedicated monomial in a graded lexicographic order where

$$\begin{aligned} & position(a_{\boldsymbol{\alpha}}) > position(a_{\boldsymbol{\beta}}) \\ \Leftrightarrow\ & \boldsymbol{\alpha} > \boldsymbol{\beta} \\ \Leftrightarrow\ & (|\boldsymbol{\alpha}| > |\boldsymbol{\beta}|) \vee ((|\boldsymbol{\alpha}| = |\boldsymbol{\beta}|) \wedge (\alpha_2 > \beta_2)) \,. \end{aligned} \tag{2.1}$$

The polynomial is stored as an array containing the coefficients of the polynomial. The position of the coefficient $a_{\boldsymbol{\alpha}}$ in this array due to the monomial order is given by the function

$$\begin{aligned} pos :\ & \mathbb{N}^2 \to \mathbb{N} \\ & \boldsymbol{\alpha} \mapsto \frac{|\boldsymbol{\alpha}|\,(|\boldsymbol{\alpha}|+1)}{2} + \alpha_2. \end{aligned}$$

Thinking of the tuples $\boldsymbol{\alpha} \in \mathbb{N}^2$ as indices that indicate the row and column of its position in the $\mathbb{N} \times \mathbb{N}$-grid, the function *pos* is similar to Cantor's diagonal argument and thus bijective. For simplicity of notation the inverse function $pos^{-1} : \mathbb{N} \to \mathbb{N}^2$, that returns the multi-index $\boldsymbol{\alpha}$ from the number $pos(\boldsymbol{\alpha})$, is sometimes used in this work, but not in the program.

Order and access are defined analogously for three dimensional polynomials in two spatial and one time dimension as they are needed in the Taylor expansion for the prediction of the time dependent course of $\mathbf{u}$. The number of coefficients of those polynomials is

$$\begin{aligned} ntc =& \sum_{l=0}^{deg} \sum_{k=0}^{l} (k+1) \\ =& \sum_{l=0}^{deg} \frac{(l+1)(l+2)}{2} \\ =& \frac{(deg+1)(deg+2)(deg+3)}{6} \end{aligned}$$

and the access to the coefficient $a_{\boldsymbol{\alpha}}$, $|\boldsymbol{\alpha}| \leq deg$, of the monomial $x_1^{\alpha_1}x_2^{\alpha_2}t^{\alpha_3}$ is made via

$$\begin{aligned} tpos:\ & \mathbb{N}^3 \to \mathbb{N} \\ & \boldsymbol{\alpha} \mapsto \frac{|\boldsymbol{\alpha}|\,(|\boldsymbol{\alpha}|+1)(|\boldsymbol{\alpha}|+2)}{6} + \frac{(\alpha_2+\alpha_3)(\alpha_2+\alpha_3+1)}{2} + \alpha_3. \end{aligned}$$

To obtain a spatial polynomial of degree *deg* in the reconstruction, the set of the cells, or the stencil, whose information is taken into account has to have a size of at least *noc*. The scheme becomes more stable, however, if the polynomial is not computed exactly from a stencil of the minimal possible size but with a least square method from a larger stencil. [DuK07] suggests that the standard size of the stencils is set to

$$stencil\_size = \left\lceil \frac{3noc}{2} \right\rceil .$$

In order to obtain a reconstruction polynomial whose oscillation is the least possible for a cell σ_i, not only one but several stencils $S_{i,j}$ provide a possible reconstruction polynomial. The number of stencils for the cell σ_i is denoted by nos_i.

2.1.1 The Grid

The basis for all considerations in this and the following chapters is the computational domain Ω, which is polygonally bounded and carries two substructures: Firstly, a triangulation $T = \{\tau_j\}$ of Delaunay-Type, and secondly, based on this triangulation, a set of polygonally bounded cells $\Sigma = \{\sigma_i\}$ that correspond to the vertexes of the triangles. T is a conforming triangulation as it satisfies the conditions stated in the following definition, [IsS96].

Definition 2.1. A *triangulation* T on Ω is a decomposition of Ω into a finite number of triangles τ_j, $j = 1, ..., \#T$ satisfying

a) $\overline{\Omega} = \bigcup_{j=1}^{\#T} \tau_j$.

b) Each $\tau_j \in T$ is closed and $\tau_j \backslash \delta\tau_j \neq \emptyset$.

c) $(\tau_j \backslash \delta\tau_j) \cap (\tau_k \backslash \delta\tau_k) = \emptyset \quad \forall j \neq k$.

d) Each $\tau_j \in T$ is bounded Lipschitz continuous.

A triangulation T is called *conforming*, if it fulfills the additional condition

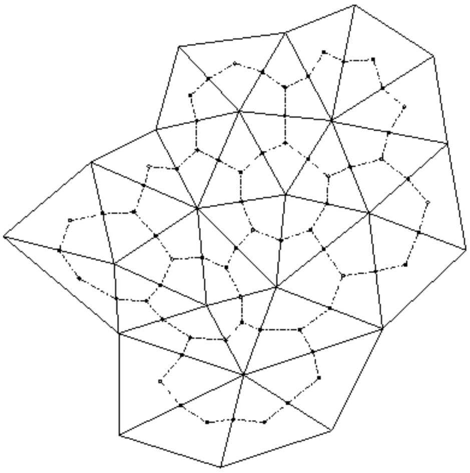

Figure 2.1: Primary and secondary grid.

e) Each edge of τ_j is either part of $\delta\Omega$ or an edge of exactly one other triangle $\tau_k \in T$, $k \neq j$.

T is also called the primary net or primary grid.

Due to structural and computational reasons it is preferable that the angles of the triangles τ_i are as near to 60° as possible. For each plane triangle it is obvious that, if one angle becomes larger, the sum of the other two will become smaller, as the sum of the angles in a plane triangle is always 180°. Thus, the larger the minimum of all $3\#T$ angles of a triangulation, the more regular the triangles $\tau_i \in T$ will become.

A Delaunay triangulation is a triangulation such that for each $\tau_i \in T$ the circumcircle of τ_i contains no vertex of another triangle $\tau_j \in T$, [Del34]. For a given set P of points in the plane, [BCK98] proved the following theorem.

Theorem 2.2. *Let P be a set of points in the plane. Any Delaunay triangulation of P maximizes the minimal angle over all triangulations of P.*

This feature comes in handy for the search of stencils in the computation of reconstruction polynomials as carried out in section 2.2.

The secondary grid Σ is derived from the primary net T by connecting the barycenter with all midpoints of the edges for each triangle as it is shown in figure 2.1. From this procedure, a set of polygonally bounded cells σ_i is obtained that make the secondary grid. Both grids are related by the fact that each point $\mathbf{x}_i$ that is a vertex of one or several triangles corresponds to one cell.

The edges that connect the cells σ_i and σ_j are denoted $l^k_{i,j}$, $k = 1, 2$ with normal vectors $\mathbf{n}^k_{i,j} \in \mathbb{R}^2$, $k = 1, 2$ as can be seen in figure 2.2. Definition

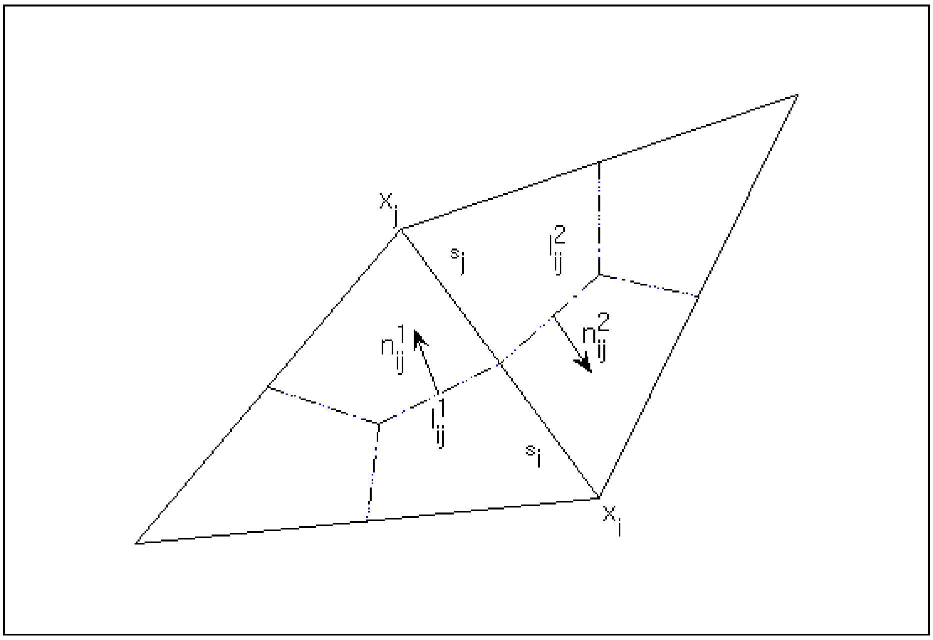

Figure 2.2: Denotation of the grid.

2.1.e) ensures, that for all edges of the primary net the median is uniquely determined for both adjacent triangles and thus the cell boundaries of the secondary grid are closed polygons.

The number of neighboring cells of a cell σ_i is denoted with non_i.

A measure for the fineness of the primary grid is the *grid spacing* h, that corresponds with the length of the edges of the triangulation. For the cells σ_i of the secondary grid it holds that $\sqrt{|\sigma_i|} = \mathcal{O}(h)$.

2.2 WENO Reconstruction

The aim of a polynomial reconstruction is to provide a piecewise polynomial approximation to the exact solution at a given time from the numerical solution which consists, in the case of finite volume schemes, of cell mean values.

The idea of simply taking the mean values of a number of cells equal to the number of coefficients of a polynomial of the desired degree and solving the resulting linear system of equations seems quite obvious at first. Problems arise from the question which cells to choose and the fact that, as Godunov showed in [God59], a reconstruction of high order produces spurious oscillations around discontinuities. This again leads to an unstable numerical scheme. The aim of the ENO/WENO scheme is to avoid or at least minimize these oscillations by computing a polynomial as smooth as possible from the given data, in which both schemes make slightly different approaches.

The ENO scheme was introduced for triangulations and general types of grids in [HaC91] and was further treated, for example, in [Abg94, HEO87, Son97]. The acronym ENO is short for essentially non-oscillatory. The idea of the ENO scheme is to choose the polynomial from a given set which is the smoothest with respect to a given oscillation indicator. This scheme suffers from the fact that the solution does not depend continuously on the data, as Friedrich showed in [Fri99] on the following simple example:

Consider three cells σ_0, σ_1, σ_2 of a one dimensional equidistant grid with grid spacing h. For the quantity $u(x)$, it holds that $\overline{u}_0 = 0$, $\overline{u}_1 = c > 0$ and $\overline{u}_2 = \varepsilon$. Sought-for is a linear reconstruction polynomial for $u(x)$ on σ_1. Obviously the slope of the reconstruction polynomial $p_0(x)$ computed from σ_0 and σ_1 is $\frac{c}{h}$, while the slope of the reconstruction polynomial $p_2(x)$ computed from σ_1 and σ_2 is $-\frac{c-\varepsilon}{h}$. For $\varepsilon > 0$, the oscillation indicator, that takes into account only the slope, will choose $p_2(x)$, for $\varepsilon < 0$ it will choose $p_0(x)$. Moreover, if $\varepsilon = 0$ the choice of one polynomial over the other will be at random.

The WENO or weighted ENO scheme, which can be considered as a generalization of the ENO scheme, avoids such problems by not simply choosing the smoothest polynomial but by returning a convex combination of all polynomials in which the portion of a basis polynomial depends proportional on its smoothness. The scheme was developed for the one dimensional case by Liu, Osher and Chan in [LOC94] and extended by Friedrich in [Fri98, Fri99] for the two dimensional case on unstructured grids.

In this work a WENO type reconstruction is used. The whole procedure described in the subsections 2.2.1 through 2.2.3 is called *reconstruction*. Its input is given by the net of cells Σ on Ω and their appropriate mean values and it results in one spatial reconstruction polynomial per cell σ_i and per quantity $u_k(\mathbf{x}, t)$, $k = 0, 1, 2,$. This polynomial preserves the cell mean value of u_k.

Definition 2.3. Let $\Pi_{deg}(\sigma_i; \mathbb{R})$ be the space of the polynomials $p: \sigma_i \to \mathbb{R}$, $\mathbf{x} \mapsto p(\mathbf{x})$ with $degree(p) \leq deg$ and u a map $u: \mathbb{R}^2 \times \mathbb{R}^+ \to \mathbb{R}$, $(\mathbf{x}, t) \mapsto u(\mathbf{x}, t) \in C^{deg}(\sigma_i; \mathbb{R})$.

The polynomial $p_i^n \in \Pi_{deg}$ is called a *reconstruction polynomial of degree deg for u on σ_i at time t^n*, if

$$\frac{1}{|\sigma_i|} \int_{\sigma_i} p_i^n(\mathbf{x}) d\mathbf{x} = \frac{1}{|\sigma_i|} \int_{\sigma_i} u(\mathbf{x}, t^n) d\mathbf{x} = \overline{u}_i^n \tag{2.2}$$

and

$$p_i^n(\mathbf{x}) - u(\mathbf{x}, t^n) = \mathcal{O}(h^{deg+1}) \ \forall \mathbf{x} \in \sigma_i$$

hold.

In this work, reconstruction polynomials p_i^n have the form

$$p_i^n(\mathbf{x}) = \sum_{i=0}^{deg} \sum_{|\boldsymbol{\alpha}|=i} a_{\boldsymbol{\alpha}} \left(\mathbf{x} - \mathbf{b}_i\right)^{\boldsymbol{\alpha}}, \tag{2.3}$$

as they are used to compute $\partial_{\boldsymbol{\alpha}} p_i^n(\mathbf{x})|_{\mathbf{x}=b_i}$, $|\boldsymbol{\alpha}| \leq deg$, in the scheme as approximation to $\partial_{\boldsymbol{\alpha}} u_k^n(\mathbf{x})|_{\mathbf{x}=b_i}$. Those derivatives can be evaluated very easily as $\boldsymbol{\alpha}! a_{\boldsymbol{\alpha}}$ in this form.

2.2.1 Stencil Search

The first step in finding a vector of appropriate smooth reconstruction polynomials $\mathbf{u}_i^n$ for a cell σ_i is to determine a set of time independent *stencils* $S_{i,j} \subset \Sigma$ that consist of cells σ_k and contain σ_i. On these stencils, the basic reconstruction polynomials $\mathbf{p}_{i,j}^n$ for each time step t^n are computed. The reconstruction polynomials $u_{i;k}^n$, $k = 1, ..., 4$, are then the weighted sums

$$u_{i;k}^n = \sum_{j=1}^{nos_i} \omega_{i,j;k}^n p_{i,j;k}^n$$

with weights $\omega_{i,j;k}^n$ satisfying $\sum_{j=1}^{nos_i} \omega_{i,j;k}^n = 1$ and $\omega_{i,j;k}^n \geq 0$.

There are many ways to select a number of cells from Σ to compute a reconstruction polynomial for a cell σ_i. It is thus necessary to define some requirements for the stencil, which restrict the number of possible stencils. These conditions should ensure that the selection of stencils has advantages for the numerical scheme.

The basic idea is to find a polynomial that has the same integral mean value on all cells of the stencil as the component u_l of $\mathbf{u}$ that is to be reconstructed. From the equations (2.2) and (2.3), it follows that for the reconstruction polynomial the following system of equations holds

$$\begin{aligned} \frac{1}{|\sigma_k|} \int_{\sigma_k} p_{i,j}^n(\mathbf{x}) d\mathbf{x} &= \frac{1}{|\sigma_k|} \sum_{|\boldsymbol{\alpha}|=0}^{deg} a_{\boldsymbol{\alpha}} \int_{\sigma_k} \left(\mathbf{x} - \mathbf{b}_i\right)^{\boldsymbol{\alpha}} \, d\mathbf{x} \\ &= \overline{u}_{k;l}^n \qquad \forall \sigma_k \in S_{i,j}. \end{aligned}$$

This can be written as $\mathbf{Ma} = \overline{\mathbf{u}}$, where $\mathbf{a}$ contains the coefficients of the polynomial, $\overline{\mathbf{u}}$ the mean values of u_l of the cells in $S_{i,j}$ and $\mathbf{M}$ the integral mean values of the monomials $(\mathbf{x}-\mathbf{b}_i)^{\boldsymbol{\alpha}}$ for the cells in $S_{i,j}$. Later in this section, the introduction of a scaling factor $\frac{1}{\sqrt{\sigma_i}^{\alpha}}$ for $\mathbf{M}$ is discussed.

First of all, a stencil should allow to compute a unique reconstruction polynomial of degree *deg* without the need to solve an under-determined linear system of equations.

Definition 2.4. Let $I_{i,j} \subset \{0, ..., \#\Sigma - 1\}$ be a set of indices with $i \in I_{i,j}$, $\#I_{i,j} \leq$ *stencil_size*. A stencil $S_{i,j} = \{\sigma_k | k \in I\} \subset \Sigma$ is called *admissible*, if the matrix

$$\mathbf{M} = (m_{k,l})_{\substack{k\in I_{i,j}\\ l=0,\ldots,noc-1}} = \left(\frac{1}{|\sigma_k|} \int_{\sigma_k} \left(\frac{\mathbf{x}-\mathbf{b}_i}{\sqrt{|\sigma_i|}} \right)^{\boldsymbol{\alpha}} d\mathbf{x} \right)_{\substack{k\in I_{i,j}\\ \boldsymbol{\alpha}=pos^{-1}(l)}} \tag{2.4}$$

has rank *noc*.

A condition for the numerical verification of the admissibility of a stencil is given in equation (2.14) on page 65.

Under the assumption that an admissible stencil $S_{i,j}$ contains σ_i and is connected, there are two extreme types: Stencils that are distributed as evenly as possible around their center σ_i and stencils that emanate from σ_i in one direction. The centered stencil has a lower diameter and the reconstructed polynomial will be based on a very compact set of cells. This is advantageous pursuing the aim of computing a reconstruction polynomial as smooth as possible, as Abgrall states in [Abg94].

Theorem 2.5. *Let S be an admissible stencil for degree deg, let $K(S) \subset \mathbb{R}^2$ be the convex hull of the union of the elements of S and let ν and ρ be the diameter of $K(S)$ and the supremum of the diameters of the circles contained in $K(S)$, respectively. Let $u \in C^{deg}(K(S); \mathbb{R})$ be a function whose derivative $D^{deg+1}u$ is bounded on $K(S)$ with*

$$M_{deg+1} = \sup \left\{ \left\| D^{deg+1} u(\mathbf{x}) \right\| |\mathbf{x} \in K(S) \right\} < +\infty.$$

If p is the polynomial of degree deg that has the same integral mean value as u for each element $\sigma_i \in S$, then for any integer m, $0 \leq m \leq deg$,

$$\sup \left\{ \| D^m u(x) - D^m P_u(x) \| \, |x \in K(S) \right\} \leq C M_{deg+1} \frac{\nu^{deg+1}}{\rho^m}$$

for some constant $C = C(m, deg, S)$.

This theorem expresses that if the function to be reconstructed is smooth in a neighborhood of σ_i, a stencil that is constructed as compact as possible

($\frac{\nu}{\rho}$ is rather small) will tend to return a better approximation polynomial than any other (more stretched) stencil.

If, on the other hand, σ_i is situated near a line of discontinuity, the resulting polynomial from a stencil that covers this discontinuity will oscillate. Abgrall even states in [Abg94] that the leading coefficients of this polynomial will tend to infinity as the mesh size tends to zero. In that case, if a discontinuity wave passes through the central stencil, directional stencils might then provide a search direction in which the function to be reconstructed is smooth, though they have a larger ratio $\frac{\nu}{\rho}$ and take into account information from more distant cells.

The implementation presented in this work contains both types of stencils. Each cell σ_i has directional stencils emanating in all directions that are proposed by the primary grid (approximately six) plus one central stencil.

2.2.1.1 Directional Stencils

The stencil search of the presented implementation makes use of the fact that all information about the structure of the grid is stored in the list of triangles. The algorithm runs through the list of triangles and computes three stencils for each triangle, one assigned to each vertex of the triangle and thus to the related cell of the secondary grid.

More precisely, for each triangle a system of neighborhoods is determined. The direct neighbors of the triangle itself are the first generation neighborhood, the neighbors of the n-th generation that are not already included in the system form the $(n+1)$-st generation.

The stencils, on the other hand, are defined via the vertexes of the triangle, as there is a one-to-one correspondence between the cells of the dual grid and the vertexes of the triangulation:

Definition 2.6. Let $\tau \in T$ be a triangle of the primary net with vertexes $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2$. A cell $\sigma_j \in \Sigma$ belongs to the stencil assigned to the vertex $\mathbf{x}_i$, $i = 0, 1, 2$ (and thus to the cell σ_i), if either $j \in \{0, 1, 2\}$, or if there exist $\alpha_1, \alpha_2 \geq 0$ such that the barycenter $\mathbf{b}_j$ of σ_j can be written in the form

$$\mathbf{b}_j = \mathbf{x}_i + \alpha_1(\mathbf{x}_{rem((i-1),3)} - \mathbf{x}_i) + \alpha_2(\mathbf{x}_{rem((i+1),3)} - \mathbf{x}_i), \tag{2.5}$$

with $rem(j, 3)$ being the remainder of the whole-number division of j by 3.

Equation (2.5) expresses the requirement that the barycenter $\mathbf{b}_j$ of σ_j is situated in the plane wedge defined by the angle

$$\angle(\mathbf{x}_{rem((i-1),3)},\ \mathbf{x}_i,\ \mathbf{x}_{rem((i+1),3)}).$$

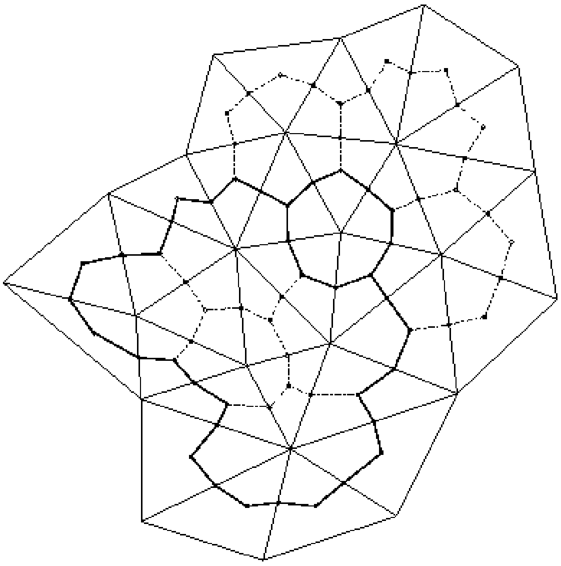

Figure 2.3: Directional stencil with six elements.

This can be easily verified by solving a linear 2 × 2 system of equations.

For each new generation of neighbors it is checked whether the cells not yet accounted for fulfill definition 2.6. Afterwards it is checked whether the stencils contain enough, that is *stencil_size*, elements. If this is true, the three stencils are assigned to the corresponding cells and the computation starts for the next triangle. Otherwise the next generation of neighbors is computed and checked.

The algorithm stops per default as soon as the 4*stencil_size*-th generation of neighbors is computed. Then, also stencils with an element number smaller than *stencil_size*, but at least *noc*, will be accepted.

This step wise approach abets the cells that are situated closer to σ_i being checked and maybe assigned to the stencil before farther cells are taken into account and thus that the stencil in the given frame staying as compact as possible.

As a consequence of the fact that the triangulation is of Delaunay-type, the stencils are generally broad enough to provide a regular matrix $\mathbf{M}$. In the case that $\mathbf{M}$ is nearly singular, and thus the stencil is not admissible, the relevant stencil is removed. As mentioned before, a feasible criterion to determine the admissibility of a stencil numerically is given in (2.14).

2.2.1.2 Central Stencils

When all directional stencils are computed and assigned to their relating cell, the last step of the stencil search is to compute a central stencil for each cell σ_i. In order to do this, the distance $\|\mathbf{b}_i - \mathbf{b}_k\|_2$ between $\mathbf{b}_i$ and the barycenters $\mathbf{b}_k$ of all cells $\sigma_k \in \bigcup_{j=1}^{nos_i-1} S_{i,j}$ in the union of the directional stencils $S_{i,j}$ belonging to σ_i and $\mathbf{b}_i$ is computed. The central stencil S_{i,nos_i} consists of the *stencil_size* cells σ_k with the smallest distance.

2.2.2 Computation of Basic Polynomials

To keep the notation simple and to avoid large amounts of indices, the notation $\sigma_i =: \sigma_0$ and

$$S := S_{0,j} := \left\{\sigma_0, \sigma_1, ..., \sigma_{\#S_{0,j}-1}\right\}$$

will always be used in this section. Furthermore, the index n denoting the time step will be neglected. The vector

$$\tilde{\mathbf{u}} := (\overline{u}_0, \overline{u}_1, ..., \overline{u}_{\#S-1})^T$$

contains the cell mean values of the function to be reconstructed. The index that specifies which component of $\overline{\mathbf{u}}_i$ is considered, is neglected for simplicity of notation, as the computation of the reconstruction polynomial is independent of the quantity for which it is carried out.

The basic idea to reconstruct the function u on σ_0 from its integral mean values on a set of cells S is to find a polynomial p, such that

$$\frac{1}{|\sigma_l|}\int_{\sigma_l} u(\mathbf{x})\ d\mathbf{x} = \overline{u}_l = \frac{1}{|\sigma_l|}\int_{\sigma_l} p(\mathbf{x})\ d\mathbf{x} \quad \forall \sigma_l \in S$$

holds. The use of a least squares method is to be preferred to using an exact method as mentioned above. To obtain a conservative scheme, it is then necessary to add a linear constraint to ensure that the cell mean value of σ_0 is preserved. According to the use of a least squares method, the number of cells in S was chosen larger than needed for an exact method. This reduces the probability to obtain a singular system matrix. Due to [DuK07], an appropriate number of cells for a reconstruction in two dimensions is $\left\lceil \frac{3noc}{2} \right\rceil$. As the derivatives of the reconstruction polynomial at the barycenter $\mathbf{b}_0$ of the cell σ_0 are needed later in the scheme, it is convenient to compute the coefficients of the polynomial expanded at $\mathbf{b}_0$ directly.

$$\mathbf{a} = (a_{0,0}, a_{1,0}, a_{0,1}, ...a_{1,deg-1}, a_{0,deg})^T = (a_{pos(\boldsymbol{\alpha})})_{0\leq|\alpha|\leq deg}$$

being the vector of coefficients of a polynomial $p(\mathbf{x})$,

$$f_l(\mathbf{a}) = \left(\sum_{|\boldsymbol{\alpha}| \leq deg} a_{\boldsymbol{\alpha}} \, |\sigma_l|^{-1} \int_{\sigma_l} (\mathbf{x} - \mathbf{b}_0)^{\boldsymbol{\alpha}} d\mathbf{x} - \overline{u}_l \right)^2 \quad \forall \sigma_l \in S$$

denotes the quadratic error of the integral mean value of p compared to u on a cell σ_l depending on $\mathbf{a}$. The total error for the stencil S is

$$F(\mathbf{a}) = \sum_{\sigma_l \in S} f_l(\mathbf{a}).$$

Finding $\mathbf{a} \in \mathbb{R}^{noc}$ such that $F(\mathbf{a})$ becomes minimal is equivalent to solving the problem

$$\mathbf{a} = \underset{\tilde{\mathbf{a}} \in \mathbb{R}^{noc}}{\operatorname{argmin}} \left\| \tilde{\mathbf{M}} \tilde{\mathbf{a}} - \tilde{\mathbf{u}} \right\|_2, \tag{2.6}$$

where $\tilde{\mathbf{M}}$ is the matrix whose rows contain the integral mean values of the monomials $(\mathbf{x} - \mathbf{b}_0)^{\boldsymbol{\alpha}}$ up to $|\boldsymbol{\alpha}| \leq \deg$ for all $\sigma_l \in S$.

As the entries of $\tilde{\mathbf{M}}$ consist of the integral mean values of the expanded monomials $(\mathbf{x} - \mathbf{b}_0)^{\boldsymbol{\alpha}}$ over the cells, the condition number of $\tilde{\mathbf{M}}$ depends on the refinement of the grid, and gets worse for finer grids. This again strongly influences the quality of the solution of the linear system of equations and thus also the quality of the reconstruction polynomials. There are different approaches to deal with this. In [Abg94], Abgrall proposes the use of a special polynomial basis to cancel out this effect.

The method used in this scheme was introduced by Friedrich in [Fri98] and consists of the multiplication with a scaling factor

$$s := \frac{1}{\sqrt{|\sigma_0|}}.$$

In the following theorem Friedrich states that the computation of the reconstruction polynomial in the representation

$$p(\mathbf{x}) =: \sum_{|\boldsymbol{\alpha}| \leq deg} s^{|\boldsymbol{\alpha}|} \tilde{a}_{\boldsymbol{\alpha}} (\mathbf{x} - \mathbf{b}_0)^{\alpha} \tag{2.7}$$

cancels out all scaling effects of the mesh width on the system matrix, at least for grids not too distorted.

Theorem 2.7. *Let $\mathbf{M}$ be the matrix of the scaled least squares problem, which is*

$$\mathbf{M} = (m_{k,l})_{\substack{k=0,\ldots,\#S-1 \\ l=0,\ldots,noc-1}} \quad \textit{with } m_{k,l} = \frac{1}{|\sigma_k|} \int_{\sigma_k} s^{|\boldsymbol{\alpha}|} (\mathbf{x} - \mathbf{b}_0)^{\boldsymbol{\alpha}} d\mathbf{x}, \ \boldsymbol{\alpha} := pos^{-1}(l).$$

Then $\mathbf{M}$ is invariant to grid scaling.

Due to equation (2.7), the coefficients of p can easily be computed by

$$a_{\boldsymbol{\alpha}} = s^{|\boldsymbol{\alpha}|}\tilde{a}_{\boldsymbol{\alpha}}. \tag{2.8}$$

The application of this scaling method and the linear constraint

$$\frac{1}{|\sigma_0|}\int_{\sigma_0} p(\mathbf{x})\ d\mathbf{x} = \sum_{|\boldsymbol{\alpha}|\leq deg} \tilde{a}_{\boldsymbol{\alpha}} \frac{1}{|\sigma_0|}\int_{\sigma_0} s^{|\boldsymbol{\alpha}|}(\mathbf{x}-\mathbf{b}_0)^{\boldsymbol{\alpha}}\ d\mathbf{x} = \overline{u}_0 \tag{2.9}$$

lead to the computation described in the following. As sub matrices and sub vectors are considered, the notation

$$\begin{aligned} m_{ij} &:= \frac{1}{|\sigma_i|}\int_{\sigma_i} s^{|\boldsymbol{\alpha}|}(\mathbf{x}-\mathbf{b}_0)^{|\boldsymbol{\alpha}|}d\mathbf{x},\ j := pos(\boldsymbol{\alpha}) \\ \mathbf{M}_{\substack{p,q\\r,s}} &:= (m_{ij})_{\substack{i=p,\dots,q\\j=r,\dots,s}} \\ \mathbf{M} &:= \mathbf{M}_{\substack{0,\#S-1\\0,noc-1}} \\ \tilde{\mathbf{a}}_{p,q} &:= (\tilde{a}_p, ..., \tilde{a}_q)^T \\ \tilde{\mathbf{u}}_{r,s} &:= (\overline{u}_r, ..., \overline{u}_s)^T \end{aligned}$$

is introduced.

Following the method in [LaH87, Bjö96], the constraint (2.9) can be used to transform the basic minimizing problem (2.6) into a new minimizing problem of a reduced dimension. The following steps need to be carried out:

a) Solving the constraint (2.9) for $\tilde{a}_0$:

$$\tilde{a}_0 = \frac{1}{m_{00}}\left(\overline{u}_0 - \sum_{j=1}^{noc-1}\tilde{a}_j m_{0j}\right) := \frac{1}{m_{00}}\left(\overline{u}_0 - \mathbf{M}_{\substack{0,0\\1,noc-1}}\tilde{\mathbf{a}}_{1,noc-1}\right).$$

As

$$m_{i0} = \frac{1}{|\sigma_i|}\int_{\sigma_i} s^{|\mathbf{0}|}(\mathbf{x}-\mathbf{b}_0)^{|\mathbf{0}|}d\mathbf{x} = 1 \text{ for } i = 0, ..., \#S-1,$$

holds, it follows that

$$\tilde{a}_0 = \overline{u}_0 - \mathbf{M}_{\substack{0,0\\1,noc-1}}\tilde{\mathbf{a}}_{1,noc-1}. \tag{2.10}$$

b) Inserting equation (2.10) into (2.6):

$$\begin{aligned} \|\mathbf{M}\tilde{\mathbf{a}} - \tilde{\mathbf{u}}\|_2 &\overset{(2.9)}{=} \left\|\mathbf{M}_{\substack{1,\#S-1\\0,noc}}\tilde{\mathbf{a}}_{0,noc-1} - \tilde{\mathbf{u}}_{1,\#S-1}\right\|_2 \\ &= \left\|\mathbf{M}_{\substack{1,\#S-1\\0,0}}\tilde{a}_0 + \mathbf{M}_{\substack{1,\#S-1\\1,noc-1}}\tilde{\mathbf{a}}_{1,noc-1} - \tilde{\mathbf{u}}_{1,\#S-1}\right\|_2 \end{aligned}$$

$$
\begin{aligned}
&\overset{(2.10)}{=} && \Big\| \mathbf{M}_{\substack{1,\#S-1\\0,0}} \left(\overline{u}_0 - \mathbf{M}_{\substack{0,0\\1,noc-1}} \tilde{\mathbf{a}}_{1,noc-1} \right) \\
& && + \mathbf{M}_{\substack{1,\#S-1\\1,noc-1}} \tilde{\mathbf{a}}_{1,noc-1} - \tilde{\mathbf{u}}_{1,\#S-1} \Big\|_2 \\
&= && \Big\| \left(\mathbf{M}_{\substack{1,\#S-1\\1,noc-1}} - \mathbf{M}_{\substack{1,\#S-1\\0,0}} \mathbf{M}_{\substack{0,0\\1,noc-1}} \right) \tilde{\mathbf{a}}_{1,noc-1} \\
& && - \left(\tilde{\mathbf{u}}_{1,\#S-1} - \mathbf{M}_{\substack{1,\#S-1\\0,0}} \overline{u}_0 \right) \Big\|_2 \\
&=: && \left\| \hat{\mathbf{M}} \tilde{\mathbf{a}}_{1,noc-1} - \hat{\mathbf{u}} \right\|_2
\end{aligned}
$$

c) Minimizing $\left\| \hat{\mathbf{M}} \tilde{\mathbf{a}}_{1,noc-1} - \hat{\mathbf{u}} \right\|_2$ with

$$
\hat{\mathbf{M}} := \mathbf{M}_{\substack{1,\#S-1\\1,noc-1}} - \mathbf{M}_{\substack{1,\#S-1\\0,0}} \mathbf{M}_{\substack{0,0\\1,noc-1}} \tag{2.11}
$$

$$
\hat{\mathbf{u}} := \tilde{\mathbf{u}}_{1,\#S-1} - \mathbf{M}_{\substack{1,\#S-1\\0,0}} \overline{u}_0 . \tag{2.12}
$$

The latter equation is an unconstrained least squares problem and can thus be solved with one of the usual algorithms.

Remark 2.8. *If the matrix* $\mathbf{M} \in \mathbb{R}^{\#S \times noc}$ *has full rank, the matrix* $\hat{\mathbf{M}} \in \mathbb{R}^{\#S-1 \times noc-1}$ *as defined in* (2.11) *has rank*$(\hat{\mathbf{M}}) = noc - 1$.

In this work, the resulting least squares problem

$$
\mathbf{a} = \underset{\tilde{\mathbf{a}} \in \mathbb{R}^{noc-1}}{\operatorname{argmin}} \left\| \hat{\mathbf{M}} \tilde{\mathbf{a}}_{1,noc-1} - \hat{\mathbf{u}} \right\|_2 \tag{2.13}
$$

is solved in the following straight forward way: Let $\mathbf{G} \in \mathbb{R}^{\#S-1 \times \#S-1}$ be a orthogonal matrix with

$$
\mathbf{G}\hat{\mathbf{M}} = \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix} ,
$$

where $\mathbf{R} \in \mathbb{R}^{noc-1 \times noc-1}$ is a regular upper triangular matrix. As the euclidean length of a vector, and thus the norm $\|\cdot\|_2$, is constant under multiplication with orthogonal matrices, the least squares problem (2.13) can be transformed into

$$
\left\| \hat{\mathbf{M}} \tilde{\mathbf{a}}_{1,noc-1} - \hat{\mathbf{u}} \right\|_2 = \left\| \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix} \tilde{\mathbf{a}}_{1,noc-1} - \mathbf{G}\hat{\mathbf{u}} \right\|_2 .
$$

Obviously, the linear system of equations $\mathbf{R}\tilde{\mathbf{a}}_{1,noc-1} = \left(\mathbf{G}\hat{\mathbf{u}}\right)_{1,noc-1}$ can be solved exactly as $\mathbf{R}$ is regular, while the occurring error of equation (2.13) is

$\left\|(\mathbf{G}\hat{\mathbf{u}})_{noc,\#S-1}\right\|_2$ and thus depends only on the initial quantities. Moreover it is possible at this point to check whether S is admissible: the condition number of $\mathbf{R}$ can be computed easily. Numerical tests up to a fourth order scheme have shown that a feasible criterion is

$$cond(\mathbf{R}) = \max_{i=0,\dots,noc-2}(r_{i,i}) \left(\min_{j=0,\dots,noc-2}(r_{j,j})\right)^{-1} \leq 10. \tag{2.14}$$

The restriction of a vector of length $\#S-1$ to the first $noc-1$ components ($\#S \geq noc$) can be realized by the multiplication with a matrix $\mathbf{P}$ that consists of the $(noc-1)\times(noc-1)$-identity matrix with $\#S-noc$ appended columns of zero entries to the right. Furthermore, the transformation from the vector $\tilde{\mathbf{u}} \in \mathbb{R}^{\#S}$, consisting of the cell mean values of the k-th conservative variable of the cells in stencil S, to the vector $\hat{\mathbf{u}} \in \mathbb{R}^{\#S-1}$ can be carried out by multiplication with the matrix $\mathbf{Q} \in \mathbb{R}^{\#S-1\times\#S}$, which consists of the $(\#S-1)\times(\#S-1)$-identity matrix with one appended column of -1 entries to the left. For the numerical scheme, the relations

$$\begin{aligned}
\mathbf{R}\tilde{\mathbf{a}}_{1,noc-1} &= \mathbf{PG}\hat{\mathbf{M}}\tilde{\mathbf{a}}_{1,noc-1} \\
&= (\mathbf{G}\hat{\mathbf{u}})_{1,noc-1} \\
&= \mathbf{PG}\hat{\mathbf{u}} \\
&= \mathbf{PGQ}\tilde{\mathbf{u}} \\
\Rightarrow \tilde{\mathbf{a}}_{1,noc-1} &= \mathbf{R}^{-1}\mathbf{PGQ}\tilde{\mathbf{u}} \\
&=: \tilde{\mathbf{D}}\tilde{\mathbf{u}}
\end{aligned}$$

result in a matrix $\tilde{\mathbf{D}} \in \mathbb{R}^{noc-1\times\#S}$ that depends only on the stencil S and not on the data in $\tilde{\mathbf{u}}$.

In the next step, $\tilde{a}_0$ can be computed by (2.10) such that the constraint (2.9) is fulfilled and the coefficients $\tilde{\mathbf{a}}$ can be re-scaled due to equation (2.8). These operations can again be carried out using a matrix multiplication: Let $\mathbf{e}_1 \in \mathbb{R}^{\#S}$ be the first unit vector and

$$\mathbf{D} := \begin{pmatrix} \mathbf{e}_1^T \\ \tilde{\mathbf{D}} \end{pmatrix} \in \mathbb{R}^{noc\times\#S},$$

then it follows that

$$\mathbf{D}\tilde{\mathbf{u}} = \begin{pmatrix} \overline{u}_0 \\ \tilde{\mathbf{a}}_{1,noc-1} \end{pmatrix}.$$

Due to equations (2.10) and (2.8) the computation of the coefficient $\tilde{a}_0 = a_0$ and the rescaling can be realized by multiplication with the matrix

$$\mathbf{S} := \begin{pmatrix} 1 & -m_{0,1} & -m_{0,2} & \cdots & -m_{0,noc-1} \\ 0 & s^{\left|pos^{-1}(1)\right|} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & s^{\left|pos^{-1}(noc-2)\right|} & 0 \\ 0 & \cdots & \cdots & 0 & s^{\left|pos^{-1}(noc-1)\right|} \end{pmatrix} \in \mathbb{R}^{noc \times noc}.$$

Finally, the matrix $\mathbf{SD}$, which only depends on the grid, is stored. The determination of the polynomial's coefficients can be carried out by

$$\mathbf{SD\tilde{u}} = \mathbf{S} \begin{pmatrix} \overline{u}_0 \\ \tilde{\mathbf{a}}_{1,noc-1} \end{pmatrix} = \mathbf{a}.$$

This least squares scheme presented for the computation of a polynomial $p(\mathbf{x})$ of degree deg for the function $u(\mathbf{x}, t^n)$ on the cell σ_0 with the constraint $\int_{\sigma_0} p(\mathbf{x}) d\mathbf{x} = \overline{u}_0$ indeed provides a reconstruction polynomial in the sense of definition 2.3, as can be seen in the following theorem, which is, to the authors knowledge, a new result for the coefficients of reconstruction polynomials obtained by a least squares method.

Theorem 2.9. *Let S be an admissible stencil for degree deg and $u(\cdot, t^n) \in C^{deg}(S)$. Then,*

$$a_{\boldsymbol{\alpha}} = \frac{1}{\boldsymbol{\alpha}!} \partial^{\boldsymbol{\alpha}} u(\mathbf{x}, t^n)|_{\mathbf{x}=\mathbf{b}_0} + \mathcal{O}(h^{deg+1-|\boldsymbol{\alpha}|}) \tag{2.15}$$

holds for the coefficients $a_{\boldsymbol{\alpha}}$ of the reconstruction polynomial p obtained by the algorithm above.

Proof. As $u(\cdot, t^n) \in C^{deg}(S)$ for the deg-th degree spatial Taylor polynomial of u at the time t^n, it holds that

$$\begin{aligned} u(\mathbf{x}, t^n) &= \sum_{i=0}^{deg} \sum_{|\boldsymbol{\alpha}|=i} \frac{(\mathbf{x} - \mathbf{b}_0)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \partial^{\boldsymbol{\alpha}} u(\mathbf{x}, t^n)|_{\mathbf{x}=\mathbf{b}_0} + \mathcal{O}(h^{deg+1}) \\ \Rightarrow \overline{u}_l &= \sum_{i=0}^{deg} \sum_{|\boldsymbol{\alpha}|=i} \frac{s^{|\boldsymbol{\alpha}|}}{|\sigma_l|} \int_{\sigma_l} (\mathbf{x} - \mathbf{b}_0)^{\boldsymbol{\alpha}} d\mathbf{x} \frac{1}{\boldsymbol{\alpha}! s^{|\boldsymbol{\alpha}|}} \partial^{\boldsymbol{\alpha}} u(\mathbf{x}, t^n)|_{\mathbf{x}=\mathbf{b}_0} \\ & \quad + \mathcal{O}(h^{deg+1}) \qquad \forall l = 0, ..., \#S - 1 \end{aligned}$$

$$\begin{aligned}
\Rightarrow \overline{u}_l - \overline{u}_0 &= \sum_{i=1}^{deg} \sum_{|\boldsymbol{\alpha}|=i} \left(\frac{s^{|\boldsymbol{\alpha}|}}{|\sigma_l|} \int_{\sigma_l} (\mathbf{x} - \mathbf{b}_0)^{\boldsymbol{\alpha}} d\mathbf{x} - \frac{s^{|\boldsymbol{\alpha}|}}{|\sigma_0|} \int_{\sigma_0} (\mathbf{x} - \mathbf{b}_0)^{\boldsymbol{\alpha}} d\mathbf{x} \right) \\
& \qquad \frac{1}{\boldsymbol{\alpha}! s^{|\boldsymbol{\alpha}|}} \partial^{\boldsymbol{\alpha}} u(\mathbf{x}, t^n)|_{\mathbf{x}=\mathbf{b}_0} + \mathcal{O}(h^{deg+1}) \\
& \forall l = 1, ..., \#S - 1 \\
\Rightarrow \hat{\mathbf{M}}\mathbf{c} &= \hat{\mathbf{u}} + \mathcal{O}(h^{deg+1})
\end{aligned}$$

with $\hat{\mathbf{M}}$ and $\hat{\mathbf{u}}$ as in equations (2.11) and (2.12) and $\mathbf{c} = (c_1, ...c_{noc-1})$ with

$$c_j = \frac{1}{\boldsymbol{\alpha}! s^{|\boldsymbol{\alpha}|}} \partial^{\boldsymbol{\alpha}} u(\mathbf{x}, t^n)|_{\mathbf{x}=\mathbf{b}_0}, \quad j = pos(\boldsymbol{\alpha}).$$

Hence $\left\| \hat{\mathbf{M}}\mathbf{c} - \hat{\mathbf{u}} \right\|_2 = \mathcal{O}(h^{deg+1})$ follows. On the other hand by the choice of $\tilde{\mathbf{a}}_{1,noc-1}$ as solution of the minimizing problem (2.6)

$$\left\| \hat{\mathbf{M}}\tilde{\mathbf{a}}_{1,noc-1} - \hat{\mathbf{u}} \right\|_2 \leq \left\| -\hat{\mathbf{M}}\mathbf{c} + \hat{\mathbf{u}} \right\|_2 = \mathcal{O}(h^{deg+1})$$

follows and thus

$$\begin{aligned}
\mathcal{O}(h^{deg+1}) &= \left\| \hat{\mathbf{M}}\tilde{\mathbf{a}}_{1,noc-1} - \hat{\mathbf{u}} \right\|_2 + \left\| -\hat{\mathbf{M}}\mathbf{c} + \hat{\mathbf{u}} \right\|_2 \\
&\geq \left\| \hat{\mathbf{M}}(\tilde{\mathbf{a}}_{1,noc-1} - \mathbf{c}) \right\|_2 \\
&= \left\| \mathbf{R}(\tilde{\mathbf{a}}_{1,noc-1} - \mathbf{c}) \right\|_2 .
\end{aligned}$$

The matrix $\mathbf{R}$ is regular, and because of theorem 2.7 $\mathbf{R}$ is also independent of h. Thus, $\|\mathbf{R}^{-1}\|_2 = \mathcal{O}(1)$ holds. With equation (2.16) it follows that

$$\begin{aligned}
\mathcal{O}(h^{deg+1}) &= \left\| \mathbf{R}^{-1} \right\|_2 \left\| \mathbf{R}(\tilde{\mathbf{a}}_{1,noc-1} - \mathbf{c}) \right\|_2 \\
&\geq \left\| \mathbf{R}^{-1}\mathbf{R}(\tilde{\mathbf{a}}_{1,noc-1} - \mathbf{c}) \right\|_2 \\
&= \left\| \tilde{\mathbf{a}}_{1,noc-1} - \mathbf{c} \right\|_2 .
\end{aligned}$$

Therefore, for each component of the vectors above

$$\tilde{a}_j - c_j = \mathcal{O}(h^{deg+1}) \tag{2.16}$$

holds. Using equation (2.8) for $|\boldsymbol{\alpha}| \geq 1$ and $\sqrt{|\sigma_0|} = \mathcal{O}(h)$ it follows that

$$\begin{aligned}
a_{\boldsymbol{\alpha}} &= s^{|\boldsymbol{\alpha}|} \tilde{a}_j \\
&\overset{(2.16)}{=} s^{|\boldsymbol{\alpha}|} \left(\frac{1}{\boldsymbol{\alpha}! s^{|\boldsymbol{\alpha}|}} \partial^{\boldsymbol{\alpha}} u(\mathbf{x}, t^n)|_{\mathbf{x}=\mathbf{b}_0} + \mathcal{O}(h^{deg+1}) \right)
\end{aligned}$$

$$
\begin{aligned}
&= \left(\frac{1}{\sqrt{|\sigma_0|}}\right)^{|\boldsymbol{\alpha}|}\left(\frac{1}{\boldsymbol{\alpha}!\left(\frac{1}{\sqrt{|\sigma_0|}}\right)^{|\boldsymbol{\alpha}|}}\partial^{\boldsymbol{\alpha}}u(\mathbf{x},t^n)|_{\mathbf{x}=\mathbf{b}_0}+\mathcal{O}(h^{deg+1})\right)\\
&= \frac{1}{\boldsymbol{\alpha}!}\partial^{\boldsymbol{\alpha}}u(\mathbf{x},t^n)|_{\mathbf{x}=\mathbf{b}_0}+\mathcal{O}(h^{deg+1-|\boldsymbol{\alpha}|}).
\end{aligned}
$$

Finally, for the coefficient $a_{(0,0)}$ the equation

$$
\begin{aligned}
a_{(0,0)} &= \overline{u}_0-\sum_{i=1}^{deg}\sum_{|\boldsymbol{\alpha}|=i}a_{\boldsymbol{\alpha}}\frac{1}{|\sigma_0|}\int_{\sigma_0}(\mathbf{x}-\mathbf{b}_0)^{\boldsymbol{\alpha}}\,d\mathbf{x}\\
&\overset{(2.16)}{=} \sum_{i=1}^{deg}\sum_{|\boldsymbol{\alpha}|=i}\left(\frac{1}{\boldsymbol{\alpha}!}\partial^{\boldsymbol{\alpha}}u(\mathbf{x},t^n)|_{\mathbf{x}=\mathbf{b}_0}-a_{\boldsymbol{\alpha}}\right)\frac{1}{|\sigma_0|}\int_{\sigma_0}(\mathbf{x}-\mathbf{b}_0)^{\boldsymbol{\alpha}}\,d\mathbf{x}\\
&\quad +u(\mathbf{b}_0,t^n)+\mathcal{O}(h^{deg+1})\\
&= \sum_{i=1}^{deg}\sum_{|\boldsymbol{\alpha}|=i}\mathcal{O}(h^{deg+1-|\boldsymbol{\alpha}|})\frac{1}{|\sigma_0|}\int_{\sigma_0}(\underbrace{\mathbf{x}-\mathbf{b}_0}_{=\mathcal{O}(h)})^{\boldsymbol{\alpha}}d\mathbf{x}\\
&\quad +u(\mathbf{b}_0,t^n)+\mathcal{O}(h^{deg+1})\\
&= u(\mathbf{b}_0,t^n)+\mathcal{O}(h^{deg+1})
\end{aligned}
$$

can be obtained, which completes the proof. □

Using theorem 2.9, it is now possible to proof the approximation of the reconstruction polynomial.

Lemma 2.10. *The equation*

$$p_0^n(\mathbf{x})-u(\mathbf{x},t^n)=\mathcal{O}(h^{deg+1})\ \forall\mathbf{x}\in\sigma_0$$

holds for the polynomial $p_0^n(\mathbf{x})$ of degree deg obtained by the scheme above for the function $u(\mathbf{x},t^n)$ on the cell σ_0.

Proof. This follows easily from

$$
\begin{aligned}
p_0^n(\mathbf{x})-u(\mathbf{x},t^n) &= \sum_{i=0}^{deg}\sum_{|\boldsymbol{\alpha}|=i}(a_{\boldsymbol{\alpha}}-\frac{1}{\boldsymbol{\alpha}!}\partial^{\boldsymbol{\alpha}}u|_{\mathbf{x}=\mathbf{b}_0})(\mathbf{x}-\mathbf{b}_0)^{\boldsymbol{\alpha}}+\mathcal{O}(h^{deg+1})\\
&= \sum_{i=0}^{deg}\sum_{|\boldsymbol{\alpha}|=i}\mathcal{O}(h^{deg+1-|\boldsymbol{\alpha}|})(\underbrace{\mathbf{x}-\mathbf{b}_0}_{=\mathcal{O}(h)})^{\boldsymbol{\alpha}}+\mathcal{O}(h^{deg+1})\\
&= \mathcal{O}(h^{deg+1}).
\end{aligned}
$$

□

Thus, following definition 2.3, the polynomial $p(\mathbf{x})$ is indeed a reconstruction polynomial of the desired degree.

2.2.3 Weighting

The last and eponymous step is the weighting of the polynomials in order to obtain a reconstruction polynomial that is as ‘smooth‘ as possible. There are two ways to do so: Either by analyzing the mean values of the cells in the corresponding stencil, or by analyzing the coefficients of the computed reconstruction polynomial. The authors of [LOC94] for example proposed a method for the one dimensional case based on the computation of a table of differences of mean values. In [JiS96], a weighting function based on the squares of the derivatives of the reconstruction polynomials was presented. However, the direct use of the reconstruction polynomial to obtain a measure of the smoothness of the reconstruction is the method that is widely followed nowadays.

In order to compute a reconstruction polynomial as the weighted sum of some basic polynomials for a cell σ_i, a weight $\omega^n_{i,j;k}$ for each polynomial $p^n_{i,j;k}$ is determined via

$$\omega^n_{i,j;k} = \frac{s(i,j)\left(\varepsilon + OI(p^n_{i,j;k})\right)^{-r}}{\sum_{l=1}^{nos_i} s(i,l)\left(\varepsilon + OI(p^n_{i,l;k})\right)^{-r}}. \tag{2.17}$$

The crucial step in equation (2.17) is the computation of the positive *oscillation indicator* $OI(p^n_{i,j;k})$, that gives a measure of the oscillations of a given polynomial $p^n_{i,j;k}$. To avoid division by zero, a very small number $\varepsilon > 0$ is added to the oscillation indicator. The exponent r is a positive integer that controls the sensitivity of the weights to oscillations. For large values of r, the weight of the smoothest polynomial tends to 1, the weights of the other polynomials to 0 and the whole scheme thus to an ENO scheme. The function $s(i,j)$ is positive and stencil dependent and determines the general influence of the stencil $S_{i,j}$, based on its shape. In general, $s(i,j)$ shall emphasize stencils with small diameter due to theorem 2.5.

Different authors recommend different sets of these constants and functions respectively. For example, Friedrich recommends the set

$$\begin{aligned} r &= 8 \\ \varepsilon &= 10^{-16} \\ OI(p) &= \left(\sum_{|\boldsymbol{\alpha}|=1} \int_{\sigma_i} \left(D^{\boldsymbol{\alpha}} p(\mathbf{x})\right)^2 \, d\mathbf{x} \right)^{\frac{1}{2}} \end{aligned}$$

$$s(i,j) = \begin{cases} 12 & S_{i,j} \text{ is central stencil} \\ 1 & \text{otherwise} \end{cases}$$

in [Fri99], while the authors in [KäI04] used

$$\begin{aligned} r &= 4 \\ \varepsilon &= 10^{-5} \\ OI(p) &= \sum_{1\leq|\boldsymbol{\alpha}|\leq deg} \int_{\sigma_i} |\sigma_i|^{|\boldsymbol{\alpha}|-1} \left(D^{\boldsymbol{\alpha}} p(\mathbf{x})\right)^2 \; d\mathbf{x} \\ s(i,j) &= 1. \end{aligned}$$

The authors in [DuK07] recommend

$$\begin{aligned} r &= 4 \\ \varepsilon &= 10^{-5} \\ OI(p) &= \sum_{1\leq|\boldsymbol{\alpha}|\leq deg} \int_{\sigma_i} \left(D^{\boldsymbol{\alpha}} p(\mathbf{x})\right)^2 \; d\mathbf{x} \\ s(i,j) &= \begin{cases} 10^3 & S_{i,j} \text{ is central stencil} \\ 1 & \text{otherwise} \end{cases} . \end{aligned} \tag{2.18}$$

Numerical tests with different sets of these constants and functions have shown that the scheme is most stable when using the third set. In this set the polynomial returned by the central stencil, that can be expected to have the smallest diameter, gets an accentuation in the weighting, which is in accordance to theorem 2.5.

2.2.4 Characteristic Variables

In this work, the reconstruction is carried out in characteristic variables, as is recommended in [DKT07]. Numerical test have shown that this indeed has a large influence on the stability of the numerical solution that totally justifies the extra computational costs.

Characteristic variables are the variables obtained when diagonalizing the x_1-split 2D shallow water equations. This set of variables can be computed only for a fixed time and in relation to a given direction. The time in this case is t^n, while the directions are given normal to the piecewise polygonal boundary of σ_i.

To be more precise, the matrices of the coefficients of the reconstruction polynomials in conservative variables $\mathbf{A}_j \in \mathbb{R}^{noc \times 3}$ are computed and stored for each stencil $S_{i,j}$. Then the conservative reconstruction polynomial

$$_0p^n_{i;k} := \sum_j \omega^n_{i,j;k} p^n_{i,j;k} \tag{2.19}$$

is computed. The 0 to the left indicates that weights for this polynomial were computed in the conservative, and not in a characteristic, formulation.

Afterwards, the coefficients of the reconstruction polynomials in characteristic variables for the x_1-split equations (1.24) with respect to each edge $\mathbf{l}^a_{i,m}$, $a = 1, 2$, of the boundary between σ_i and its neighboring cells σ_m are computed from $\mathbf{A}_j$. Referring to the edge $\mathbf{l}^a_{i,m}$, these reconstruction polynomials in characteristic variables are denoted $_{m,a}\tilde{p}^n_{i,j;k}$. The transformation is carried out by the multiplication with matrices $\mathbf{R}^{\mathbf{n}^a_{i,m}} \in \mathbb{R}^{3\times 3}$, where $\mathbf{n}^a_{i,m}$ is the normal vector to $\mathbf{l}^a_{i,m}$, $a = 1, 2$. These transformation matrices have the form

$$\mathbf{R}^{\mathbf{n}^a_{i,m}} = -\frac{1}{2\sqrt{\hat{u}_0}} \begin{pmatrix} -\frac{\hat{u}_1}{\hat{u}_0} - \sqrt{\hat{u}_0} & 1 & 0 \\ 2\sqrt{\hat{u}_0}\frac{\hat{u}_2}{\hat{u}_0} & 0 & -2\sqrt{\hat{u}_0} \\ \frac{\hat{u}_1}{\hat{u}_0} - \sqrt{\hat{u}_0} & -1 & 0 \end{pmatrix}$$

with $\hat{\mathbf{u}} = \frac{1}{2}\mathbf{T}(\mathbf{n}^a_{i,m})\,(\overline{\mathbf{u}}_i + \overline{\mathbf{u}}_m)$ being the arithmetic mean value of the rotated vectors of conservative variables of both adjacent cells.

The matrices $\mathbf{R}^{\mathbf{n}^a_{i,m}}$ is the inverse of the matrix of the right eigenvectors $\mathbf{r}_i(\hat{\mathbf{u}})$ of the Jacobian matrix $f_1'(\hat{\mathbf{u}})$ of the x_1-split 2D shallow water equations (1.24), rotated with respect to the edge $\mathbf{l}^a_{i,m}$.

This method is more effective than the canonic approach, which is the computation of the characteristic reconstruction polynomials out of the set of cell mean values in characteristic variables. But both methods lead to the same polynomial, as can be seen easily in the following:

Let $\overline{\mathbf{U}} \in \mathbb{R}^{\#S_{i,j}\times 3}$ be the matrix that contains the vectors of the cell mean values in conservative variables of all cells $\sigma_h \in S_{i,j}$, $\mathbf{A}_j \in \mathbb{R}^{noc\times 3}$ the matrix that consists of the vectors of the coefficients of the three reconstruction polynomials $p^n_{i,j;k}$, $k = 0, 1, 2$ and $\mathbf{SD} \in \mathbb{R}^{noc\times \#S_{i,j}}$ the reconstruction matrix, such that $\mathbf{SD}\overline{\mathbf{U}} = \mathbf{A}_j$. Then,

$$\begin{aligned} \mathbf{R}^{\mathbf{n}^a_{i,m}}\mathbf{T}(\mathbf{n}^a_{i,j})\mathbf{A}_j^T &= \mathbf{R}^{\mathbf{n}^a_{i,m}}\mathbf{T}(\mathbf{n}^a_{i,j})\left(\mathbf{SD}\overline{\mathbf{U}}\right)^T \\ &= \mathbf{R}^{\mathbf{n}^a_{i,m}}\mathbf{T}(\mathbf{n}^a_{i,j})\overline{\mathbf{U}}^T(\mathbf{SD})^T \\ &= \mathbf{SD}\left(\mathbf{R}^{\mathbf{n}^a_{i,m}}\mathbf{T}(\mathbf{n}^k_{i,j})\overline{\mathbf{U}}^T\right)^T . \end{aligned}$$

For each set of characteristic coefficients, a weighted polynomial

$$_{m,a}p^n_{i;k} := \sum_j {}_{m,a}\omega^n_{i,j;k} p^n_{i,j;k} \tag{2.20}$$

is determined using the weighting recommended in [DuK07] as mentioned above.

In this sum, the polynomials in conservative variables are added with the weights being computed due to the polynomials in characteristic variables. This was done in order to save the rotation back as a source of rounding errors.

From the set of $2non_i + 1$ polynomials

$$_0p^n_{i;k} \cup \left\{ _{m,a}p^n_{i;k} | m = 1, ..., non_i, a = 1, 2 \right\}$$

obtained in the process, the polynomial $u^n_{i;k}$ with

$$OI(u^n_{i;k}) = \min(OI(_0p^n_{i;k}), \min_{m,a}(OI(_{m,a}p^n_{i;k})))$$

is chosen as reconstruction polynomial for the cell σ_i at time t^n.

2.3 Space Time Expansion

The Space Time Expansion scheme, or shorter STE scheme, is a generalization of the Lax-Wendroff scheme presented in [LaW60]. It is a predictor-corrector scheme. The predictor consists of one polynomial $U^n_{i;k}(\mathbf{x}, t)$ in space and time per cell and function. These polynomials are utilized to predict the values of the functions $\mathbf{u}$ at the integration points. In this prediction, the influence of the neighboring cells on the evolution of the cell σ_i is neglected. In the corrector step, the fluxes over the cell boundaries are computed via a Riemann solver, as described in section 2.4. This idea was first presented in [HEO87] and then adopted in [GLM07, GLM08].

The fact that one polynomial per function will be computed for the whole cell and that the Riemann problems at the integration points are solved as the last step in flux computation, distinguishes this scheme from the ADER approach. The abbreviation ADER stands for Arbitrary high order using DERivatives and was invented by Titarev and Toro in [TiT02]. In their scheme, one polynomial in time is computed for each spatial integration point per function. The Riemann problems are solved in the process of expanding the polynomial and the fluxes are computed by exactly integrating these polynomials in time.

The computation of the polynomials $\mathbf{U}^n_i(\mathbf{x}, t)$ is based on a Taylor expansion $\mathbf{T}^n_i$ of the 2D shallow water equations with center $(\mathbf{b_i}, t^n)$,

$$\begin{aligned} T^n_{i;k}(\mathbf{x}, t) = &\sum_{0=|\boldsymbol{\alpha}|}^{deg} \frac{[(\mathbf{x}, t) - (\mathbf{b}_i, t^n)]^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \left. \frac{\partial^{|\boldsymbol{\alpha}|} u_k(\mathbf{x}, t)}{\partial(\mathbf{x}, t)^{\boldsymbol{\alpha}}} \right|_{(\mathbf{b_i}, t^n)} \\ &+ \mathcal{O}(h^{deg+1}). \end{aligned} \tag{2.21}$$

The space-time polynomial $\mathbf{U}_i^n(\mathbf{x}, t)$ has the form

$$U_{i;k}^n(\mathbf{x}, t) = \sum_{0=|\boldsymbol{\alpha}|}^{deg} \left[(\mathbf{x}, t) - (\mathbf{b}_i, t^n)\right]^{\boldsymbol{\alpha}} A_{\boldsymbol{\alpha};k}, \tag{2.22}$$

where $A_{\boldsymbol{\alpha};k}$ is an approximation to $\frac{1}{\boldsymbol{\alpha}!}\partial^{\boldsymbol{\alpha}} u_k|_{(\mathbf{b}_i, t^n)}$. In order to obtain this approximation, the function u_k itself and its spatial derivatives are replaced by $u_{i;k}^n(\mathbf{x})$ and its derivatives, while for the summands containing time derivatives the Cauchy-Kovalewskaja Procedure will be applied.

2.3.1 Cauchy-Kovalewskaja Procedure

The Cauchy-Kovalewskaja Procedure, or short CKP, also known as Lax-Wendroff Procedure, was presented in [LaW60]. It provides approximations to derivatives from $u_k(\mathbf{x}, t)$ with respect to time, as they are needed in (2.22).

To keep it as short and clear as possible, the notation

$$f_{\boldsymbol{\alpha}} = f_{(\alpha_1, \alpha_2, \alpha_3)} = \partial^{\boldsymbol{\alpha}} f(\mathbf{x}, t)$$

will be used.

If the functions $\mathbf{u}(\mathbf{x}, t)$ are assumed to be smooth enough to change the order of partial derivatives, the shallow water equations (1.20) themselves are used to obtain the time derivatives: Knowing the spatial derivatives of the solution $\mathbf{u}$, the differential equation

$$\mathbf{u}_{(0,0,1)} = -\mathbf{f}_1(\mathbf{u})_{(1,0,0)} - \mathbf{f}_2(\mathbf{u})_{(0,1,0)} + \mathbf{g}(\mathbf{u}) \tag{2.23}$$

provides a rule for their calculation.

Under the condition that the higher spatial derivatives of $\mathbf{u}$ are known, higher time and time-space derivatives can be computed by differentiating equation (2.23) successively. For example, using equation (2.23) and the second order spatial derivatives $\mathbf{u}_{(2,0,0)}$, $\mathbf{u}_{(1,1,0)}$ and $\mathbf{u}_{(0,2,0)}$, the second order time and mixed derivatives

$$\begin{aligned}
\mathbf{u}_{(1,0,1)} &= -\mathbf{f}_1(\mathbf{u})_{(2,0,0)} - \mathbf{f}_2(\mathbf{u})_{(1,1,0)} + \mathbf{g}(\mathbf{u})_{(1,0,0)} \\
\mathbf{u}_{(0,1,1)} &= -\mathbf{f}_1(\mathbf{u})_{(1,1,0)} - \mathbf{f}_2(\mathbf{u})_{(0,2,0)} + \mathbf{g}(\mathbf{u})_{(0,1,0)} \\
\mathbf{u}_{(0,0,2)} &= -\mathbf{f}_1(\mathbf{u})_{(1,0,1)} - \mathbf{f}_2(\mathbf{u})_{(0,1,1)} + \mathbf{g}(\mathbf{u})_{(0,0,1)}
\end{aligned}$$

can be computed easily.

In this section, only the case of a flat topography, that is $\mathbf{g}(\mathbf{u}) \equiv 0$, will be discussed. The case of a non-flat topography is covered separately in section 2.5.

Although the solution $\mathbf{u}$ is not known, naturally, the reconstruction polynomials $\mathbf{u}_i^n$ provide an approximation for $\mathbf{u}$ and its spatial derivatives $\mathbf{u}_{(\boldsymbol{\alpha})}$ with $|\boldsymbol{\alpha}| \leq deg$, following theorem 2.9.

In this context, the derivatives $\mathbf{f}_i(\mathbf{u})_{\boldsymbol{\alpha}},\ i = 1, 2$, require differentiating products and quotients of functions, such as $f_{1;2} = f_{2;1} = \frac{u_1 u_2}{u_0} = (\Phi v_1)v_2$. In the case of differentiating a product of functions, the n-dimensional Leibniz rule for two functions $F(\mathbf{x}, t), G(\mathbf{x}, t)$

$$\begin{aligned}(FG)_{\boldsymbol{\alpha}} =&\frac{\partial^{|\boldsymbol{\alpha}|}}{\partial x_1^{\alpha_1} x_2^{\alpha_2} t^{\alpha_3}}(FG)(\mathbf{x}, t)\\ =&\sum_{i=0}^{\alpha_1}\sum_{j=0}^{\alpha_2}\sum_{k=0}^{\alpha_3}\binom{\alpha_1}{i}\binom{\alpha_2}{j}\binom{\alpha_3}{k}F_{(i,j,k)}G_{(\alpha_1-i,\alpha_2-j,\alpha_3-k)} \qquad (2.24)\end{aligned}$$

for $n = 3$ can be applied.

In the case of differentiating a quotient, [Dys01] provides a rule for the computation of the $\boldsymbol{\alpha}$-th partial derivative of a function $F(\mathbf{x}, t) = \frac{FG}{G}(\mathbf{x}, t)$ if the partial derivatives of $F(\mathbf{x}, t)$ and $G(\mathbf{x}, t)$ up to degree $|\boldsymbol{\alpha}| - 1$ and the $\boldsymbol{\alpha}$-th partial derivative of $(FG)(\mathbf{x}, \mathbf{t})$ are known. A slight modification of this rule, concerning the summation, for computational speedup is used in this work:

$$\begin{aligned}F_{\boldsymbol{\alpha}} = \Bigg[&(FG)_{\boldsymbol{\alpha}} - \sum_{i=0}^{\alpha_1}\sum_{j=0}^{\alpha_2}\sum_{k=0}^{\alpha_3-1}\binom{\alpha_1}{i}\binom{\alpha_2}{j}\binom{\alpha_3}{k}F_{(i,j,k)}G_{(\alpha_1-i,\alpha_2-j,\alpha_3-k)}\\ &-\sum_{i=0}^{\alpha_1}\sum_{j=0}^{\alpha_2-1}\binom{\alpha_1}{i}\binom{\alpha_2}{j}F_{(i,j,\alpha_3)}G_{(\alpha_1-i,\alpha_2-j,0)}\\ &-\sum_{i=0}^{\alpha_1-1}\binom{\alpha_1}{i}F_{(i,\alpha_2,\alpha_3)}G_{(\alpha_1-i,0,0)}\Bigg]\frac{1}{G_{(0,0,0)}}.\end{aligned} \qquad (2.25)$$

In the following, a proposition concerning the approximation order of the derivatives obtained by using this rule with approximations of the derivatives, like they are guaranteed by theorem 2.9, is made.

Theorem 2.11. *Assume, there exist approximations $f_{\boldsymbol{\gamma}}$ and $g_{\boldsymbol{\gamma}}$ for the $\boldsymbol{\gamma}$-th partial derivatives of F and G respectively with*

$$\begin{aligned}F_{\boldsymbol{\gamma}}|_{(\mathbf{b}_i,t^n)} &= \boldsymbol{\gamma}! f_{\boldsymbol{\gamma}} + \mathcal{O}(h^{deg+1-|\boldsymbol{\gamma}|})\\ G_{\boldsymbol{\gamma}}|_{(\mathbf{b}_i,t^n)} &= \boldsymbol{\gamma}! g_{\boldsymbol{\gamma}} + \mathcal{O}(h^{deg+1-|\boldsymbol{\gamma}|}) \qquad (2.26)\end{aligned}$$

for $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \gamma_3)$ *with* $|\boldsymbol{\gamma}| < |\boldsymbol{\beta}|$ *and* $\gamma_l \leq \beta_l$, $l = 1, 2, 3$. *In addition, assume that the approximation* $(fg)_{\boldsymbol{\beta}}$ *of the* $\boldsymbol{\beta}$*-th partial derivative of* FG *with*

$$(FG)_{\boldsymbol{\beta}}|_{(\mathbf{b}_i,t^n)} = \boldsymbol{\beta}!(fg)_{\boldsymbol{\beta}} + \mathcal{O}(h^{deg+1-|\boldsymbol{\beta}|}) \qquad (2.27)$$

exists.

Then,

$$F_{\boldsymbol{\beta}}|_{(\mathbf{b}_i,t^n)} - \boldsymbol{\beta}!f_{\boldsymbol{\beta}} = \mathcal{O}\left(h^{deg+1-|\boldsymbol{\beta}|}\right)$$

holds for $\boldsymbol{\beta}!f_{\boldsymbol{\beta}}$ *with* $|\boldsymbol{\beta}| \leq deg$.

Proof. In the following, the writing of the point of the evaluation $(\mathbf{b}_i, t^n)$ for the functions F, G, FG will be neglected. Using equations (2.25), (2.26), $|\boldsymbol{\gamma}| < |\boldsymbol{\beta}|$ and

$$\binom{\beta_1}{i}\binom{\beta_2}{j}\binom{\beta_3}{k}(\beta_1 - i)!(\beta_2 - j)!(\beta_3 - k)!i!j!k! = \boldsymbol{\beta}!,$$

for the difference $F_{\boldsymbol{\beta}} - \boldsymbol{\beta}!f_{\boldsymbol{\beta}}$ the following holds:

$$\begin{aligned}
F_{\boldsymbol{\beta}} - \boldsymbol{\beta}!f_{\boldsymbol{\beta}} =& \frac{(FG)_{\boldsymbol{\beta}}}{G_{(0,0,0)}} \\
&- \Bigg[\sum_{i=0}^{\beta_1}\sum_{j=0}^{\beta_2}\sum_{k=0}^{\beta_3-1}\binom{\beta_1}{i}\binom{\beta_2}{j}\binom{\beta_3}{k} G_{(\beta_1-i,\beta_2-j,\beta_3-k)}F_{(i,j,k)} \\
&\quad + \sum_{i=0}^{\beta_1}\sum_{j=0}^{\beta_2-1}\binom{\beta_1}{i}\binom{\beta_2}{j} G_{(\beta_1-i,\beta_2-j,0)}F_{(i,j,\beta_3)} \\
&\quad + \sum_{i=0}^{\beta_1-1}\binom{\beta_1}{i} G_{(\beta_1-i,0,0)}F_{(i,\beta_2,\beta_3)}\Bigg]\frac{1}{G_{(0,0,0)}} \\
&- \frac{\boldsymbol{\beta}!(fg)_{\boldsymbol{\beta}}}{g_{(0,0,0)}} \\
&+ \boldsymbol{\beta}!\Bigg[\sum_{i=0}^{\beta_1}\sum_{j=0}^{\beta_2}\sum_{k=0}^{\beta_3-1} g_{(\beta_1-i,\beta_2-j,\beta_3-k)}f_{(i,j,k)} \\
&\quad + \sum_{i=0}^{\beta_1}\sum_{j=0}^{\beta_2-1} g_{(\beta_1-i,\beta_2-j,0)}f_{(i,j,\beta_3)} \\
&\quad + \sum_{i=0}^{\beta_1-1} g_{(\beta_1-i,0,0)}f_{(i,\beta_2,\beta_3)}\Bigg]\frac{1}{g_{(0,0,0)}} \\
\overset{(2.26),(2.27)}{=}& \frac{1}{g^2_{(0,0,0)} + \mathcal{O}\left(h^{deg+1}\right)}\Bigg[\mathcal{O}\left(h^{deg+1}\right) - \mathcal{O}\left(h^{deg+1-|\boldsymbol{\beta}|}\right)
\end{aligned}$$

$$\begin{aligned}
&-\sum_{i=0}^{\beta_1}\sum_{j=0}^{\beta_2}\sum_{k=0}^{\beta_3-1}\mathcal{O}\left(h^{deg+1-(|\boldsymbol{\beta}|-i-j-k)}\right)+\mathcal{O}\left(h^{deg+1-(i+j+k)}\right)\\
&-\sum_{i=0}^{\beta_1}\sum_{j=0}^{\beta_2-1}\mathcal{O}\left(h^{deg+1-(|\boldsymbol{\beta}|-i-j-\beta_3)}\right)+\mathcal{O}\left(h^{deg+1-(i+j+\beta_3)}\right)\\
&-\sum_{i=0}^{\beta_1-1}\mathcal{O}\left(h^{deg+1-(|\boldsymbol{\beta}|-i-\beta_2-\beta_3)}\right)+\mathcal{O}\left(h^{deg+1-(i+\beta_2+\beta_3)}\right)\Bigg]\\
=&\mathcal{O}\left(h^{deg+1-|\boldsymbol{\beta}|}\right)
\end{aligned}$$

□

Remark 2.12. *If $f_{\boldsymbol{\gamma}}$ and $g_{\boldsymbol{\gamma}}$ exist with*

$$\begin{aligned}
F_{\boldsymbol{\gamma}}|_{(\mathbf{b}_i,t^n)}-f_{\boldsymbol{\gamma}}&=\mathcal{O}\left(h^{deg+1-|\boldsymbol{\gamma}|}\right)\\
G_{\boldsymbol{\gamma}}|_{(\mathbf{b}_i,t^n)}-g_{\boldsymbol{\gamma}}&=\mathcal{O}\left(h^{deg+1-|\boldsymbol{\gamma}|}\right)
\end{aligned}$$

for $\gamma_i\le\beta_i$, $i=1,2,3$, inserting these approximations into equation (2.24) *gives an approximation $(fg)_{\boldsymbol{\beta}}$ to $(FG)_{\boldsymbol{\beta}}|_{(\mathbf{b}_i,t^n)}$ with*

$$(FG)_{\boldsymbol{\beta}}|_{(\mathbf{b}_i,t^n)}-(fg)_{\boldsymbol{\beta}}=\mathcal{O}\left(h^{deg+1-|\boldsymbol{\beta}|}\right). \tag{2.28}$$

Given the appropriate derivatives, the computation of $A_{\boldsymbol{\alpha};0}$ as approximation to $\frac{1}{\boldsymbol{\alpha}!}u_{\boldsymbol{\alpha};0}$ as

$$A_{\boldsymbol{\alpha};0}=A_{(\alpha_1+1,\alpha_2,\alpha_3-1);1}+A_{(\alpha_1,\alpha_2+1,\alpha_3-1);2}$$

does not pose any problem, but the equation for the computation of $A_{\boldsymbol{\alpha};i}$, $i=1,2$ contains derivatives of the terms

$$\frac{u_1u_1}{u_0}=(\Phi v_1)v_1,\quad \frac{u_1u_2}{u_0}=(\Phi v_1)v_2,\quad \frac{u_2u_2}{u_0}=(\Phi v_2)v_2.$$

To cope with this problem, auxiliary functions

$$g_1(\mathbf{x},t)=\sum_{|\alpha|=0}^{deg-1}g_{\boldsymbol{\alpha};1}(\mathbf{x},t)^{\boldsymbol{\alpha}}\qquad\text{and}\qquad g_2(\mathbf{x},t)=\sum_{|\alpha|=0}^{deg-1}g_{\boldsymbol{\alpha};2}(\mathbf{x},t)^{\boldsymbol{\alpha}} \tag{2.29}$$

are introduced to approximate

$$v_1(\mathbf{x},t)=\frac{u_1}{u_0}(\mathbf{x},t)\qquad\text{and}\qquad v_2(\mathbf{x},t)=\frac{u_2}{u_0}(\mathbf{x},t),$$

respectively.

The computation of all coefficients $A_{\boldsymbol{\alpha};k}$ of the Taylor polynomial $\mathbf{U}_i^u$ can be executed as a loop with increasing degree α_3 of the time derivative as follows:

$\boldsymbol{\alpha_3 = 0}$: For the coefficients $A_{(\alpha_1,\alpha_2,0);k}$ of the spatial monomials of $\mathbf{U}_i^n$

$$A_{(\alpha_1,\alpha_2,0);k} = a_{(\alpha_1,\alpha_2);k}$$

holds where $a_{(\alpha_1,\alpha_2);k}$ are the coefficients of $\mathbf{u}_i^n$. Thus, with theorem 2.9, for $A_{(\alpha_1,\alpha_2,0);k}$ it holds that

$$A_{(\alpha_1,\alpha_2,0);k} = \frac{1}{\alpha_1!\alpha_2!}\partial^{(\alpha_1,\alpha_2,0)}u_k|_{(\mathbf{b}_i,t^n)} + \mathcal{O}\left(h^{deg+1-(\alpha_1+\alpha_2)}\right).$$

Approximations $g_{(\alpha_1,\alpha_2,0);k}$, $k = 1, 2$, for the spatial derivatives of the auxiliary variables $\frac{1}{\alpha_1!\alpha_2!}v_{(\alpha_1,\alpha_2,0);k}|_{(\mathbf{b}_i,t^n)}$ can be computed using equation (2.25). For these approximations,

$$g_{(\alpha_1,\alpha_2,0);k} = \frac{1}{\alpha_1!\alpha_2!}v_{(\alpha_1,\alpha_2,0);k}|_{(\mathbf{b}_i,t^n)} + \mathcal{O}\left(h^{deg+1-(\alpha_1+\alpha_2)}\right)$$

is obtained by theorem 2.11. Finally, approximations to the spatial derivatives of the products u_0u_0, u_1v_1, u_1v_2, u_2v_2 can be computed using equation (2.24). Again, with remark 2.12

$$(A_0A_0)_{(\alpha_1,\alpha_2,0)} = (u_0u_0)_{(\alpha_1,\alpha_2,0)}|_{(\mathbf{b}_i,t^n)} + \mathcal{O}\left(h^{deg+1-(\alpha_1+\alpha_2)}\right)$$

holds and analogous results are obtained for $(A_1g_1)_{(\alpha_1,\alpha_2,0)}$, $(A_1g_2)_{(\alpha_1,\alpha_2,0)}$ and $(A_2g_2)_{(\alpha_1,\alpha_2,0)}$. Thus, the initialization for the CKP is complete.

$\boldsymbol{\alpha_3 = \alpha_3 + 1}$: The coefficients $A_{\boldsymbol{\alpha};k}$, $k = 0, 1, 2$, can be computed by the already determined quantities as

$$\begin{aligned}
A_{\boldsymbol{\alpha};0} &= -A_{(\alpha_1+1,\alpha_2,\alpha_3-1);1} - A_{(\alpha_1,\alpha_2+1,\alpha_3-1);2}\\
A_{\boldsymbol{\alpha};1} &= -(A_1g_1)_{(\alpha_1+1,\alpha_2,\alpha_3-1)} - \frac{1}{2}(A_0A_0)_{(\alpha_1+1,\alpha_2,\alpha_3-1)}\\
&\quad - (A_1g_2)_{(\alpha_1,\alpha_2+1,\alpha_3-1)}\\
A_{\boldsymbol{\alpha};2} &= -(A_2g_1)_{(\alpha_1+1,\alpha_2,\alpha_3-1)} - (A_2g_2)_{(\alpha_1,\alpha_2+1,\alpha_3-1)}\\
&\quad - \frac{1}{2}(A_0A_0)_{(\alpha_1,\alpha_2+1,\alpha_3-1)}
\end{aligned}$$

Obviously, as $|\boldsymbol{\alpha}| = (\alpha_1 + 1) + \alpha_2 + (\alpha_3 - 1) = \alpha_1 + (\alpha_2 + 1) + (\alpha_3 - 1)$, for $A_{\boldsymbol{\alpha};k}$, $k = 0, 1, 2$, it holds that

$$A_{\boldsymbol{\alpha};k} - \frac{1}{\boldsymbol{\alpha}!}\partial^{\boldsymbol{\alpha}}u_k|_{(\mathbf{b}_i,t^n)} = \mathcal{O}\left(h^{deg+1-|\boldsymbol{\alpha}|}\right). \tag{2.30}$$

After having computed the coefficients $A_{\boldsymbol{\alpha};k}$, the auxiliary quantities $g_{\boldsymbol{\alpha};k}$, $k=1,2$, and the products $(A_1 g_1)_{\boldsymbol{\alpha}}$, $(A_1 g_2)_{\boldsymbol{\alpha}}$, $(A_2 g_2)_{\boldsymbol{\alpha}}$, $(A_0 A_0)_{\boldsymbol{\alpha}}$ can be computed by first using equation (2.25) and then (2.24). For all these quantities it holds again, with theorem 2.11 and remark 2.12, that the approximation order is $deg + 1 - |\boldsymbol{\alpha}|$.

The space-time Taylor polynomials $\mathbf{U}_i^n(\mathbf{x}, t)$ are determined by the coefficients $A_{\boldsymbol{\alpha};k}$, $|\boldsymbol{\alpha}| \leq deg$ as approximation for $\mathbf{u}(\mathbf{x}, t)$ for the space-time cell $\sigma_i \times [t^n, t^{n+1}]$.

In the preceding passage, with equation (2.30) a proposition similar to the result obtained in theorem 2.9, this time concerning the coefficients of $\mathbf{U}_i^n(\mathbf{x}, t)$, was proven.

Another result, this time concerning the approximation order of the polynomial $\mathbf{U}_i^n(\mathbf{x}, t)$ in space and time, can be shown thus.

Theorem 2.13. *Let* $\mathbf{U}_i^n(\mathbf{x}, t) = \sum_{|\boldsymbol{\alpha}|=0}^{deg} \mathbf{A}_{\boldsymbol{\alpha}} (\mathbf{x} - \mathbf{b}_i, t - t^n)^{\boldsymbol{\alpha}}$ *with*

$$A_{\boldsymbol{\alpha};k} - \frac{1}{\boldsymbol{\alpha}!} \partial^{\boldsymbol{\alpha}} u_k|_{(\mathbf{b}_i, t^n)} = \mathcal{O}\left(h^{deg+1-|\boldsymbol{\alpha}|}\right).$$

Then for $(\mathbf{x}, t) \in \{\sigma_i\} \times [t^n, t^{n+1}]$ *it holds that*

$$u_k(\mathbf{x}, t) - U_{i;k}^n(\mathbf{x}, t) = \mathcal{O}\left(h^{deg+1}\right).$$

Proof. For the Taylor series

$$T_{i;k}^n(\mathbf{x}, t) = \sum_{|\boldsymbol{\alpha}|=0}^{deg} \frac{1}{\boldsymbol{\alpha}!} \partial^{\boldsymbol{\alpha}} u_k|_{\mathbf{b}_i, t^n} (\mathbf{x} - \mathbf{b}_i, t - t^n)^{\boldsymbol{\alpha}}$$

it holds that

$$T_{i;k}^n - u_k = \mathcal{O}\left(h^{deg+1}\right) \text{ for } (\mathbf{x}, t) \in \sigma_i \times \left[t^n, t^{n+1}\right].$$

Thus,

$$\begin{aligned} U_{i;k}^n(\mathbf{x}, t) - u_k(\mathbf{x}, t) = & U_{i;k}^n(\mathbf{x}, t) - T_{i;k}^n(\mathbf{x}, t) + \mathcal{O}\left(h^{deg+1}\right) \\ = & \sum_{|\boldsymbol{\alpha}|=0}^{deg} \left(A_{\boldsymbol{\alpha};k} - \frac{1}{\boldsymbol{\alpha}!} \partial^{\boldsymbol{\alpha}} u_k|_{\mathbf{b}_i, t^n} \right) (\mathbf{x} - \mathbf{b}_i, t - t^n)^{\boldsymbol{\alpha}} \\ & + \mathcal{O}\left(h^{deg+1}\right) \\ = & \sum_{|\boldsymbol{\alpha}|=0}^{deg} \mathcal{O}\left(h^{deg+1-|\boldsymbol{\alpha}|}\right) \underbrace{(\mathbf{x} - \mathbf{b}_i, t - t^n)}_{=\mathcal{O}(h)}{}^{\boldsymbol{\alpha}} \end{aligned}$$

$$
\begin{aligned}
&+\mathcal{O}\left(h^{deg+1}\right)\\
=&\mathcal{O}\left(h^{deg+1}\right)
\end{aligned}
$$

for $k = 0, 1, 2$. □

2.4 Riemann Solver

Riemann problems, as mentioned in definition 1.7, are special initial value problems (1.27) that appear in the context of this work at all integration points $(\tilde{\mathbf{x}}_{i,j}^{k,l}, t^{n,m})$ of the numerical scheme (1.49) and (1.50). Due to the finite volume ansatz, at those points exist, because of the two adjacent cells, two different approximations for $\mathbf{u}$, according to the cell-wise reconstruction.

The general solution of this problem is computed via a numerical flux function, or Riemann solver, $\mathbf{H}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$ that depends on the data given in the cells to the left and the right and the vector normal to the edge between these cells.

Riemann solvers are functions that return either the exact value or an approximation of the analytic flux $\mathbf{f}(\mathbf{u}_S)$, or the fluxes $\mathbf{f}(\mathbf{u}_l)$, $\mathbf{f}(\mathbf{u}_r)$, respectively, for the given left and right state of a Riemann problem.

Definition 2.14. The numerical flux function $\mathbf{H}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$ is called *consistent* to the partial differential equation, if

$$
\mathbf{H}(\mathbf{u}, \mathbf{u}, \mathbf{n}) = \sum_{i=1}^{2} \mathbf{f}_i(\mathbf{u}) n_i.
$$

Assuming $\mathbf{H}$ is smooth enough, [Son96] proves the following theorem concerning the spatial order of the numerical scheme.

Theorem 2.15. *If the order of accuracy of the quadrature rule is $\mathcal{O}(h^q)$ in* (1.50) *and if $(\mathbf{U}_i^n - \mathbf{u})(\mathbf{x}, t) = \mathcal{O}(h^r)$, $r > 0$, holds on $\sigma_i \in \Sigma$, then the finite volume approximation is of order $\mathcal{O}(h^{\min(q,r)})$ in space, provided* $\mathbf{u}$ *as well as* $\mathbf{H}$ *are smooth enough functions.*

As the 2D shallow water equations are hyperbolic, the solution of the full Riemann problem consists normally of four, including topography five, states that are separated by three, respectively four, waves that travel with a certain constant wave speed [Smo83, LeV02]. There exist different types of Riemann solvers. The main distinctions are 'complete versus incomplete' and 'exact versus approximative'. A complete Riemann solver takes into account all waves of the problem while an incomplete solver might neglect

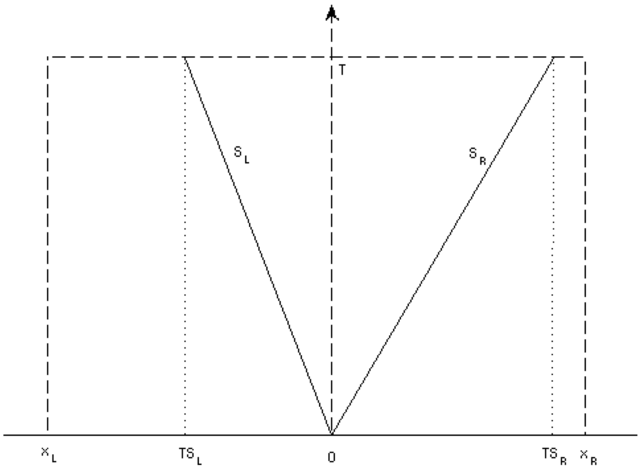

Figure 2.4: Ansatz for the HLL Riemann solver.

some. An exact solver will give the exact solution to the problem while an approximative one will just give an approximation. In both cases the approximation can be computed faster than the exact solution.

2.4.1 The HLL Riemann Solver

Harten, Lax and van Leer presented in [HLL83] a novel approach for solving the Riemann problem approximately. The HLL Riemann solver accounts for two waves with velocity S_L, S_R that separate three constant states $\mathbf{u}_L$, $\mathbf{u}_S$ and $\mathbf{u}_R$ and is independent from the differential equation. Thus, it is complete for the 1D shallow water equations but incomplete for the x_1-split 2D shallow water equations, as it neglects the contact discontinuity. It is assumed that both waves are shocks and thus have no width.

The guiding idea of the HLL is that for a conservation law the content of the quantities $\mathbf{u}$ in a fixed spatial area $[x_L, x_R]$ with $x_L < 0 < x_R$ changes in time only due to fluxes $\mathbf{f}(\mathbf{u})$ over the boundary of the area. Thus, on the one hand, the consistency condition

$$\begin{aligned}\int_{x_L}^{x_R} \mathbf{u}(x,T)\ dx = &\int_{x_L}^{x_R} \mathbf{u}(x,0)\ dx \\ &+ \int_0^T \mathbf{f}_1(\mathbf{u}(x_L,t))dt - \int_0^T \mathbf{f}_1(\mathbf{u}(x_R,t))\ dt \\ =\ & x_R\mathbf{u}_R - x_L\mathbf{u}_L + T(\mathbf{F}_L - \mathbf{F}_R) \end{aligned} \tag{2.31}$$

with $\mathbf{F}_K := \mathbf{f}_1(\mathbf{u}(x_K, 0))$ holds, as the states were assumed to be constant. Obviously, this equation is valid only for $x_L \leq TS_L$ and $TS_R \leq x_R$ respectively. This condition restricts the time step depending on the cell size and the wave speed.

On the other hand, splitting the integral at time T, the equation

$$\begin{aligned}\int_{x_L}^{x_R} \mathbf{u}(x,T)\ dx &= \int_{x_L}^{TS_L} \mathbf{u}(x,T)\ dx + \int_{TS_L}^{TS_R} \mathbf{u}(x,T)\ dx \\ &\quad + \int_{TS_R}^{x_R} \mathbf{u}(x,T)\ dx \\ &= \int_{TS_L}^{TS_R} \mathbf{u}(x,T)\ dx + (TS_L - x_L)\mathbf{u}_L + (x_R - TS_R)\mathbf{u}_R \end{aligned} \tag{2.32}$$

holds. The combination of equations (2.31) and (2.32) yields

$$\int_{TS_L}^{TS_R} \mathbf{u}(x,T)\ dx = T(S_R\mathbf{u}_R - S_L\mathbf{u}_L + \mathbf{F}_L - \mathbf{F}_R). \tag{2.33}$$

It is known from theorem 1.15 that $\mathbf{u}(x,T)|_{[TS_L, TS_R]}$ is constant. Thus,

$$\int_{TS_L}^{TS_R} \mathbf{u}(x,T)\ dx =: (TS_R - TS_L)\,\mathbf{u}_S.$$

By dividing equation (2.33) by the length of the integration interval, the integral mean value of the state between the waves can be computed, provided the wave velocities S_L and S_R are exact:

$$\mathbf{u}_S = \frac{S_R\mathbf{u}_R - S_L\mathbf{u}_L + \mathbf{F}_L - \mathbf{F}_R}{S_R - S_L}. \tag{2.34}$$

Summing up, Harten, Lax and van Leer proposed the following approximation for the solution of the Riemann problem:

$$\mathbf{u}(x,t) = \begin{cases} \mathbf{u}_L & \text{if} \quad \frac{x}{t} \leq S_L, \\ \mathbf{u}_S & \text{if} \quad S_L \leq \frac{x}{t} \leq S_R, \\ \mathbf{u}_R & \text{if} \quad S_R \leq \frac{x}{t}, \end{cases}$$

with $\mathbf{u}_S$ as in equation (2.34).

The corresponding flux along the t-axis in the supersonic cases $0 \leq S_L$ and $S_R \leq 0$ respectively is computed in the natural way as $\mathbf{f}(\mathbf{u}_L)$ and $\mathbf{f}(\mathbf{u}_R)$, but in the subsonic case another consideration was made.

Restricting equation (2.33) to the control volumes $[x_L, 0] \times [0, T]$ and $[0, x_R] \times [0, T]$ leads to

$$\int_{TS_L}^{0} \mathbf{u}(x,T)\, dx = -TS_L\mathbf{u}_L + T(\mathbf{F}_L - \mathbf{F}_{0L}), \tag{2.35a}$$

$$\int_{0}^{TS_R} \mathbf{u}(x,T)\, dx = TS_R\mathbf{u}_R + T(\mathbf{F}_{0R} - \mathbf{F}_R), \tag{2.35b}$$

where $\mathbf{F}_{0K},\ K \in \{L, R\}$, is the flux along the t-axis. Solving these equations for the new fluxes gives

$$\mathbf{F}_{0L} = \mathbf{F}_L - S_L\mathbf{u}_L - \frac{1}{T}\int_{TS_L}^{0} \mathbf{u}(x,T)\, dx,$$
$$\mathbf{F}_{0R} = \mathbf{F}_R - S_R\mathbf{u}_R + \frac{1}{T}\int_{0}^{TS_R} \mathbf{u}(x,T)\, dx.$$

Comparing the sum of the equations (2.35) to (2.33), it follows that

$$\mathbf{F}_{0L} = \mathbf{F}_{0R} =: \mathbf{F}^{HLL}.$$

$\mathbf{F}^{HLL}$ thus can be computed as

$$\mathbf{F}^{HLL} = \mathbf{F}_L + S_L(\mathbf{u}_S - \mathbf{u}_L) = \mathbf{F}_R + S_R(\mathbf{u}_S - \mathbf{u}_R),$$

which is in accordance with the Rankine-Hugoniot conditions (1.35).

The intercell flux in the finite volume method is then given by

$$\mathbf{F}^{HLL} = \begin{cases} \mathbf{F}_L & \text{if} \quad 0 \leq S_L, \\ \frac{S_R\mathbf{F}_L - S_L\mathbf{F}_R + S_LS_R(\mathbf{u}_R - \mathbf{u}_L)}{S_R - S_L} & \text{if} \quad S_L \leq 0 \leq S_R, \\ \mathbf{F}_R & \text{if} \quad 0 \geq S_R. \end{cases} \tag{2.36}$$

The HLL Riemann solver is consistent. This can be easily seen from equation (2.36), as with $\mathbf{u}_L = \mathbf{u}_R := \mathbf{u}$ follows $\mathbf{F}_L = \mathbf{F}_R := \mathbf{F}$ and thus $\mathbf{F}^{HLL} = \mathbf{F}$.

2.4.2 The HLLC Riemann Solver

The HLLC Riemann solver is based on similar considerations but additionally takes into account the contact discontinuity that travels with speed S_* with $S_L \leq S_* \leq S_R$. These three waves separate the four states $\mathbf{u}_L$, $\mathbf{u}_{*L}$, $\mathbf{u}_{*R}$ and $\mathbf{u}_R$. The HLLC was presented in [TSS94] for the two dimensional Euler equations and was applied to the 2D shallow water equations in [FrT93, FrT95].

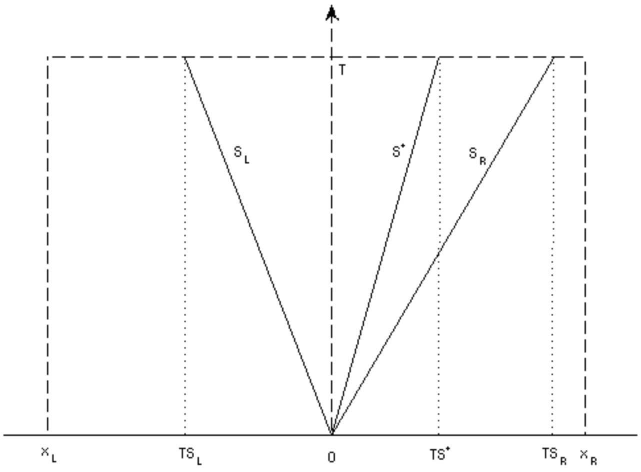

Figure 2.5: Ansatz for HLLC Riemann solver.

The considerations are mostly the same as for the HLL, but equation (2.32) is split in the following way:

$$\frac{1}{T(S_R - S_L)} \int_{TS_L}^{TS_R} \mathbf{u}(x,T)\ dx = \frac{1}{T(S_R - S_L)} \int_{TS_L}^{TS_*} \mathbf{u}(x,T)\ dx, \\ + \frac{1}{T(S_R - S_L)} \int_{TS_*}^{TS_R} \mathbf{u}(x,T)\ dx \qquad (2.37)$$

where the integral averages for both states in the middle are defined as

$$\mathbf{u}_{*L} := \frac{1}{T(S_* - S_L)} \int_{TS_L}^{TS_*} \mathbf{u}(x,T)\ dx, \\ \mathbf{u}_{*R} := \frac{1}{T(S_R - S_*)} \int_{TS_*}^{TS_R} \mathbf{u}(x,T)\ dx. \qquad (2.38)$$

Equations (2.33), (2.37) and (2.38) then yield the new consistency condition

$$\left(\frac{S_* - S_L}{S_R - S_L}\right) \mathbf{u}_{*L} + \left(\frac{S_R - S_*}{S_R - S_L}\right) \mathbf{u}_{*R} = \frac{S_R \mathbf{u}_R - S_L \mathbf{u}_L + \mathbf{F}_L - \mathbf{F}_R}{S_R - S_L}. \qquad (2.39)$$

The approximate solution of the Riemann problem is then given by

$$\mathbf{u}(x,t) = \begin{cases} \mathbf{u}_L & \text{if} \quad \frac{x}{t} \le S_L, \\ \mathbf{u}_{*L} & \text{if} \quad S_L \le \frac{x}{t} \le S_*, \\ \mathbf{u}_{*R} & \text{if} \quad S_* \le \frac{x}{t} \le S_R, \\ \mathbf{u}_R & \text{if} \quad S_R \le \frac{x}{t}. \end{cases}$$

Similar to the HLL, integrating over appropriate control volumes or applying the Rankine-Hugoniot conditions, respectively, leads to

$$\mathbf{F}_{*L} = \mathbf{F}_L + S_L(\mathbf{u}_{*L} - \mathbf{u}_L) \tag{2.40}$$

$$\mathbf{F}_{*R} = \mathbf{F}_{*L} + S_*(\mathbf{u}_{*R} - \mathbf{u}_{*L}) \tag{2.41}$$

$$\mathbf{F}_{*R} = \mathbf{F}_R + S_R(\mathbf{u}_{*R} - \mathbf{u}_R). \tag{2.42}$$

Inserting (2.40) and (2.42) into (2.41) again results in the consistency condition (2.39). So there are three equations for the four unknowns $\mathbf{u}_{*L}$, $\mathbf{u}_{*R}$, $\mathbf{F}_{*L}$ and $\mathbf{F}_{*R}$. This problem can be solved by taking the Riemann invariants (1.31)-(1.33) into account, that additionally in primitive formulation imply

$$\begin{aligned} H_{*L} =& \quad H_{*R} &&=: H_* \\ v_{*L;1} =& \quad v_{*R;1} &&=: v_{*;1} = S_* \\ v_{L;2} =& \quad v_{*L;2} \\ v_{R;2} =& \quad v_{*R;2}. \end{aligned}$$

As the Rankine-Hugoniot conditions only have to be applied to the conservative form, equation (2.40) gives

$$\begin{aligned} & \Phi_{*L} v_{*L;1} = \Phi_L v_{L;1} + S_L\left(\Phi_{*L} - \Phi_L\right) \\ \Rightarrow \quad & \Phi_{*L}(\underbrace{v_{*L;1}}_{S_*} - S_L) = \Phi_L(v_{L;1} - S_L) \\ \Rightarrow \quad & \Phi_{*L} = \Phi_* = \Phi_L\left(\frac{v_{L;1} - S_L}{S_* - S_L}\right). \end{aligned}$$

Equation (2.42) provides similar results for $\mathbf{u}_{*R}$. Thus, for $\mathbf{u}_{*K}$, $K \in \{L, R\}$,

$$\mathbf{u}_{*K} = \Phi_K\left(\frac{v_{K;1} - S_K}{S_* - S_K}\right)\begin{pmatrix} 1 \\ S_* \\ v_{K;2} \end{pmatrix} \tag{2.43}$$

holds. As Φ_{*L} equals Φ_{*R}, it is additionally possible to determine S_* via

$$\begin{aligned}\Phi_L\left(\frac{v_{L;1}-S_L}{S_*-S_L}\right) &= \Phi_R\left(\frac{v_{R;1}-S_R}{S_*-S_R}\right)\\ \Rightarrow S_* &= \frac{\Phi_L S_R(v_{L;1}-S_L)-\Phi_R S_L(v_{R;1}-S_R)}{\Phi_L(v_{L;1}-S_L)-\Phi_R(v_{R;1}-S_R)}. \qquad (2.44)\end{aligned}$$

The HLLC flux can finally be written as

$$\mathbf{F}^{HLLC} = \begin{cases}\mathbf{F}_L & \text{if } 0 \le S_L,\\ \mathbf{F}_{*L} = \mathbf{F}_L + S_L(\mathbf{u}_{*L}-\mathbf{u}_L) & \text{if } S_L \le 0 \le S_*,\\ \mathbf{F}_{*R} = \mathbf{F}_R + S_R(\mathbf{u}_{*R}-\mathbf{u}_R) & \text{if } S_* \le 0 \le S_R,\\ \mathbf{F}_R & \text{if } S_R \le 0.\end{cases} \qquad (2.45)$$

The HLLC Riemann solver is consistent, as from $\mathbf{u}_L = \mathbf{u}_R := \mathbf{u} = (\Phi, \Phi v_1, \Phi v_2)^T$ for $K \in \{L, R\}$ it follows from equation (2.44) that $S_* = v_1$ and thus

$$\mathbf{u}_{*K} = \Phi\left(\frac{v_1-S_K}{v_1-S_K}\right)\begin{pmatrix}1\\ v_1\\ v_2\end{pmatrix} = \mathbf{u}.$$

Then, consistency follows from $\mathbf{F}_L = \mathbf{F}_R := \mathbf{F}$, as by (2.45) $\mathbf{F}^{HLLC} = \mathbf{F}$ holds.

2.4.3 Wave Speed Estimations

The remaining problem is to obtain approximations for S_L and S_R. Two easy estimates proposed by Davis in [Dav88] consist of the eigenvalues related to the waves

$$\begin{aligned}S_L &= v_{L;1} - \sqrt{\Phi_L} = \lambda_{L;1},\\ S_R &= v_{R;1} + \sqrt{\Phi_R} = \lambda_{R;3}\end{aligned}$$

and

$$\begin{aligned}S_L &= \min(v_{L;1} - \sqrt{\Phi_L},\ v_{R;1} - \sqrt{\Phi_R}),\\ S_R &= \max(v_{L;1} + \sqrt{\Phi_L},\ v_{R;1} + \sqrt{\Phi_R}).\end{aligned}$$

Toro recommends

$$S_L = v_{L;1} - \sqrt{\Phi_L} q_L,\ S_R = v_{R;1} - \sqrt{\Phi_R} q_R$$

in [Tor01] where $q_K,\ K \in \{L,\ R\}$ are defined as

$$q_K = \begin{cases} \sqrt{\frac{(H_* + H_K)H_*}{2H_K^2}} & \text{if} \quad H_* > H_K, \\ 1 & \text{if} \quad H_* \leq H_K. \end{cases}$$

Hereby, H_* is an estimate for the exact solution for H in the region between S_L and S_R, that can be obtained, for example, by assuming S_L and S_R were both rarefactions and using the Riemann invariants

$$H_* = \frac{1}{\mathrm{g}} \left[\frac{1}{2}(\sqrt{\Phi_L} + \sqrt{\Phi_R}) + \frac{1}{4}(v_{L;1} - v_{R;1}) \right]^2. \tag{2.46}$$

The estimation recommended by Toro is the one that is used in the implementation of the HLL and HLLC in the program.

2.5 Including the Topography

In this section, the inclusion of source terms due to the bottom topography in the numerical scheme is treated.

Higher order schemes for the 2D shallow water equations that take into account topographical source terms are a subject of current research. Schemes were developed for example by Gallordo et al. in [GPC07] or by Xing and Shu in [XiS11]. These existing schemes are, to the authors knowledge, restricted in the order and/or bound to quadrilateral meshes.

An important point when including source terms is the well balanced-ness of the resulting scheme. Schemes that cannot balance the effect of the source term and the flux usually fail to capture steady states well and produce spurious oscillation near the steady state, [XiS11].

Definition 2.16. A numerical scheme for balance laws is called *well balanced*, if for a given *steady state solution*, that is a solution $\mathbf{u}$ with $\partial_t \mathbf{u} = \mathbf{0}$, the integral of the source terms and the numerical fluxes over the cell boundaries sum up to zero for each cell σ_i.

Hence, a well balanced scheme conserves a given steady state solution.

Generally, following [NXS07], steady state solutions of the 1D shallow water equations can be characterized by the relations

$$\Phi v_1 = const, \quad \text{and} \quad \frac{v_1^2}{2} + \Phi + \mathrm{g}top = const,$$

which represent, together with $v_2 = const$, the Riemann invariants related to the topography wave of the x_1-split 2D shallow water equations. For steady state solutions of the 2D shallow water equations according to [AuB05],

$$\nabla \cdot (\Phi \mathbf{v}) = 0 \;\text{ and }\; \nabla \left(\frac{|\mathbf{v}|^2}{2} + \Phi + \mathrm{g}\,top \right) + \begin{pmatrix} v_2 \\ -v_1 \end{pmatrix} \nabla \cdot \begin{pmatrix} v_2 \\ -v_1 \end{pmatrix} = 0$$

holds.

In this work the term well balanced is used in a weaker sense, namely in the sense of preserving the so called *still water steady states*. These problems are also referred to as *lake at rest*: The initial conditions describe the basin of a lake, given by *top*, and the water in the lake at rest with

$$(\Phi + \mathrm{g}\,top) = const \text{ and } \mathbf{v} = \mathbf{0}.$$

To include the influence of the topography, and to obtain a well balanced scheme, only a specialized Riemann solver and a few changes in the previously presented scheme are necessary.

At the start of a computation, a reconstruction polynomial

$$u_{i;3}(\mathbf{x}) = \sum_{|\boldsymbol{\alpha}|=0}^{deg} a_{\boldsymbol{\alpha};3} (\mathbf{x} - \mathbf{b}_i)^{\boldsymbol{\alpha}}$$

for *top* for each cell is computed, following the WENO procedure as presented in section 2.2. As *top* is constant with respect to time, this has to be done only once.

Moreover, for every spatial integration point $\tilde{\mathbf{x}}_{i,j}^{k,l}$, $l = 1, ..., N_x$, on the edge $\mathbf{l}_{i,j}^k$ between the cells σ_i and σ_j, the value $K_{i,j}^{k,l}$ related to the extent of the discontinuity in the reconstruction of topography is computed as

$$K_{i,j}^{k,l} = -\mathrm{g} \left(u_{i;3}(\tilde{\mathbf{x}}_{i,j}^{k,l}) - u_{j;3}(\tilde{\mathbf{x}}_{i,j}^{k,l}) \right). \tag{2.47}$$

2.5.1 Well Balanced Reconstruction

In order to obtain a reconstruction that fits in the previously presented scheme and results in a well balanced scheme, instead of reconstructing the Geo potential $\Phi = \mathrm{g}H$, the water surface $s := \mathrm{g}(H + top)$ is reconstructed for each stencil as before. This is only an extension of the previously reconstruction scheme, with which it is identical for $top(\mathbf{x}) \equiv 0$. From the obtained polynomials $p_{i,j;s}$ the basic reconstruction polynomials for Φ, $p_{i,j;0}$, can be computed as

$$p_{i,j;0}(\mathbf{x}) = p_{i,j;s}(\mathbf{x}) - \mathrm{g}u_{i;3}(\mathbf{x}).$$

It became obvious during the validation of the well balanced-ness of the scheme that rounding errors need to be avoided as far as possible. The first possible source of rounding errors in the reconstruction is the computation of $\hat{\mathbf{u}}$ as in equation (2.12). In the case of a still water steady state, $\hat{\mathbf{u}} = \mathbf{0}$ should hold for all quantities, which is often violated by small rounding errors occurring during the computation of the cell mean values. A simple remedy is to check whether $|\hat{u}_k| \leq 10^{-14}$ and to set $|\hat{u}_k| = 0$ if this is the case for all components $\hat{u}_k$ of $\hat{\mathbf{u}}$. Another source is the weighting. In the process of computing $u^n_{i;0}$ it turned out to be important to compute the sums (2.19) and (2.20) using $p_{i,j;s}$ instead of $p_{i,j;0}$ and subtracting the coefficients of $\mathrm{g}u_{i;3}$ as a last step.

In the case of the lake at rest, the reconstruction returns the polynomials

$$u^n_{i;0}(\mathbf{x}) = \mathit{const} - \mathrm{g}u_{i;3}(\mathbf{x}), \quad u^n_{i;1}(\mathbf{x}) = u^n_{i;2}(\mathbf{x}) \equiv 0, \tag{2.48}$$

as, as mentioned above, for the reconstruction vector $\hat{\mathbf{u}}$ as defined in equation (2.12) $\hat{\mathbf{u}} = \mathbf{0}$ for all quantities holds.

2.5.2 Including Source Terms in the Space Time Expansion

The idea of including the source terms into the space time expansion was carried out in [DuM07], but their resulting scheme was not well balanced. In doing so, the equation for the time derivatives of $\mathbf{u}$ changes into

$$\begin{aligned}
\partial_t u_0 &= -\partial_{x_1} u_1 - \partial_{x_2} u_2 \\
\partial_t u_1 &= -\partial_{x_1}\left(\frac{u_1^2}{u_0} + \frac{1}{2}u_0^2\right) - \partial_{x_2}\left(\frac{u_1 u_2}{u_0}\right) - \mathrm{g}\Phi\partial_{x_1} \mathit{top} \\
\partial_t u_2 &= -\partial_{x_1}\left(\frac{u_1 u_2}{u_0}\right) - \partial_{x_2}\left(\frac{u_2^2}{u_0} + \frac{1}{2}u_0^2\right) - \mathrm{g}\Phi\partial_{x_2} \mathit{top}.
\end{aligned}$$

The computation of the approximations of the higher mixed derivatives of the source terms is carried out analogously to the other quantities by inserting the reconstruction polynomial and the already known approximations. It is easily included into the CKP. As $\partial_t \mathit{top} = 0$, the Leibniz rule (2.24) for the derivatives of the source terms implies

$$(\Phi top_{(1,0,0)})_{\boldsymbol{\alpha}} = \sum_{i=0}^{\alpha_1}\sum_{j=0}^{\alpha_2}\binom{\alpha_1}{i}\binom{\alpha_2}{j}\Phi_{(\alpha_1-i,\alpha_2-j,\alpha_3)}top_{(i+1,j,0)}$$

and

$$(\Phi top_{(0,1,0)})_{\boldsymbol{\alpha}} = \sum_{i=0}^{\alpha_1}\sum_{j=0}^{\alpha_2}\binom{\alpha_1}{i}\binom{\alpha_2}{j}\Phi_{(\alpha_1-i,\alpha_2-j,\alpha_3)}top_{(i,j+1,0)},$$

respectively. When inserting the coefficients of the reconstruction polynomials, the statements about approximation order continue to hold, as one can see by considering the fact that for the computation of $\mathbf{A}_{\boldsymbol{\alpha};1}$ with $\alpha_3 > 0$ the derivative $(\Phi top_{(1,0,0)})_{(\alpha_1,\alpha_2,\alpha_3-1)}$ is used, which is $\mathcal{O}(h^{deg-|\boldsymbol{\alpha}|+1})$.

In the case of the lake at rest, for the Cauchy-Kovalewskaja procedure by (2.48), (2.24) and (2.25)

$$\begin{aligned}
\Phi_t|_{(\mathbf{b}_i,t^n)} &= \big(-(\underbrace{\Phi v_1}_{=0})_{x_1} - (\underbrace{\Phi v_2}_{=0})_{x_2}\big)|_{(\mathbf{b}_i,t^n)}\\
&= -\underbrace{A_{(1,0,0);1}}_{=0} - \underbrace{A_{(0,1,0);2}}_{=0} = 0\\
(\Phi v_1)_t|_{(\mathbf{b}_i,t^n)} &= \left(-\left(\frac{(\Phi v_1)^2}{\Phi} + \frac{1}{2}\Phi^2\right)_{x_1} - \left(\frac{\Phi v_1 \Phi v_2}{\Phi}\right)_{x_2} - \mathrm{g}\Phi top_{x_1}\right)|_{(\mathbf{b}_i,t^n)}\\
&= \left(-(\Phi v_1)_{x_1}\frac{\Phi v_1}{\Phi} + \Phi v_1\left((\Phi v_1)_{x_1} - \frac{\Phi v_1}{\Phi}\Phi_{x_1}\right)\frac{1}{\Phi}\right.\\
&\quad - \Phi_{x_1}\Phi - (\Phi v_2)_{x_2}\frac{\Phi v_1}{\Phi} + \Phi v_2\left((\Phi v_1)_{x_2} - \frac{\Phi v_1}{\Phi}\Phi_{x_2}\right)\frac{1}{\Phi}\\
&\quad \left. - \mathrm{g}\Phi top_{x_1}\right)|_{(\mathbf{b}_i,t^n)}\\
&= \mathrm{g}a_{(1,0);3}(const - \mathrm{g}a_{(0,0);3}) - \mathrm{g}(const - \mathrm{g}a_{(0,0);3})a_{(1,0);3}\\
&= 0
\end{aligned}$$

holds. Analogously, $(\Phi v_2)_t|_{(\mathbf{b}_i,t^n)} = 0$ can be obtained. The same results hold for all higher mixed derivatives $\mathbf{u}_{\boldsymbol{\alpha}}$ with $\alpha_3 \geq 1$. Thus, for the vector of space time reconstruction polynomials $\mathbf{U}_i^n$,

$$U_{i;k}^n(\mathbf{x},t) = u_{i;k}^n(\mathbf{x}),\ k = 0,1,2 \tag{2.49}$$

holds and the polynomials are constant with respect to time, which is compatible with the steady state solution. During the validation of the well balanced-ness of the scheme it turned out to be necessary to compute the derivatives of the terms $(\frac{1}{2}\Phi^2)_{x_1} = \Phi_{x_i}\Phi$ and $\mathrm{g}top_{x_i}\Phi$ together as derivatives of $\Phi(\Phi + \mathrm{g}top)_{x_i}$, $i = 1, 2$ to avoid rounding errors.

2.5.3 Riemann Solver including Topography

The Riemann solver presented by Chinnayya, LeRoux and Seguin in [Seg99, ChL99, CLS04] additionally accounts for the topography and the wave that arises hence. This work mainly refers to [Seg99, CLS04], as friction is not taken into account here.

By including the topography wave (the 4-wave), whose velocity always vanishes (see equation (1.26)), a new problem arises from the fact that the exact order of the waves is not known any more. As before for the wave speeds $S_1 \leq S_2 \leq S_3$ holds, but this system is given relatively while the speed $S_{top} = 0$ is given absolutely and independent from the others.

For each time space integration point $(\tilde{\mathbf{x}}_{i,j}^{k,l}, t^{n,m})$, $l = 1, ..., N_x$, $m = 1, ..., N_t$, the Riemann solver gets the reconstructed and rotated primitive values of the adjacent cells σ_i, σ_j at the time $t^{n,m}$ as

$$\mathbf{w}_L = \begin{pmatrix} H_L \\ v_{L;n} \\ v_{L;t} \end{pmatrix} := \mathbf{T}(\mathbf{n}_{i,j}^k)\mathbf{cp}(\mathbf{U}_i^n(\tilde{\mathbf{x}}_{i,j}^{k,l}, t^{n,m})),$$

$$\mathbf{w}_R = \begin{pmatrix} H_R \\ v_{R;n} \\ v_{R;t} \end{pmatrix} := \mathbf{T}(\mathbf{n}_{i,j}^k)\mathbf{cp}(\mathbf{U}_j^n(\tilde{\mathbf{x}}_{i,j}^{k,l}, t^{n,m})). \tag{2.50}$$

Here, $\mathbf{cp} : \mathbb{D} \to \mathbb{D}$, $\mathbf{u} \mapsto \mathbf{w}$ is the function that maps a set of conservative variables to its related set of primitive variables by

$$\mathbf{cp}(\mathbf{u}) = \begin{cases} \left(\frac{u_0}{\mathrm{g}}, \frac{u_1}{u_0}, \frac{u_2}{u_0}\right)^T & \text{if } u_0 > 0 \\ (0, 0, 0) & \text{else.} \end{cases}$$

The wave related to topography is situated exactly at the cell boundary and separates a state for the cell on the left and another one for the cell on the right side. This is reflected by the fact that two states $\mathbf{w}_l$ and $\mathbf{w}_r$, one for each cell, are computed indeed. In contrast to the HLL and HLLC approach where the resulting flux is not evaluated directly as $\mathbf{f}_1(\mathbf{u})$, the Godunov fluxes

$$\mathbf{T}^{-1}(\mathbf{n}_{i,j}^k)\mathbf{f}_1\left(\mathbf{cp}^{-1}(\mathbf{w}_l)\right) \text{ and } \mathbf{T}^{-1}(\mathbf{n}_{i,j}^k)\mathbf{f}_1\left(\mathbf{cp}^{-1}(\mathbf{w}_r)\right),$$

respectively, are computed for each side.

The ansatz for this Riemann solver is not made via integrals as for the HLL and HLLC, but by the construction of a continuous topography between two cells σ_L and σ_R by inserting a linear function $a_\varepsilon(x) : [-\varepsilon, \varepsilon] \mapsto \mathbb{R}$ with $a_\varepsilon(-\varepsilon) = u_{L;3}(-\varepsilon)$ and $a_\varepsilon(\varepsilon) = u_{R;3}(\varepsilon)$ at the discontinuity. The Riemann problem is solved then for this continuous topography depending on ε and the limit for $\varepsilon \to 0$ is considered. In the implementation used in this work, which is proposed in [Seg99] and represents a slight simplification of the original approach in [ChL99], this approach results in the same solution as the consideration of Riemann invariants and Rankine-Hugoniot conditions, the difference is present only in very special cases.

Considering the Riemann invariants and Rankine-Hugoniot conditions, the Riemann solver can be motivated geometrically as intersection of point sets in the (H, v_n)-plane, the phase plane. Due to the fact that one of the Riemann invariants (1.34) states that the *flow rate* $B := Hv_n$ is constant across the topography wave, the mass of water in the system nevertheless remains constant.

On the premise that $top_L \geq top_R$, which can always be obtained by rotating the system of coordinates, a state $\mathbf{w}_S$ is computed initially on the intersection of the sets RL and RR as defined in the equations (1.47) and (1.48). At first, the topography is ignored in that computation. From the Riemann invariants ρ_4^i, $i = 1, 2, 3$, that remain constant across the 4-wave as stated in theorem 1.39,

$$\rho_4^1 : \qquad H_l v_{l;n} = H_r v_{r;n} := B \tag{2.51a}$$

$$\rho_4^2 : \qquad v_{l;t} = v_{r;t} \tag{2.51b}$$

$$\rho_4^3 : \qquad \frac{B^2}{2H_r^2} + \mathrm{g}H_r + \mathrm{g}top_r = \frac{B^2}{2H_l^2} + \mathrm{g}H_l + \mathrm{g}top_l. \tag{2.51c}$$

follows. The last equation can be rearranged to

$$\Psi_B(H_r) - \Psi_B(H_l) = -\mathrm{g}(top_r - top_l) =: K \tag{2.52}$$

with

$$\Psi_B(H) := \frac{B^2}{2H^2} + gH. \tag{2.53}$$

The states $\mathbf{w}_l$ and $\mathbf{w}_r$ that are to the left and right of the 4-wave are computed then considering the Riemann invariants ρ_4^m, $m = 1, 2, 3$ and $K_{i,j}^{k,l}$.

The Riemann invariants (1.31) through (1.34) state that the only point where v_t changes is across the 2-wave , where $H,\ v_n$ and top remain constant. Thus, the determination of v_t at the 4-wave will be included only at the very end of the computation. $sign(v_{l;n}) = sign(v_{r;n})$ follows from $H_l v_{l;n} = H_r v_{r;n}$ and $H_K \geq 0,\ K \in \{l, r\}$. If $sign(v_{l;n}) \geq 0$, thus the tangential velocity v_t is $v_{l;t} = v_{r;t} = v_{L;t}$, otherwise $v_{l;t} = v_{r;t} = v_{R;t}$ holds.

As only v_n appears in the following, only v instead of v_n will be written to keep the notation clearer.

In [CLS04], an important proposition concerning existence and uniqueness of the solutions $\mathbf{w}_l,\ \mathbf{w}_r$ is made.

Theorem 2.17. *Assume that $top_L \geq top_R$. If $\mathbf{w}_l$ is supersonic (respectively subsonic), there exists one and only one state $\mathbf{w}_r$ such that the Riemann invariants* (2.51) *hold. Moreover, $\mathbf{w}_r$ is supersonic (respectively subsonic).*

If $\mathbf{w}_l$ is sonic, then two solutions are admissible, one supersonic and the other subsonic.

Being given initial states $\mathbf{w}_L,\ \mathbf{w}_R$, the solver thus returns the states $\mathbf{w}_l$, $\mathbf{w}_r$ that are to the left and to the right of the 4-wave with $S_{top} = 0$ and fulfill the following properties:

a. $H_l v_l = H_r v_r =: B_l$.

b. $\Psi_{B_l}(H_r) - \Psi_{B_l}(H_l) = K$.

Depending on the position of $\mathbf{w}_S$ in the phase plane, the problem will be tentatively sorted into one of three possible cases that depend on the expected position of the solution $\mathbf{w}_l$, $\mathbf{w}_r$ in the phase plane, see figure 2.6:

a. the positive supersonic case : $\mathbf{w}_l,\ \mathbf{w}_r \in T^+ = \{(H, v)|c \leq v\}$,

b. the subsonic case: $\mathbf{w}_l,\ \mathbf{w}_r \in F = \{(H, v)| - c < v < c\}$,

c. the negative supersonic case: $\mathbf{w}_l,\ \mathbf{w}_r \in T^- = \{(H, v)|v \leq -c\}$.

By the influence of the topography, that possibly accelerates the flow, the case may change subsequently to one with a lower number.

2.5.3.1 Determination of $\mathbf{w}_S$

When analyzing the problem in the phase plane, it follows that the state $\mathbf{w}_S$ that connects the given states $\mathbf{w}_L$ and $\mathbf{w}_R$ is the intersection of RL and RR as defined in (1.47),(1.48). H_S is used in the following computation as upper, or lower, boundary for the determination of H_l or H_r, in the cases where

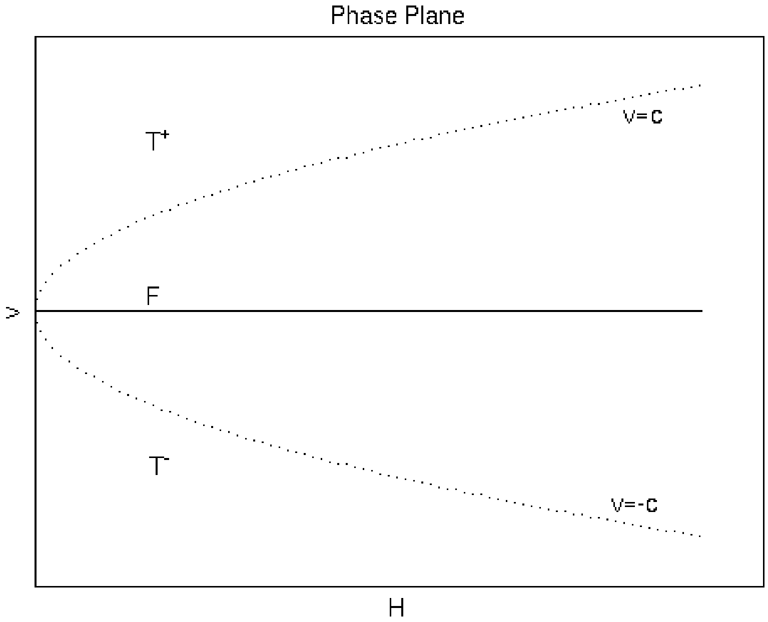

Figure 2.6: The phase plane.

$H_l \leq H_S \leq H_r$ holds. Therefore, in this implementation, unlike the usual proceeding, $\mathbf{w}_S$ is computed by a Newton iteration, as the usual estimations, like (2.46), are too inaccurate for the computation in the case of very small K.

2.5.3.2 Determination of $\mathbf{w}_l$ and $\mathbf{w}_r$

To determine the states $\mathbf{w}_l$ and $\mathbf{w}_r$, a lot of case differentiations have to be made. The main differentiation takes place due to the situation of $\mathbf{w}_S$, which gives a preliminary classification that might be corrected later due to the largeness of the topography jump that has its expression in K. The first two cases provide a good understanding of the whole solver. The third case will not be presented in this work, but can be found in [Seg99].

a Solution in positive supersonic regime.
$\mathbf{w}_R$ has no influence on the solution at all.

a.1 $\mathbf{w}_L \in T^+$.
All information comes from the left, so $\mathbf{w}_l = \mathbf{w}_L$

a.2 $\mathbf{w}_L \notin T^+$. The state $\mathbf{w}_l$ is sonic on the 1-rarefaction wave.
$\mathbf{w}_l$ is the intersection between RL and the graph of $v = c$.

Compute $\mathbf{w}_r$ supersonic on B_l with $\Psi_{B_l}(H_r) - \Psi_{B_l}(H_l) = K$.

b Solution expected to be in subsonic regime.

b.1 $\mathbf{w}_L \notin T^+$
Compute

$$\begin{aligned}
\mathbf{w}_{L_{max}} &= \{RL\} \cap \{v_1 = \sqrt{gH}\} \\
B_{max} &= H_{L_{max}} v_{L_{max}} \\
\mathbf{w}_{R_{max}} &= \{v = \frac{B_{max}}{H}\} \cap \{RR\} \\
K_{max} &= \Psi_{B_{max}}(H_{R_{max}}) - \Psi_{B_{max}}(H_{L_{max}}).
\end{aligned}$$

b.1.1 $K > K_{max}$
By simply determining $\mathbf{w}_l$ and $\mathbf{w}_r$ following case a.2, a non-physical solution may appear, as the state $\mathbf{w}_r$ is connected to $\mathbf{w}_R$ via a new inter state $\mathbf{w}_{S^*}$.
If K is not large enough, the velocity S^* of the 1-shock connecting $\mathbf{w}_r$ with $\mathbf{w}_{S^*}$ would be negative. It is easily confirmed with the first Rankine-Hugoniot condition (1.41a) that $S^* \geq 0 \Leftrightarrow H_{S^*}v_{S^*} \geq B_{max}$, as $H_r < H_{S^*}$. Compute $\mathbf{w}_N \in \{v = \frac{B_{max}}{H}\}$ such that $\mathbf{w}_N$ can be connected via a 1-shock to $\mathbf{w}_{R_{max}}$.

b.1.1.1 $K \geq \Psi(H_N) - \Psi(H_{L_{max}})$
The influence of K is so strong that the solution is situated in T^+, thus case a.2 holds, see figure 2.10.

b.1.1.2 $K < \Psi(H_N) - \Psi(H_{L_{max}})$
A stationary shock wave appears between $\mathbf{w}_l$ and $\mathbf{w}_r$.
Take $\mathbf{w}_l = \mathbf{w}_{L_{max}}$ and $\mathbf{w}_r = \mathbf{w}_{R_{max}}$.

b.1.2 $K \leq K_{max}$
Determine $\mathbf{w}_l \in RL$ and $\mathbf{w}_r \in RR$ with $\Psi(H_r) - \Psi(H_l) = K$ and $H_l v_l = H_r v_r$, see figures 2.7, 2.8, and 2.9.

b.2 $\mathbf{w}_L \in T^+$

$$\begin{aligned}
\mathbf{w}_{L_{max}} &= \{\{RL\} \cap \{v_1 = \frac{B_L}{H}\}\} \backslash \{L\} \\
B_{max} &= B_L \\
\mathbf{w}_{R_{max}} &= \{v_1 = \frac{B_{max}}{H}\} \cap \{RR\} \\
K_{max} &= \Psi(H_{R_{max}}) - \Psi(H_{L_{max}}).
\end{aligned}$$

The further treatment of this case is analog to case b.1 with the only difference that, if $K \geq \Psi(H_N) - \Psi(H_{L_{max}})$, case a.1 holds.

The series of figures 2.7 to 2.10 depicts the dependence of the solution on K in the case b.1. The setting for the whole series is $\mathbf{w}_L = \binom{1}{0}$, and $\mathbf{w}_R = \binom{2}{0}$.

The Riemann solver presented above is consistent, as $\mathbf{w}_S = \mathbf{w}$ follows from $\mathbf{w}_L = \mathbf{w}_R =: \mathbf{w}$ and $\mathbf{w}_l = \mathbf{w}_r = \mathbf{w}_S = \mathbf{w}$ follows from $K = 0$.

Using the modifications that were discussed in section 2.5 so far, it is finally possible to prove that the scheme presented in this work, using a combined space time series to provide the values for the flux computation, is well balanced indeed.

Theorem 2.18. *The numerical scheme presented above using the modified reconstruction, the modified Cauchy-Kovalewskaja procedure and the Riemann solver for topography is well balanced in the sense that it preserves still water steady states.*

Proof. For the vectors $\mathbf{w}_L$, $\mathbf{w}_R$ that were computed from the left and right reconstructed vectors $\mathbf{u}_L = \mathbf{U}_i^n(\tilde{\mathbf{x}}_{i,j}^{k,l}, t^{n,m})$, $\mathbf{u}_R = \mathbf{U}_j^n(\tilde{\mathbf{x}}_{i,j}^{k,l}, t^{n,m})$ according to (2.50) at each time space integration point $(\tilde{\mathbf{x}}_{i,j}^{k,l}, t^{n,m})$, $l = 1, ..., N_x$, $m = 1, ..., N_t$,

$$\mathbf{w}_L = \begin{pmatrix} \frac{const}{\mathrm{g}} - u_{i;3}^n \\ 0 \\ 0 \end{pmatrix}, \qquad \mathbf{w}_R = \begin{pmatrix} \frac{const}{\mathrm{g}} - u_{j;3}^n \\ 0 \\ 0 \end{pmatrix}$$

holds. These vectors depend only on the spatial integration point but remain constant with respect to the time integration points. This is a consequence of the fact that as with the well balanced-ness of the reconstruction and the modified Cauchy-Kovalewskaja procedure the vector of space time polynomials $\mathbf{U}_i^n$, $i = 0, ..., \#\Sigma - 1$, is constant with respect to time. As $w_{k;1} = v_{k;1} = 0$, $k \in \{L, R\}$, both states therefore have the same flow rate, namely $B = 0$. Computing $\Psi_0(H_L) - \Psi_0(H_R)$, the equation

$$\begin{aligned} \Psi_0(H_L) - \Psi_0(H_R) &= const - \mathrm{g}u_{i;3}^n(\tilde{\mathbf{x}}_{i,j}^{k,l}) - \left(const - \mathrm{g}u_{j;3}^n(\tilde{\mathbf{x}}_{i,j}^{k,l})\right) \\ &= -\mathrm{g}\left(u_{i;3}^n(\tilde{\mathbf{x}}_{i,j}^{k,l}) - u_{j;3}^n(\tilde{\mathbf{x}}_{i,j}^{k,l})\right) \\ &= K_{i,j}^{k,l} \end{aligned}$$

holds. Thus, the initial states $\mathbf{w}_L$, $\mathbf{w}_R$ are also the solution of the Riemann problem provided by the Riemann solver.

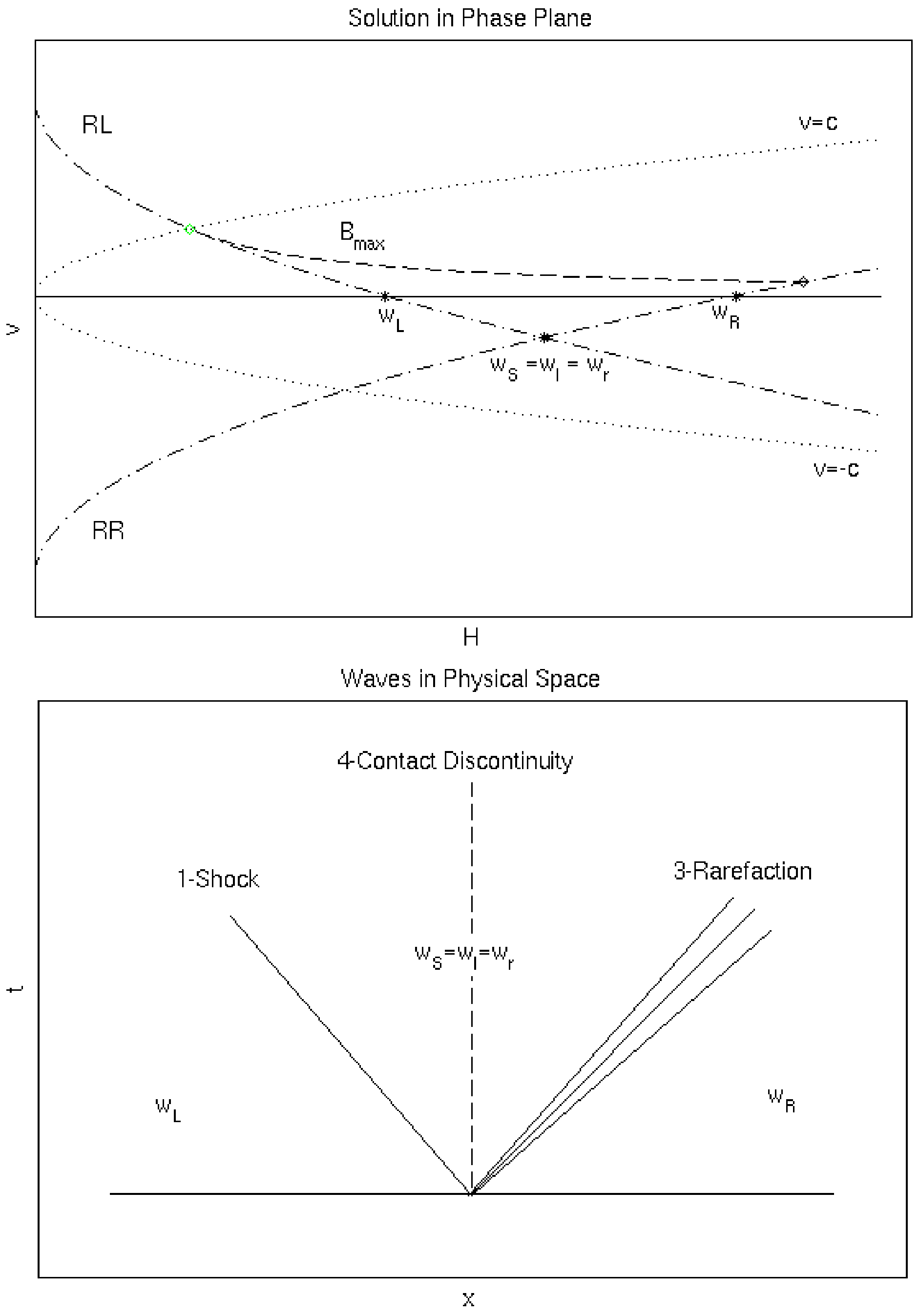

Figure 2.7: Solution of the Riemann Problem in the case of continuous topography ($K = 0$) in the phase plane and the physical space.

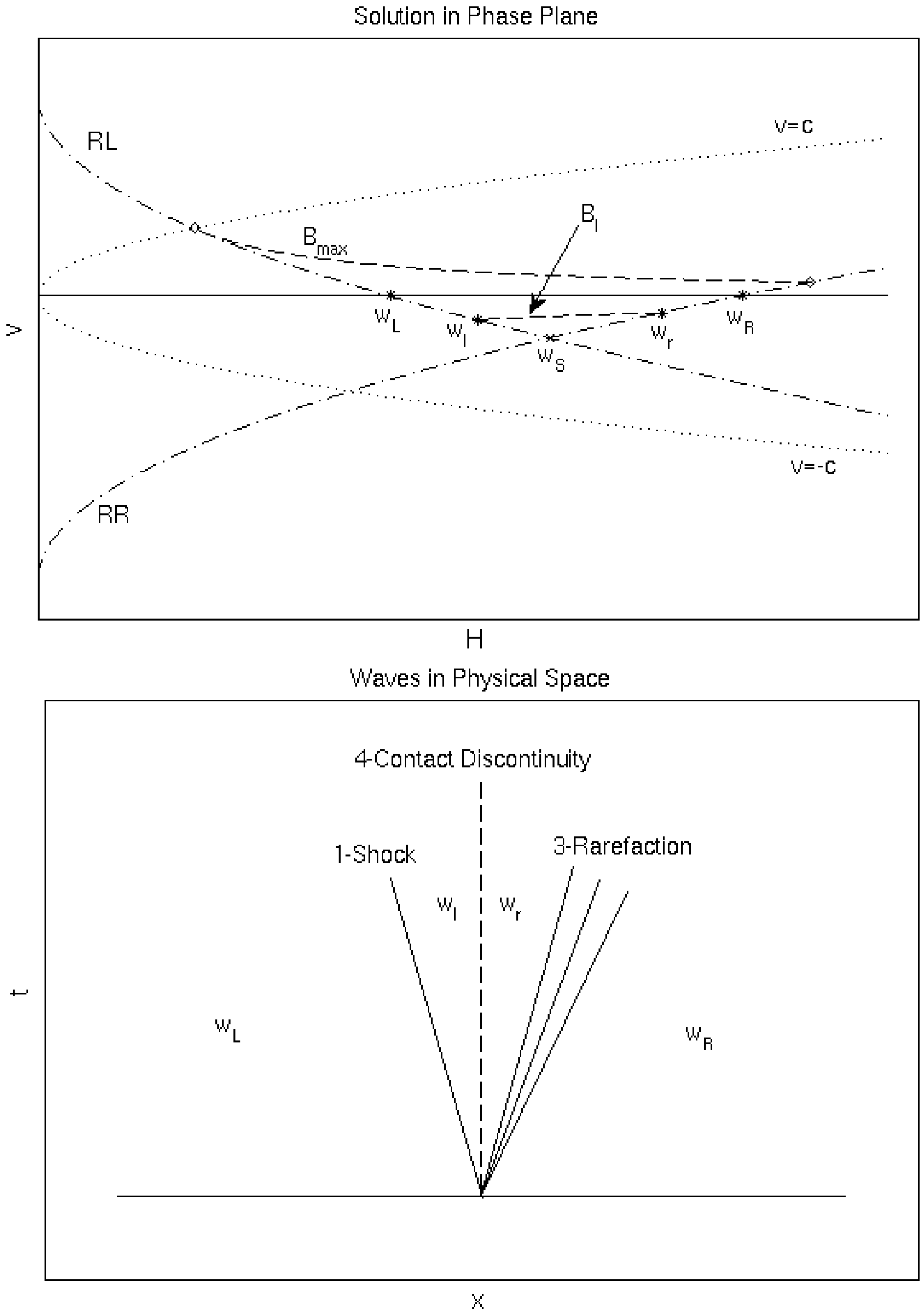

Figure 2.8: Solution of the Riemann Problem in the case of a small K in the phase plane and the physical space.

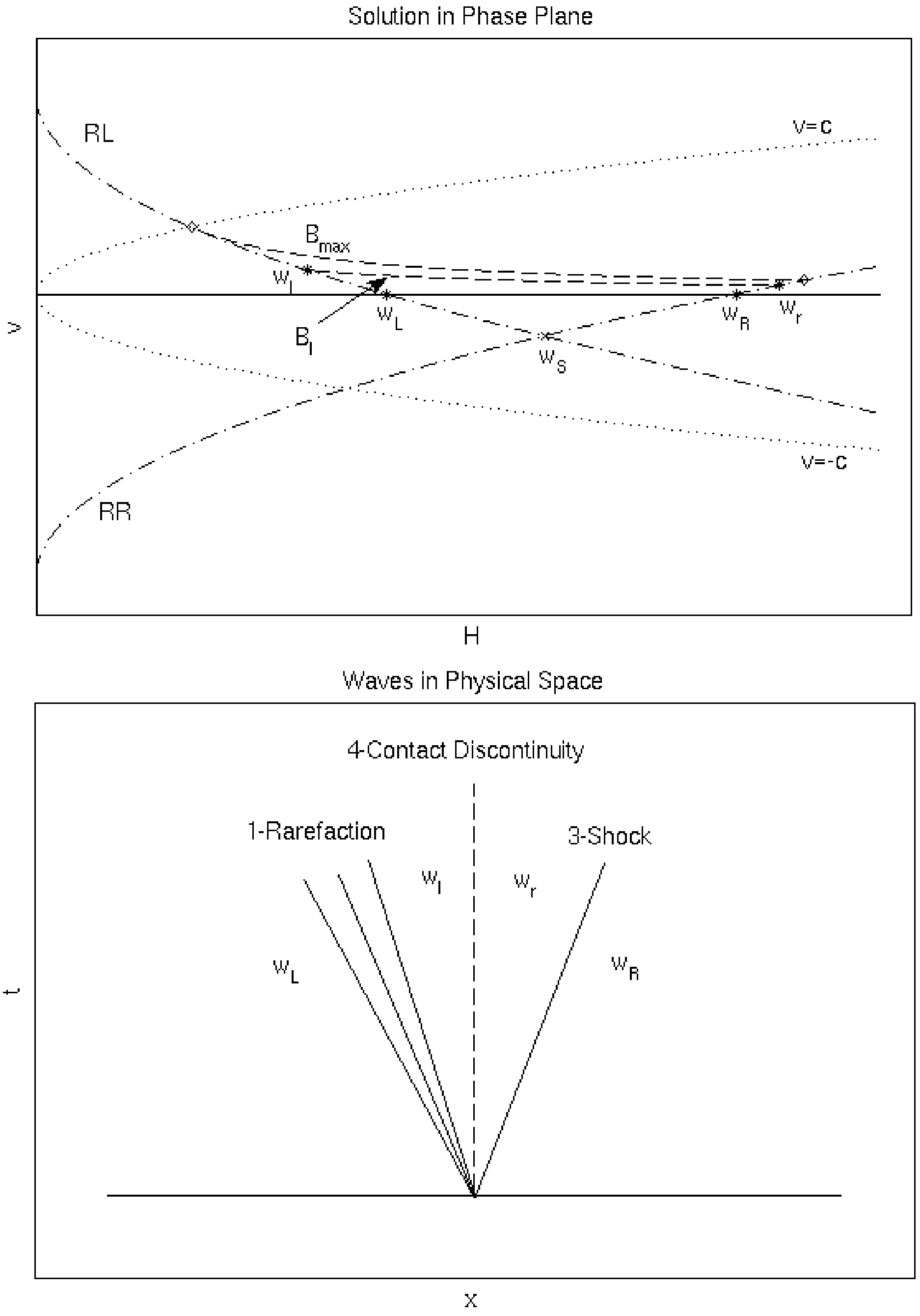

Figure 2.9: Solution of the Riemann Problem in the case of a medium-sized K in the phase plane and the physical space.

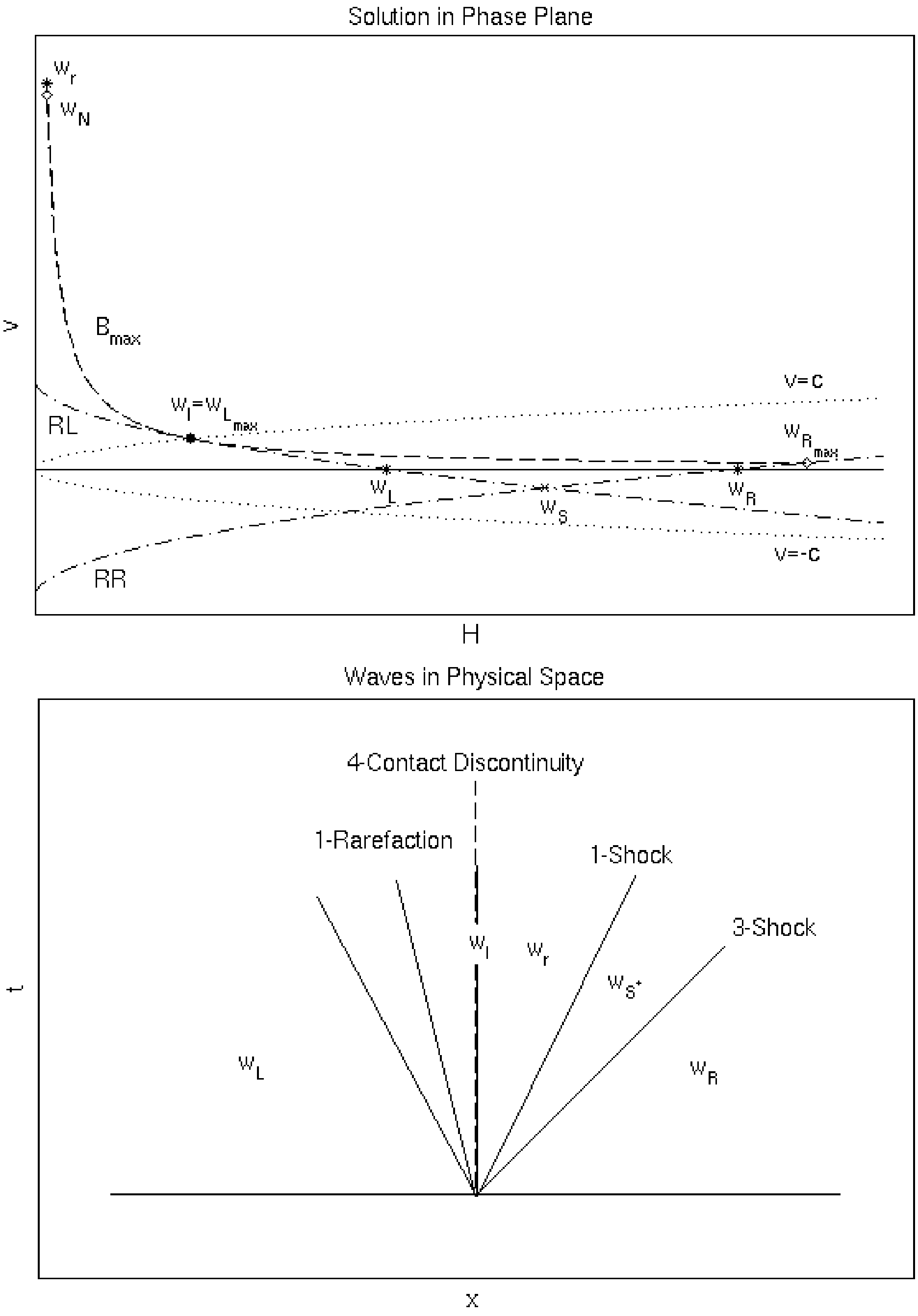

Figure 2.10: Solution of the Riemann Problem in the case of a large K in the phase plane and the physical space, case b.1.1.

For the net amount of the fluxes of the cell σ_i finally

$$\begin{aligned}
&\sum_{\mathbf{l}_{i,j}^k \in \delta\sigma_i} \int_{t^n}^{t^{n+1}} \int_{\mathbf{l}_{i,j}^k} \mathbf{T}^{-1}(\mathbf{n}_{i,j}^k)\mathbf{f}_1(\mathbf{cp}^{-1}(\mathbf{w}_L))\ ds\ dt \\
&= \sum_{\mathbf{l}_{i,j}^k \in \delta\sigma_i} \int_{t^n}^{t^{n+1}} \int_{\mathbf{l}_{i,j}^k} \mathbf{T}^{-1}(\mathbf{n}_{i,j}^k)\mathbf{f}_1(\mathbf{T}(\mathbf{n}_{i,j}^k)\mathbf{U}_i^n)\ ds\ dt \\
&= \sum_{\mathbf{l}_{i,j}^k \in \delta\sigma_i} \int_{t^n}^{t^{n+1}} \int_{\mathbf{l}_{i,j}^k} \sum_{m=1}^{2} \mathbf{f}_m(\mathbf{U}_i^n))n_{i,j;m}\ ds\ dt \\
&= \sum_{\mathbf{l}_{i,j}^k \in \delta\sigma_i} \int_{t^n}^{t^{n+1}} \int_{\mathbf{l}_{i,j}^k} \frac{1}{2} \begin{pmatrix} 0 \\ (const - \mathrm{g}u_{i;3})^2\, n_{i,j;1} \\ (const - \mathrm{g}u_{i;3})^2\, n_{i,j;2} \end{pmatrix} ds\ dt \\
&= \int_{t^n}^{t^{n+1}} \int_{\sigma_i} \begin{pmatrix} 0 \\ -\mathrm{g}\,(const - \mathrm{g}u_{i;3})\,\partial_{x_1} u_{i;3} \\ -\mathrm{g}\,(const - \mathrm{g}u_{i;3})\,\partial_{x_2} u_{i;3} \end{pmatrix} d\mathbf{x}\ dt \\
&= \int_{t^n}^{t^{n+1}} \int_{\sigma_i} \mathbf{g}(\mathbf{U}_i^n)\ d\mathbf{x}\ dt
\end{aligned}$$

holds. Obviously, the fluxes and the source terms are in balance when using an adequate high order quadrature rule for the computation of the integrals.

□

2.5.4 Dry Bed Problems

An interesting feature for the application of the scheme to real world problems is solving *dry bed problems.* This term covers two different cases: the covering with water of previously dry regions as well as the dry falling of previously wet regions. The difficulties in the treatment of dry states arise from the fact that the shallow water equations is only hyperbolic for wet regions, as otherwise it formally holds for the first and third eigenvector that $\mathbf{r}_1(\mathbf{u}) = (1, v_1, v_2) = \mathbf{r}_3(\mathbf{u})$. Moreover, the wave structure is different at the wet/dry front, see [Tor01]. Especially the state $\mathbf{w}_S$ does not exist any more.

In the case of covering a dry region, a simple remedy seems to be to introduce a small artificial water height $H = \varepsilon > 0$ to use in the computation instead of $H = 0$. This, however, leads to wrong results as in this case the wet-dry front is a shock wave which does not match with the physical behavior. This can be easily verified using the Rankine-Hugoniot conditions (1.42): Assume $\mathbf{u}_0$ to be a state with $\Phi_0 > 0$ and $\mathbf{u}$ a dry state with $\Phi = \Phi v_1 = \Phi v_2 = 0$. Then from the first equation

$$\Phi(v_1 - S) = \Phi_0(v_{0;1} - S)$$

the conclusion $S = v_{0;1}$ can be drawn. With the second equation

$$\Phi v_1(v_1 - S) + \frac{1}{2}\Phi^2 = \Phi_0 v_{0;1}(v_{0;1} - S) + \frac{1}{2}\Phi_0^2$$

it follows that $\Phi_0 = 0$, which contradicts the assumption.

In the case of dry falling, the positivity condition

$$RL(0) = v_{L;1} + 2c_L > v_{R;1} - 2c_R = RR(0),$$

that guarantees the existence of a state $\mathbf{w}_S$ with positive water height $H_S > 0$ at the intersection of RL and RR, is violated.

In figure 2.11, the solution of the case $RL(0) \leq 0 \leq RR(0)$ is depicted. The waves in physical space show the formation of a dry zone at the cell boundary. Figure 2.12 shows the case $0 \leq RL(0) \leq RR(0)$. The velocity $v_{L;1}$ is not 'high', in the sense of negative, enough to counteract the gravitation induced velocity $2\sqrt{gH_L}$ such that a mass flow over the cell boundary occurs. The third case, $RL(0) \leq RR(0) \leq 0$, works analogously.

The Riemann solver presented in this section can cope with both types of problems, the covering as well as the dry falling. Indeed, analyzing the problem in the phase plane it becomes clear that the case of covering a dry region is already contained within the case of a region falling dry: In the presence of a dry zone the state at the cell boundary is influenced by at least one of the initial states. Assuming that $\mathbf{w}_R = \mathbf{0}$, and thus RR ceases to exist, the solution depicted in the figures 2.11, 2.12 continues to hold true as it depends only on the question whether $RL(0) > 0$ or not. Analogously, for the solution of the case $\mathbf{w}_L = \mathbf{0}$ only on the question whether $RR(0) < 0$ is relevant.

The computation of topographical influences works in these setting exactly as in the wet bed case: If $RL(0) > 0$, the topography is included exactly as in the positive supersonic case a, depending on whether $\mathbf{w}_L \in T^+$ or not. If $RR(0) < 0$, the topography solution is found exactly as in a negative supersonic sub-case that is described in [Seg99].

Numerical problems occur in the presence of a dry zone in the Cauchy-Kovalewskaja procedure: In order to compute the (necessary) $\boldsymbol{\alpha}$-th derivatives of the auxiliary variables v_1, v_2, $|\boldsymbol{\alpha}|$ repeated divisions by the coefficient

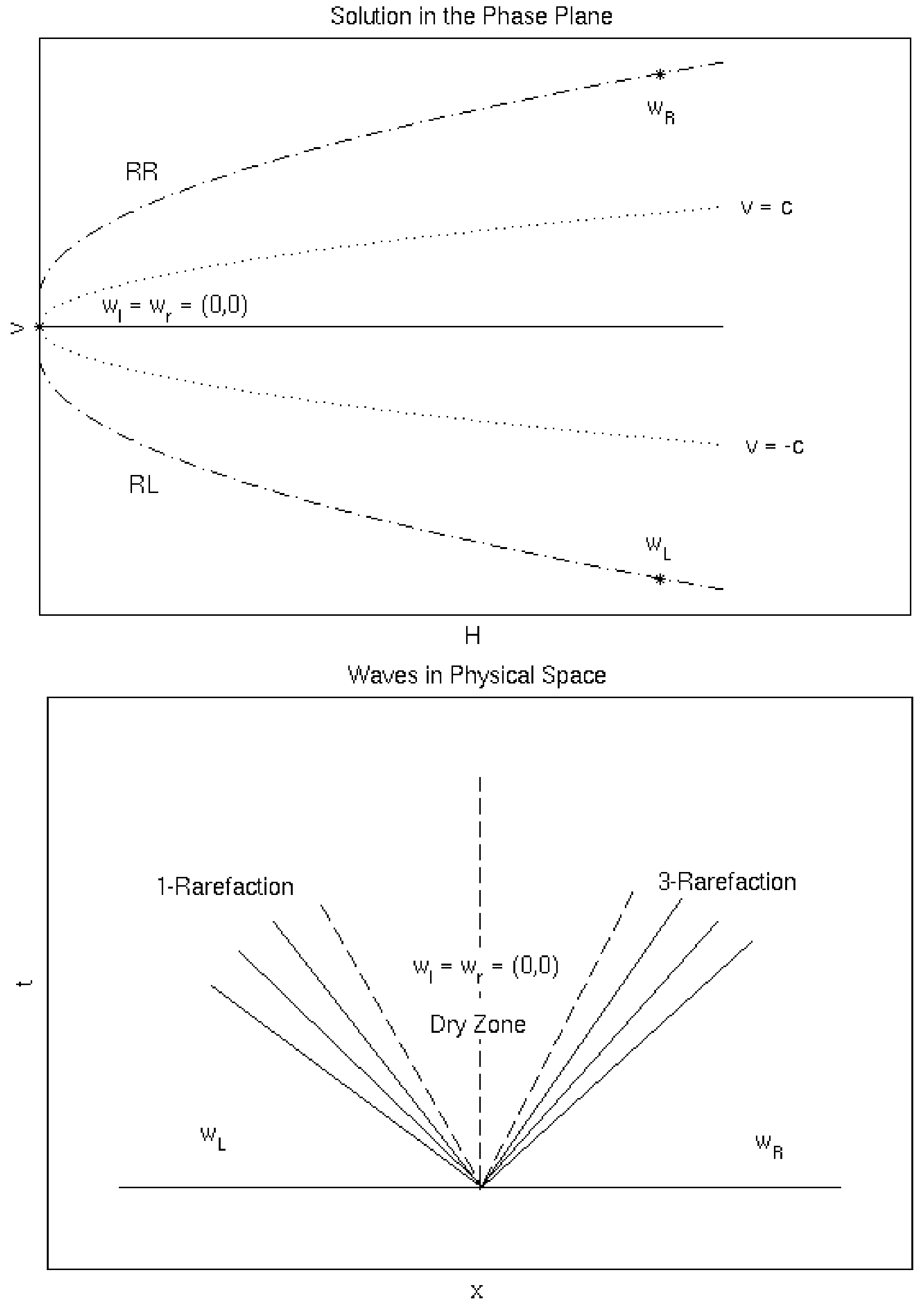

Figure 2.11: Solution of the Riemann Problem in the presence of a dry zone at the cell boundary in the phase plane and the physical space.

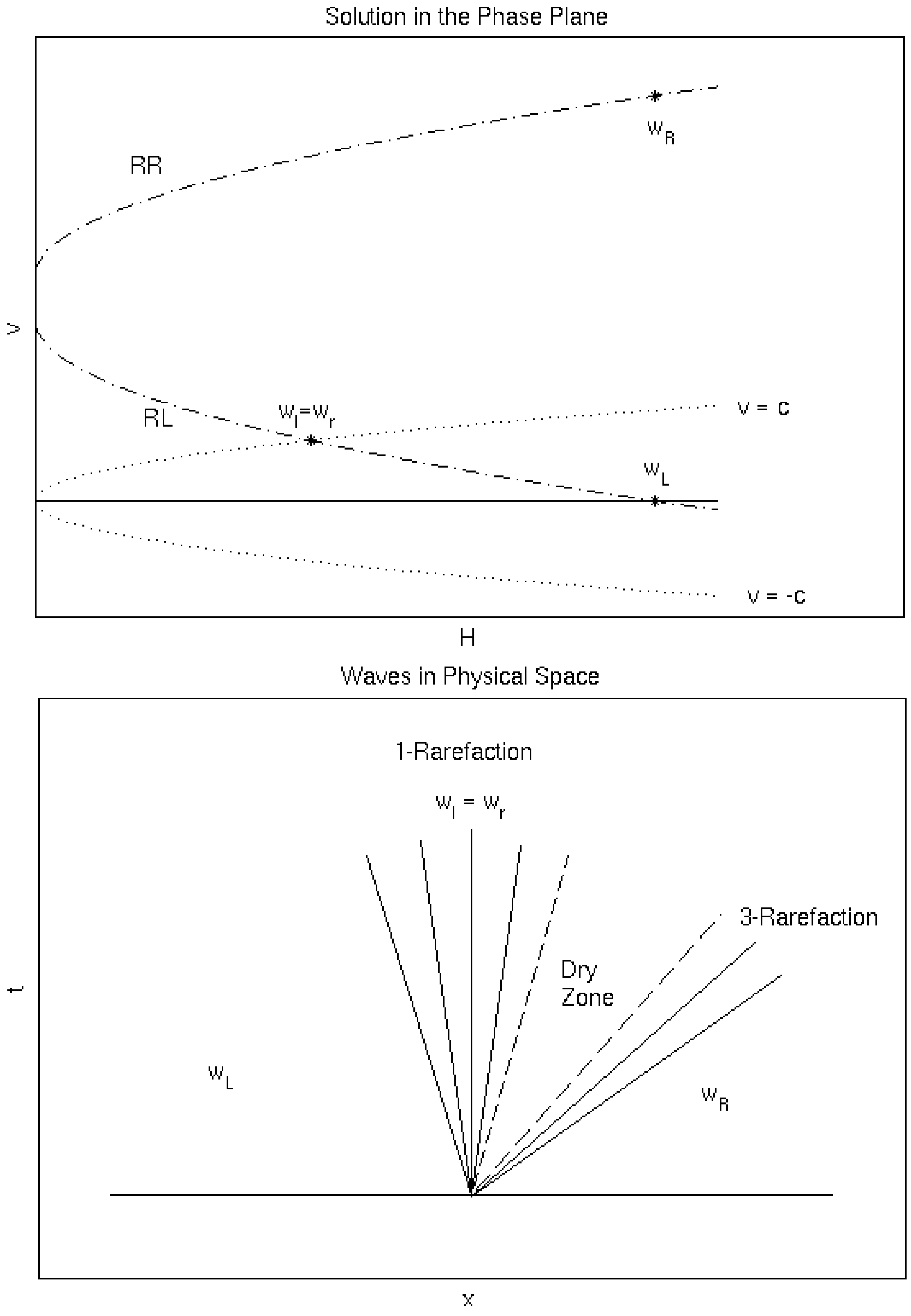

Figure 2.12: Solution of the Riemann Problem in the presence of a dry zone with flow over the cell boundary in the phase plane and the physical space, no topography.

$A_{(0,0,0);0}$ of the constant monomial are necessary. For these coefficients,

$$A_{(0,0,0);0} - u_0(\mathbf{b}_i, t^n) = \mathcal{O}(h^{deg+1})$$

holds with theorem 2.9. Thus, in a nearly dry zone $A_{(0,0,0);0}$ can be expected to be very small, and even small errors in its computation may lead to significantly wrong computed velocities [Tor01]. Moreover, reconstruction polynomials of a higher degree may, due to natural oscillation, provide regions with $U^n_{i;0}(\mathbf{x}, t) \leq 0$ in a wet cell. This is, of course, nonphysical as the water height is never negative. Worse, these polynomials are used in computing the space time expansion of the other variables and in integrating the source terms.

Development in the context of high order positivity preserving WENO reconstruction is a subject of recent research, for example in [XiS11] and the works mentioned therein. The limiter presented in that work can prevent $u_{i;0}$ from becoming negative on σ_i. Though the idea could be transferred to the three dimensional case and thus to $U_{i;0}$ on $\sigma_i \times [t^n, t^{n+1}]$, the limiter works under the condition that the minimal value of the function or its feasible approximation on σ_i, or $\sigma_i \times [t^n, t^{n+1}]$ respectively, is known. This poses indeed a problem for higher degree polynomials in more than one variable.

There is no simple and fast method to detect the location of the minimal value of $u^n_{i;0}$ on σ_i for polynomials of a degree higher than one, and thus it is not possible to directly ascertain the positivity of even the spatial reconstruction polynomial in the presence of very shallow water. On the other hand, for linear polynomials it is quite easy to check whether $u^n_{i;0}$, and even $U^n_{i;0}$, are positive on σ_i and $\sigma_i \times [t^n, t^{n+1}]$, respectively. Thus, in the vicinity of very shallow water, the order of the scheme is reduced in two steps. The consideration that is pursued in this work is to take into account the oscillation indicator $OI(u^n_{i;0})$, which is designed to detect oscillation. A large value of OI indicates that the values of $u^n_{i;0}|_{\sigma_i}$ span a wide range. Thus it is natural to use the oscillation indicator to decide whether a reduction of the reconstruction order for σ_i is advisable. The constant coefficient $a_{(0,0);0}$ gives the default value, and if

$$a_{(0,0);0} - \frac{r_i}{\sqrt{|\sigma_i|}} \sqrt{OI(u^n_{i;0})} < c_{red}, \tag{2.54}$$

with r_i being the maximal distance of the vertexes of σ_i to $\mathbf{b}_i$, the order of the numerical scheme is reduced to two in σ_i. The maximal distance r_i is constant and can thus be computed and stored once for each cell. Our tests have shown that $c_{red} = 5 \cdot 10^{-1}$ is a feasible choice. The choice of c_{red} is such that the reduction is handled quite liberally as the time development of the approximation can not be considered yet in the computation.

For linear polynomials, it is possible to estimate for each cell σ_i the maximal possible time step Δt_i, that guarantees values greater than a given constant c_{crit} for $U^n_{i;0}(\mathbf{x}, t),\ (\mathbf{x}, t) \in \sigma_i \times [0, \Delta t_i]$. Considering the value of the polynomial in the direction of the steepest descent

$$\mathbf{d} = \frac{-\nabla \mathbf{u}^n_i(\mathbf{b}_i)}{\|\nabla \mathbf{u}^n_i(\mathbf{b}_i)\|_2} = \frac{1}{\sqrt{a^2_{(1,0);0} + a^2_{(0,1);0}}} \begin{pmatrix} -a_{(1,0);0} \\ -a_{(0,1);0} \end{pmatrix}$$

at the maximal distance of the vertexes of σ_i to $\mathbf{b}_i$, it is very easy then to check whether the linear spatial reconstruction polynomial is positive on σ_i via

$$\begin{aligned} \Phi_{crit}(0) &= U^n_{i;0}(\mathbf{b}_i - r_i \mathbf{d}, 0) \\ &= u^n_{i;0}(\mathbf{b}_i - r_i \mathbf{d}) \\ &= a_{(0,0);0} - r_i \sqrt{a^2_{(1,0);0} + a^2_{(0,1);0}}. \end{aligned}$$

This condition is equivalent to (2.54) in the case of a linear reconstruction polynomial.

If $\Phi_{crit}(0)$ is smaller than c_{crit}, Δt_i is set to zero. Otherwise, Δt_i is determined using

$$\begin{aligned} c_{crit} = \Phi_{crit}(\Delta t_i) &= \Phi_{crit}(0) + A_{(0,0,1);0} \Delta t_i \\ &= \Phi_{crit}(0) - (a_{(1,0);1} + a_{(0,1);2}) \Delta t_i \\ \Leftrightarrow \Delta t_i &= \frac{\Phi_{crit}(0) - c_{crit}}{a_{(1,0);1} + a_{(0,1);2}}. \end{aligned}$$

Obviously, if $a_{(1,0);1} + a_{(0,1);2} \leq 0$ there won't occur any restrictions to Δt_i as $\Phi_{crit}(\Delta t)$ is not decreasing with advancing time. To take this into account and to avoid division by zero, the maximal time step is computed via

$$\Delta t_i = \frac{\Phi_{crit}(0) - c_{crit}}{\max(EPS, a_{(1,0);1} + a_{(0,1);2})}.$$

Comparing Δt_i with the general time step Δt computed by the scheme based on the velocities, if

$$\Delta t_i \leq \Delta t,$$

the order of the scheme is reduced to one for the cell σ_i. Thus, negative water heights can be avoided while computing the fluxes over the cell boundaries.

Our numerical experiments have shown that the choice $c_{crit} = 2.5 \cdot 10^{-1}$ is suitable for a stable scheme.

It is, however, not possible to guarantee that $|\sigma|^{-1}\, update_0 \leq \overline{u}^n_{i;0}$. If this condition is violated, $\overline{u}^{n+1}_i$ is set to zero for all components so that the loss of conservation of mass and momentum needs to be accepted.

Chapter 3

Computations

In this chapter, the scheme presented in the previous chapter is at first validated with respect to the desired numerical order. At the end of the chapter, some applications to more complex problems that have no reference solution are given. The validation happens in several stages: firstly, in section 3.1 it is shown that the theoretical order of reconstruction indeed is reached in the scheme. Secondly, in section 3.2 the full STE-scheme is validated with the linear advection equation. Next, the full scheme, using the HLLC Riemann-solver, is tested with the 2D-shallow water equations in section 3.3. The well balanced-ness is demonstrated in section 3.4. Next, the behavior of the scheme in the treatment of dry bed problems is depicted in section 3.5. Finally, the combination of topography and the occurring and vanishing of dry bed zones is treated in section 3.6 on the example of an oscillation lake.

Two applications of the scheme are shown at the end of this chapter. Firstly, a dam break with dry zones was computed in a channel that contains cone shaped obstacles given by topographical source terms. The results of this computation are shown in section 3.7. Secondly, the flowing of water from a reservoir through a channel winding down a hill is depicted in section 3.8.

The term ‘order‘ in this chapter is used in the sense of order of the whole scheme. The solution of a model problem is computed on the domain $\Omega = [-1.5, 1.5] \times [-1.5, 1.5]$ on a sequence of secondary grids G_1, G_2, ... ,G_n that result from the repeated application of red-refinement to the triangulation of a base grid G_0.

Generally, for the grid spacing h_n of grid G_n, $h_n \approx 2h_{n+1}$ holds due to the fact that the refinement is carried out on the primary grid, while the measure $\sqrt{|\sigma_i|} = \mathcal{O}(h)$ applies to the secondary grid. If the numerical scheme has numerical order k, one expects that $L_2(h_n) = \mathcal{O}(h_n^k)$ holds for the error of the numerical solution on grid G_n with grid spacing h_n. Thus, the numerical order of the scheme can be computed as

$$\text{order} = \log_2 \frac{L_2(h_{n-1})}{L_2(h_n)} \approx \log_2 \frac{\mathcal{O}(h_{n-1}^k)}{\mathcal{O}(h_n^k)} = \log_2 \left(2^k \frac{\mathcal{O}(h_n^k)}{\mathcal{O}(h_n^k)}\right) \approx k.$$

3.1 Validation of reconstruction order

The test includes only the WENO-reconstruction. One aim is to check whether polynomials of degree smaller than or equal to the reconstruction polynomial's degree are recovered adequately on all grids. Moreover, it is checked whether for polynomials of higher degree the theoretical reconstruction order is obtained. The degree of the reconstruction polynomial is *order - 1*.

The reconstruction is tested on the polynomials

$$p(\mathbf{x}) = \sum_{|\boldsymbol{\alpha}|=0}^{n} \mathbf{x}^{\boldsymbol{\alpha}} = \sum_{k=0}^{n} \sum_{\alpha_1=0}^{k} x_1^{\alpha_1} x_2^{k-\alpha_1}, \; n = 1, 2, 3, 4,$$

and is carried out with the set of variables (2.18) proposed in [DuK07] and in characteristic variables. The results displayed in table 3.1 show that the numerical order indeed matches the theoretical order.

3.2 Validation of the full scheme for the linear advection equation

The next step is the validation of the scheme on a less complicated equation. Convenient for this purpose is the two dimensional linear advection equation

$$\partial_t H + \mathbf{v} \cdot \nabla_{\mathbf{x}} H = 0, \tag{3.1}$$

where $\mathbf{v} = (v_1, v_2)^T$ is a constant velocity field with respect to time.

The test is carried out on the former sequence of grids $G_1, ..., G_5$. The velocity field $\mathbf{v}(\mathbf{x})$ and the function $H(\mathbf{x}, t)$ are chosen as analytical solution of the differential equation (3.1):

$$\begin{aligned}
\mathbf{v}(\mathbf{x}) &= \begin{pmatrix} -2\pi x_2 \\ 2\pi x_1 \end{pmatrix} \\
H(\mathbf{x}, t) &= \sin((x_1 \cos(2\pi t) + x_2 \sin(2\pi t))\pi) \\
&\quad \sin((-x_1 \sin(2\pi t) + x_2 \cos(2\pi t))\pi).
\end{aligned}$$

Order		2		3		4	
n	Grid	L_2-error	quot	L_2-error	quot	L_2-error	quot
1	G_0	$3.72e-16$		$3.59e-16$		$3.76e-16$	
	G_1	$3.68e-16$		$3.49e-16$		$3.86e-16$	
	G_2	$3.94e-16$		$3.93e-16$		$4.02e-16$	
	G_3	$4.01e-16$		$3.79e-16$		$4.08e-16$	
2	G_0	$2.70e-02$		$7.28e-16$		$8.61e-16$	
	G_1	$5.02e-03$	2.43	$8.36e-16$		$8.87e-16$	
	G_2	$1.02e-03$	2.30	$8.77e-16$		$9.09e-16$	
	G_3	$2.12e-04$	2.27	$8.16e-16$		$8.45e-16$	
3	G_0	$4.97e-02$		$1.50e-02$		$1.61e-15$	
	G_1	$1.34e-02$	1.89	$1.76e-03$	3.08	$1.73e-15$	
	G_2	$2.90e-03$	2.21	$2.42e-04$	2.88	$1.73e-15$	
	G_3	$6.13e-04$	2.24	$3.18e-05$	2.93	$1.67e-15$	
4	G_0	$1.49e-01$		$2.97e-02$		$2.30e-02$	
	G_1	$3.88e-02$	1.94	$5.18e-03$	2.52	$1.17e-03$	4.21
	G_2	$8.36e-03$	2.21	$7.48e-04$	2.79	$3.42e-05$	5.10
	G_3	$1.76e-03$	2.25	$1.01e-04$	2.89	$2.03e-06$	4.07

Table 3.1: Reconstruction order

The initial values for the cells σ_i are $\overline{H}_i^0 \approx \int_{\sigma_i} H(x_1, x_2, t^0)d\mathbf{x}$, for which the integration is again carried out by a spatial quadrature rule of order 9. During the computation, on the boundaries of Ω the exact solution is prescribed.

For the velocity field given above, the higher mixed derivatives of H provided by the Cauchy Kovalewskaja procedure take the form

$$\begin{aligned} H_{(a,b,c)} &= 2\pi x_2 H_{(a+1,b,c-1)} + b2\pi H_{(a+1,b-1,c-1)} \\ &\quad - 2\pi x_1 H_{(a,b+1,c-1)} - a2\pi H_{(a-1,b+1,c-1)}, \qquad a,b,c \geq 1, \end{aligned}$$

where $H_{(a,b,c)} := \frac{\partial^{a+b+c}}{\partial x_1^a \partial x_2^b \partial t^c}$.

The fluxes over the cell boundaries $\mathbf{l}_{ij}^k$ with normal vector $\mathbf{n} := \mathbf{n}_{ij}^k$ are computed via the (natural) upwind approach: if the normal velocity $v_n := n_1 v_1 + n_2 v_2 > 0$, then the value of σ_i is taken for the computation of the flux, otherwise the value of σ_j.

Order	2		3		4	
Grid	L_2-error	quot	L_2-error	quot	L_2-error	quot
G_1	$1.02e-01$		$3.98e-02$		$7.33e-02$	
G_2	$3.05e-02$	1.74	$6.74e-03$	2.56	$1.10e-02$	2.74
G_3	$7.84e-03$	1.96	$1.24e-03$	2.44	$1.47e-03$	2.90
G_4	$1.58e-03$	2.31	$1.25e-04$	3.31	$2.28e-06$	9.33
G_5	$2.84e-04$	2.48	$1.70e-05$	2.88	$1.24e-07$	4.20

Table 3.2: Linear advection equation, time step size = 0.0004, 2500 time steps.

The average numerical order of the fourth order scheme over the grids G_1 to G_5 is $\frac{\ln\left(\frac{7.33e-02}{1.24e-07}\right)}{4\ln 2} = 4.79$.

Thus, it can be stated that numerical order of the scheme matches the theoretical.

3.3 Validation of the full scheme for the 2D Shallow Water Equations without topography

Finally, the full scheme as it is described in chapter 2 is validated. The validation problem is similar to the test case used for the linear advection equation, but due to the different equation, in this case source terms will appear. The Riemann solver used in this test is the HLLC, as the source terms are not constant with respect to time as it is required for the use of the Riemann solver presented in section 2.5.3. The validation problem, that is also a solution of the 2D shallow water equations, consists of the equations

$$\begin{aligned}\Phi(x_1,x_2,t) &= g(\sin((x_1\cos(2\pi t)+x_2\sin(2\pi t))\pi)\\ &\qquad \sin((-x_1\sin(2\pi t)+x_2\cos(2\pi t))\pi)+2)\\ \mathbf{v}(x_1,x_2,t) &= \begin{pmatrix} -2\pi x_2 \\ 2\pi x_1 \end{pmatrix}\\ top(x_1,x_2,t) &= \frac{-1}{g}\left(\Phi - 2\pi^2(x_1^2+x_2^2)\right),\end{aligned}$$

where $top(x_1,x_2,t)$ can be obtained by inserting $\Phi(x_1,x_2,t)$ and $\mathbf{v}(x_1,x_2,t)$ into equation (1.18).

The initialization of the conserved quantities $\overline{\mathbf{u}}_i^0$ is done again with a spatial quadrature rule of order 9. The Cauchy Kovalewskaja procedure uses the exact derivatives of the source terms $\Phi(\partial_{x_1}\Phi - 4\pi^2 x_1)$ and $\Phi(\partial_{x_2}\Phi - 4\pi^2 x_2)$, respectively. The integration of the source terms in each time step is carried out for the exact source term and with a quadrature rule of the order according to the theoretical order of the scheme.

Order	2		3		4	
Grid	L_2-error	quot	L_2-error	quot	L_2-error	quot
G_0	$1.40e-01$		$1.14e-01$		$5.89e-02$	
G_1	$3.80e-02$	1.88	$2.00e-02$	2.51	$1.32e-02$	2.16
G_2	$9.50e-03$	2.00	$1.54e-03$	3.70	$9.54e-04$	3.79
G_3	$2.27e-03$	2.07	$1.78e-04$	3.11	$5.90e-05$	4.02

Table 3.3: Shallow water equations, time step size = 0.00001, 2000 time steps, reconstruction in characteristic variables

3.4 Well Balanced-ness of the scheme

The term well balanced is used in this context, as discussed in section 2.5, in the sense of preserving still water steady states, or so-called lake at rest problems. The reconstruction is carried out as mentioned, and the topography Riemann solver is applied generally.

The validation problem consists of the initial equations

$$\begin{aligned}\Phi(x_1,x_2,t^0) &= \mathrm{g}(0.7 - \mathit{top}(x_1,x_2))\\ \mathbf{v}(x_1,x_2,t^0) &= \begin{pmatrix}0\\0\end{pmatrix}\\ \mathit{top}(x_1,x_2) &= -0.3\tanh(5x_1).\end{aligned}$$

The initialization of the conserved quantities $\overline{\mathbf{u}}_i^0$ is done again with a spatial quadrature rule of order 9. The Cauchy Kovalewskaja procedure uses in the n-th time step the derivatives of the reconstruction polynomials of the source terms $\mathrm{g}u_{0;i}^n\partial_{x_1}u_{i;3}$ and $\mathrm{g}u_{0;i}^n\partial_{x_2}u_{i;3}$, respectively. The integration of the source terms in each time step is carried out using the integral cell mean values $|\sigma_i|^{-1}\int_{\sigma_i}(\mathbf{x}-\mathbf{b}_i)^{\boldsymbol{\alpha}}$, $|\boldsymbol{\alpha}| \leq 2\deg -1$, computed in the initialization step of the scheme and the coefficients provided by the reconstruction polynomials.

In the case of demonstrating well balanced-ness, it is necessary to increase the order of the quadrature rules used to compute the fluxes to $2\deg$, as can be seen in the proof of theorem 2.18. The measures considered herein are the residuum

$$\mathit{res} := \frac{\sqrt{\sum_{i=0}^{\#\Sigma-1}\frac{|\sigma_i|}{|\Omega|}\sum_{j=0}^{2}\left(\mathit{update}(u_{i;j})\right)^2}}{t^1-t^0}$$

of the first time step and the maximum of the updates of the conserved variables in the first time step

$$\mathit{maxup} := \max_{\substack{\sigma_i\in\Sigma\\ j=0,1,2}} \mathit{update}(u_{i;j}).$$

The results, that are given in the table 3.4, are in the range of the computational epsilon EPS $= 10^{-14}$ used in this scheme. The space time series for $\Phi(\mathbf{x},t)$ is indeed constant with respect to time in the sense that for all coefficients $A_{\boldsymbol{\alpha};0}$ with $\alpha_3 > 0$ it holds that $A_{\boldsymbol{\alpha};j} = 0.0$. In the other components, as expected, $U_{i;j} \equiv 0,\ j = 1,2$ holds.

The errors in the flux computation are due to rounding and stem from the rotation of the conservative values to the direction normal to the cell boundary and the rotation of the computed flux back to (x_1,x_2)-direction, and the use of a quadrature rule. This conjecture is conform with the fact that the maxima of all updates *maxup* are all in the same order of magnitude, while the residuum *res* approximately doubles for each refinement of the grid, that leads to a halving of the time step.

Order	2		3		4	
Grid	*res*	*maxup*	*res*	*maxup*	*res*	*maxup*
G_0	3.02*e*-13	1.01*e*-14	2.64*e*-13	3.34*e*-15	2.23*e*-13	5.06*e*-15
G_1	4.87*e*-13	1.35*e*-14	5.39*e*-13	8.94*e*-15	5.81*e*-13	1.34*e*-14
G_2	8.66*e*-13	1.52*e*-14	9.95*e*-13	9.04*e*-15	1.17*e*-12	1.13*e*-14
G_3	1.52*e*-12	1.23*e*-14	2.32*e*-12	1.28*e*-14	2.02*e*-12	1.34*e*-14

Table 3.4: Shallow water equations, time step size net-dependent, one time step.

3.5 Correct Solution of Dry Bed Problems

Toro states in [Tor01] that the difficulties in treating dry bed problems arise on the one hand from the different wave structure compared to the wet bed case and on the other hand from the need to compute the particle velocity $v(\mathbf{x}, \mathbf{t})$ from the conserved variables $\Phi(\mathbf{x}, \mathbf{t})$, $\Phi v(\mathbf{x}, \mathbf{t})$. This easily leads to a wrong computed velocity of the wet/dry front. Using the Riemann solver presented in section 2.5 takes care of the first problem, restricting $\Phi(\mathbf{x}, \mathbf{t})$ by reducing the order of the scheme as described later in the same section solves the latter.

The solution of a Riemann problem containing a left dry state is given in [Tor01] by

$$H(x,t) = \begin{cases} 0 & \frac{x}{t} \leq RR(0) \\ \frac{(-v_R+2c_R+\frac{x}{t})^2}{9\text{g}} & RR(0) < \frac{x}{t} \leq v_R + c_R \\ H_R & v_R + c_R < \frac{x}{t} \end{cases},$$

$$v(x,t) = \begin{cases} 0 & \frac{x}{t} \leq RR(0) \\ \frac{(v_R-2c_R+\frac{2x}{t})}{3} & RR(0) < \frac{x}{t} \leq v_R + c_R \\ v_R & v_R + c_R < \frac{x}{t} \end{cases}, \tag{3.2}$$

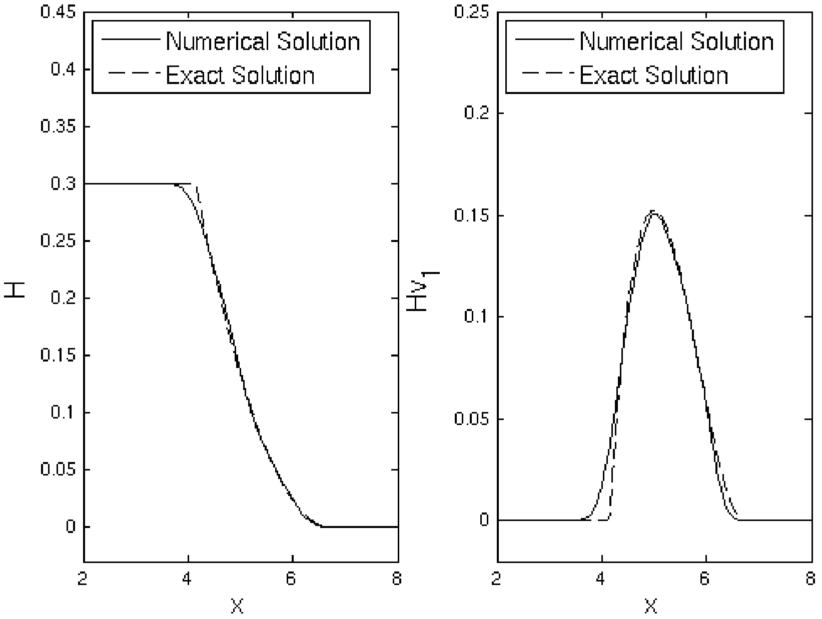

Figure 3.1: Solution of the Riemann Problem with a right dry initial state, completely first order.

while the solution in case of a right dry state is given by

$$H(x,t) = \begin{cases} H_L & \frac{x}{t} \leq v_L - c_L \\ \frac{(v_L + 2c_L - \frac{x}{t})^2}{9\text{g}} & v_L - c_L < \frac{x}{t} \leq RL(0) \\ 0 & RL(0) < \frac{x}{t} \end{cases} ,$$

$$v(x,t) = \begin{cases} v_L & \frac{x}{t} \leq v_L - c_L \\ \frac{(v_L + 2c_L + \frac{2x}{t})}{3} & v_L - c_L < \frac{x}{t} \leq RL(0) \\ 0 & RL(0) < \frac{x}{t} \end{cases} . \tag{3.3}$$

In the figures 3.1 to 3.3, the numerical and the exact solution for the water height H and the flow rate Hv_1 of the Riemann problem (1.27) with the initial values

$$\mathbf{u}_L = \begin{pmatrix} 0.1\text{g} \\ -0.3\text{g} \\ 0 \end{pmatrix}, \qquad \mathbf{u}_R = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

after $0.5s$ are depicted. The switches that show the reduction of the numerical order are displayed by the dotted and the dash-dotted line. In the regions

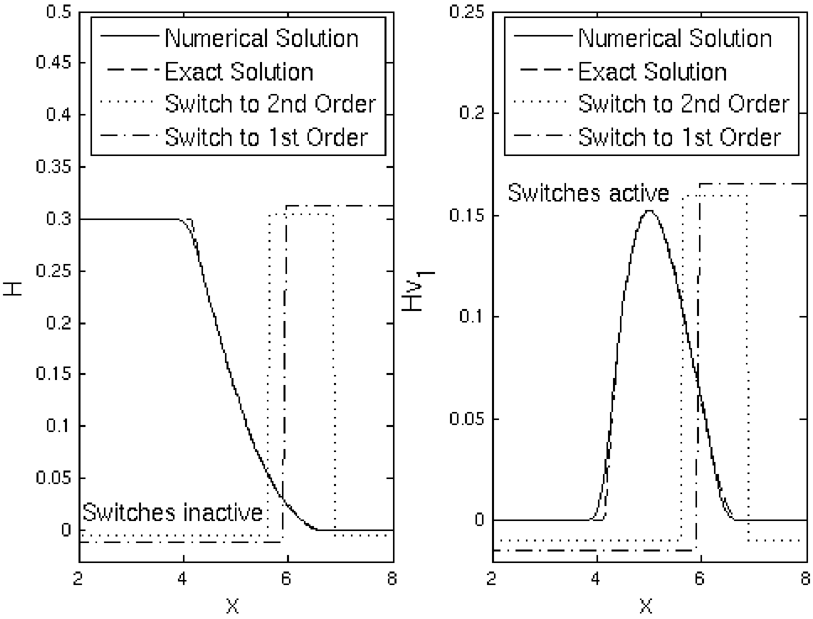

Figure 3.2: Solution of the Riemann Problem with a right dry initial state, second order with reduction to first order if necessary.

of the plot where both lines are at the bottom, the order of the scheme is not restricted. The regions where the dotted line is elevated were considered critical due to $OI(u_{i;0}^n)$ and the first reduction of the order to two took place. The regions where additionally the dash-dotted line is elevated are those where the second switch became active and the order was reduced from two to one. The regions where only the dash-dotted line is elevated were dry at the beginning of the time step that is depicted, and the order of the scheme was set to one even before the reconstruction was carried out to save computing time. The figures show a one dimensional cut of the result of a two dimensional computation. A good agreement of the numerical and the exact solution can be observed, also with respect to the progress of the wet/dry front.

In the figures 3.4 to 3.6, the numerical and the exact solution of the Riemann problem (1.27) with the initial values

$$\mathbf{u}_L = \begin{pmatrix} 0.1\mathrm{g} \\ -0.3\mathrm{g} \\ 0 \end{pmatrix}, \qquad \mathbf{u}_R = \begin{pmatrix} 0.1\mathrm{g} \\ 0.3\mathrm{g} \\ 0 \end{pmatrix}$$

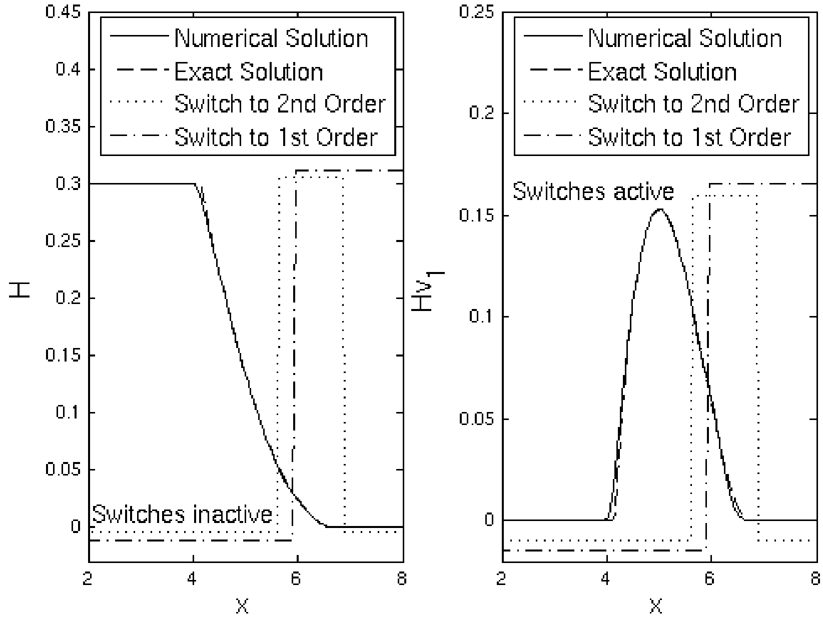

Figure 3.3: Solution of the Riemann Problem with a right dry initial state, third order with reduction to second and first order if necessary.

after 1s are depicted. The exact solution develops a wet zone, as

$$RL(0) = v_{L;1} + 2c_L \approx -0.96 < 0.96 \approx v_{R;1} - 2c_R = RR(0).$$

The figures show a one dimensional cut of the result of a two dimensional computation. Though the reduction to first order is clearly apparent, generally a good agreement of the numerical and the exact solution can be observed.

3.6 The Oscillating Lake

An interesting test case that covers the combination of topography and dry bed states is a lake with a periodically oscillating surface that was suggested

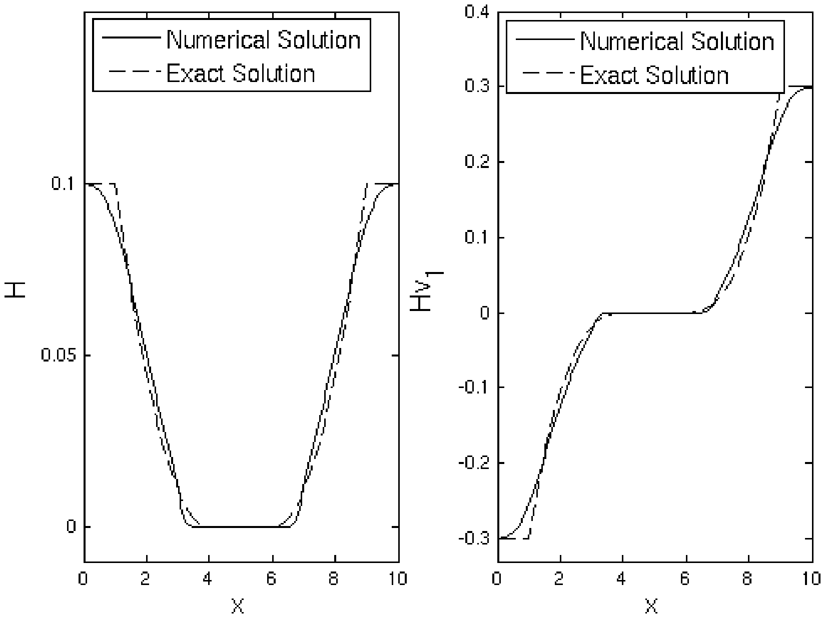

Figure 3.4: Solution of the Riemann Problem with evolution of a dry zone, completely first order.

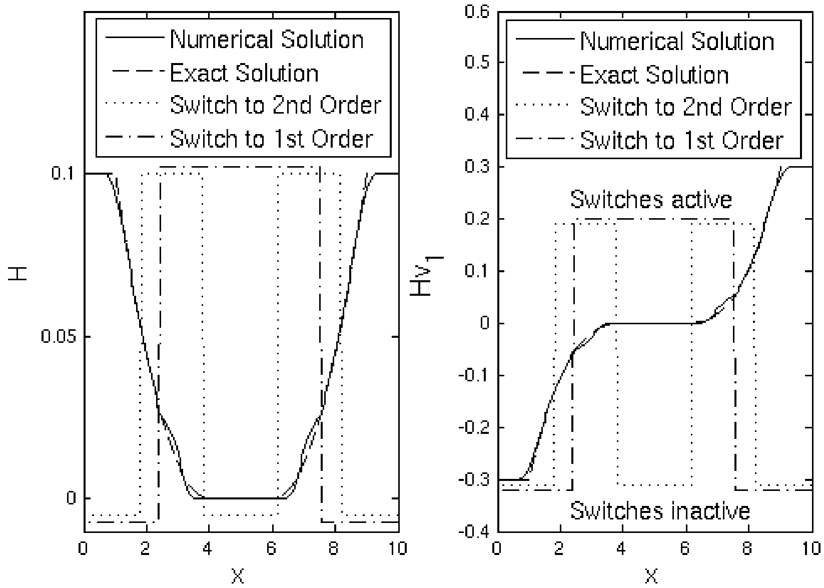

Figure 3.5: Solution of the Riemann Problem with evolution of a dry zone, second order with reduction to first order if necessary.

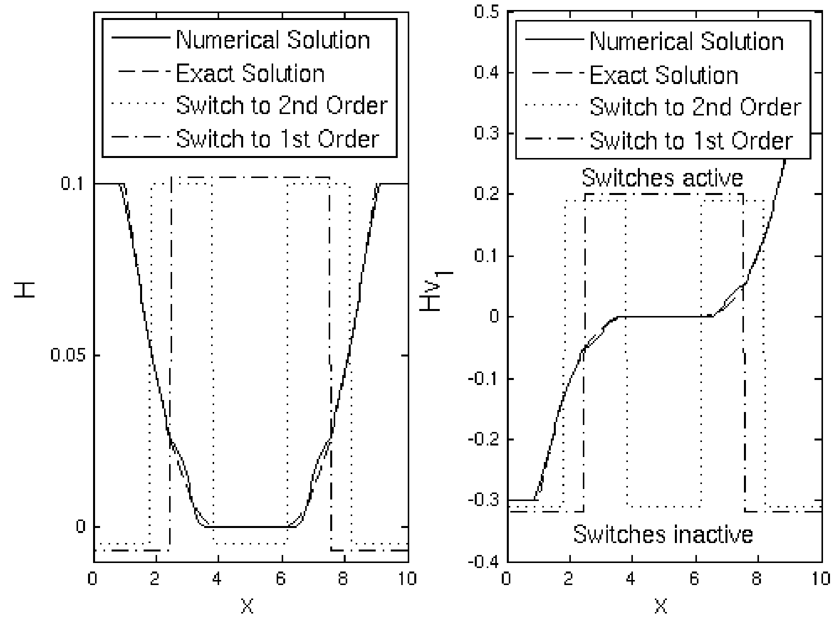

Figure 3.6: Solution of the Riemann Problem with evolution of a dry zone, third order with reduction to second and first order if necessary.

in [GPC07]. The problem consists of the equations

$$\begin{aligned}\Phi(\mathbf{x},t) =& \max(0, \mathrm{g}(\frac{dh_0}{a^2}(2(x_1-2.5)\cos(\frac{\sqrt{2\mathrm{g}h_0}}{a}t)\\ &+2(x_2-2.5)\sin(\frac{\sqrt{2\mathrm{g}h_0}}{a}t)-d)-top(\mathbf{x})))\\ \mathbf{v}(\mathbf{x},t) =& \begin{pmatrix} -d\sqrt{2\mathrm{g}h_0}\sin(\frac{\sqrt{2\mathrm{g}h_0}}{a}t) \\ d\sqrt{2\mathrm{g}h_0}\cos(\frac{\sqrt{2\mathrm{g}h_0}}{a}t)\end{pmatrix}\\ top(\mathbf{x}) =& -h_0\left(1-\frac{(x_1-2.5)^2+(x_2-2.5)^2}{a^2}\right).\end{aligned}$$

The test was carried out on the domain $\Omega = [0.0, 5.0] \times [0.0, 5.0]$ on a grid with 76,234 cells. The parameters were set to $a = 1, d = 0.5, h_0 = 0.2$. The oscillation period is $T = \frac{2\pi}{\sqrt{0.4\mathrm{g}}} \approx 3.17s$.

The figures 3.7 to 3.9 show the results of the computations of first up to third order. A considerable improvement of the quality of the solution can be observed in particular between the figure 3.7 and 3.8 due to the increase of the order of the scheme from one to two. Further increasing of the order

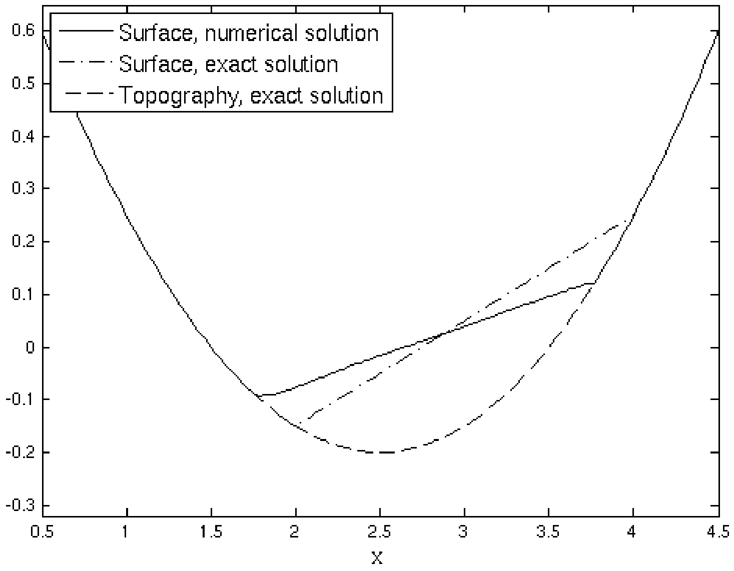

Figure 3.7: Oscillating lake at $t = 2T$, completely first order.

has only small effect. The reason is probably that the region of the solution where the order is reduced is large compared to the region that has the higher prescribed order and that this region moves over almost the whole integration area during the computation.

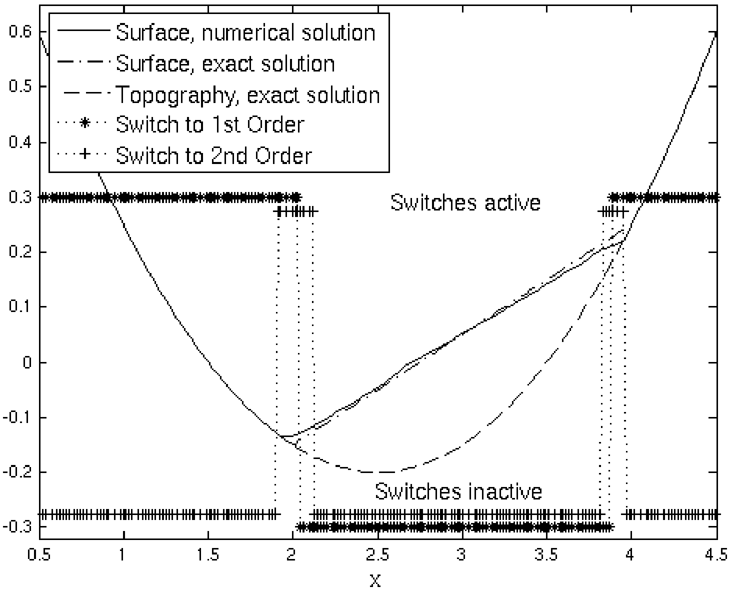

Figure 3.8: Oscillating lake at $t = 2T$, second order with reduction to first order if necessary.

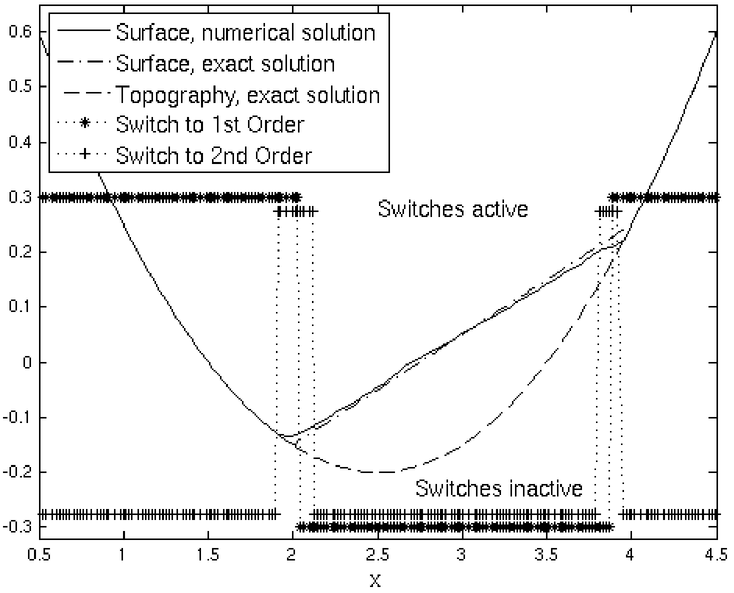

Figure 3.9: Oscillating lake at $t = 2T$, third order with reduction to second and first order if necessary.

3.7 Dam Break with Dry Zones and Obstacles

This test again was proposed in [GPC07]. The computational domain is a channel of 75 m length and 30 m width with three cone-shaped obstacles that is surrounded by fixed walls. The computation is carried out on a grid with 10,712 cells. The initial conditions are given by

$$\begin{aligned}
\Phi(\mathbf{x},0) =& \begin{cases} 1.875\mathrm{g} & \text{if} \quad x_1 \leq 16 \\ 0 & \text{else} \end{cases} \\
\mathbf{v}(\mathbf{x},0) =& \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\
\mathit{top}(\mathbf{x}) =& \max(0, c_1(\mathbf{x}), c_2(\mathbf{x}), c_3(\mathbf{x})) \\
c_1(\mathbf{x}) =& 1 - 0.1\sqrt{(x_1-30)^2 + (x_2-22.5)^2} \\
c_2(\mathbf{x}) =& 1 - 0.1\sqrt{(x_1-30)^2 + (x_2-7.5)^2} \\
c_2(\mathbf{x}) =& 2.8 - 0.28\sqrt{(x_1-47.5)^2 + (x_2-15)^2}
\end{aligned}$$

The figures 3.10 to 3.15 show the numerical solution of the given dam break problem at the time levels $t = 10, 15, 20, 25, 30, 35$. The surface of the three dimensional plots depicts the computed surface of the water in the channel which is the sum of the computed water height and the bottom elevation. The coloring represents the water height in all plots. There is no reference solution, but the results are consistent to the topography and in accordance with those obtained in [GPC07]. The increase of the sharpness of the resolution with increasing order of the computation can be observed quite clearly in the region behind the two small cones. The wave interaction that takes place here is smeared beyond recognition in the figures 3.10 and 3.11, but can be clearly seen in the figures 3.12 and 3.13. It gains in sharpness further in the figures 3.14 and 3.15.

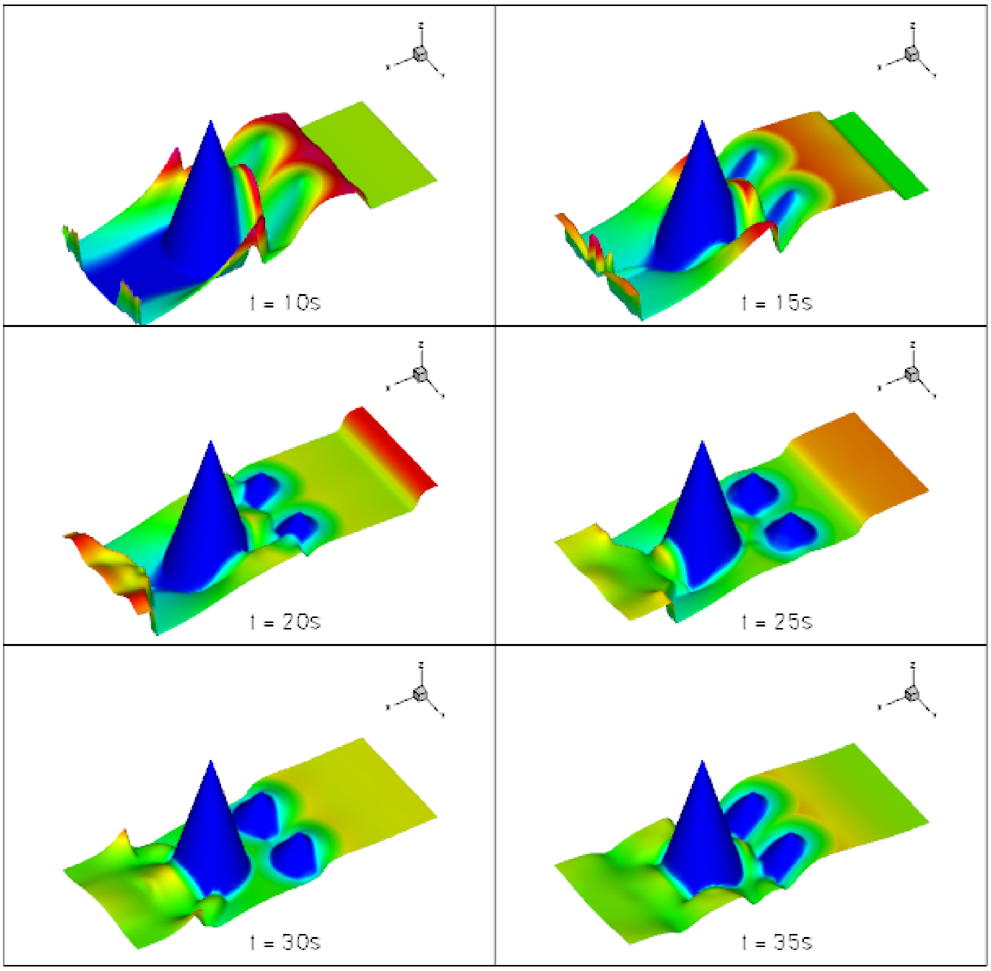

Figure 3.10: Dam break in a canal with cone-shaped obstacles at the times $t = 10, 15, 20, 25, 30, 35$, first order.

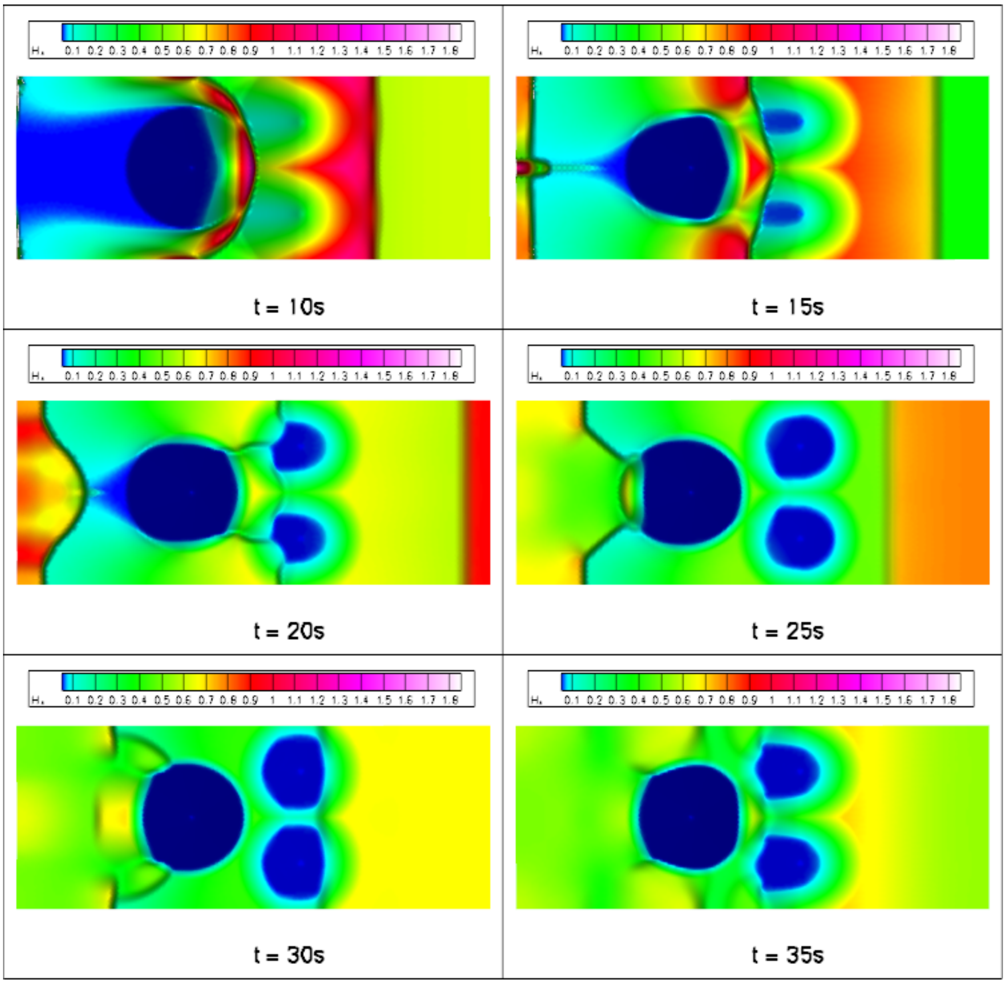

Figure 3.11: Top view of a dam break in a canal with cone-shaped obstacles at the times $t = 10, 15, 20, 25, 30, 35$, first order.

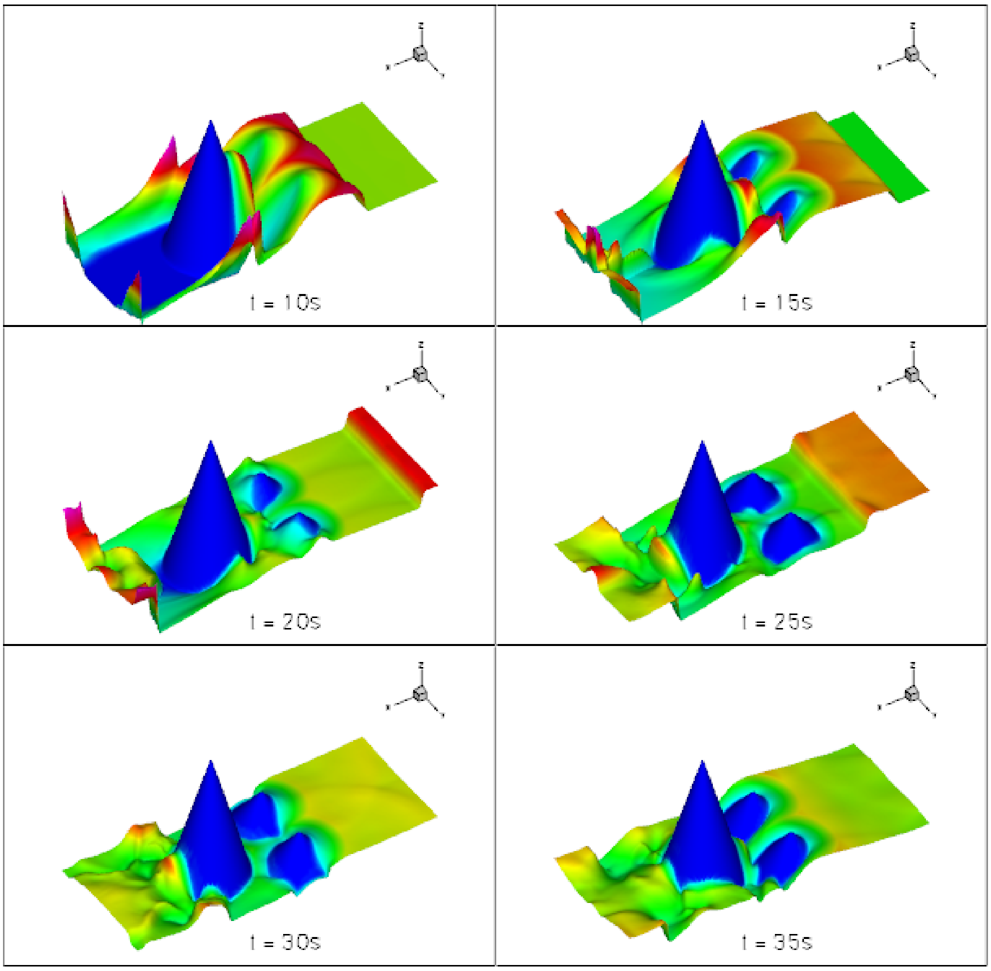

Figure 3.12: Dam break in a canal with cone-shaped obstacles at the times $t = 10, 15, 20, 25, 30, 35$, second order with reduction of order if necessary.

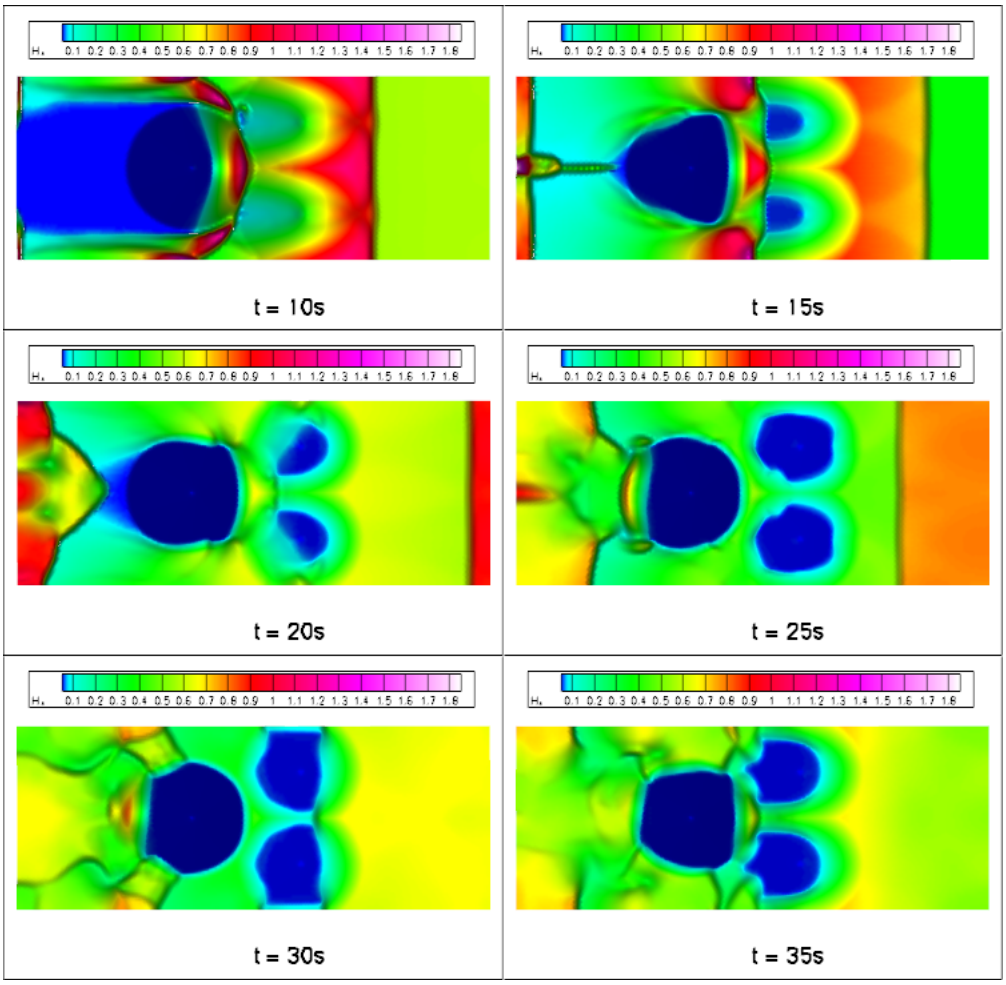

Figure 3.13: Top view of a dam break in a canal with cone-shaped obstacles at the times $t = 10, 15, 20, 25, 30, 35$, second order with reduction of order if necessary.

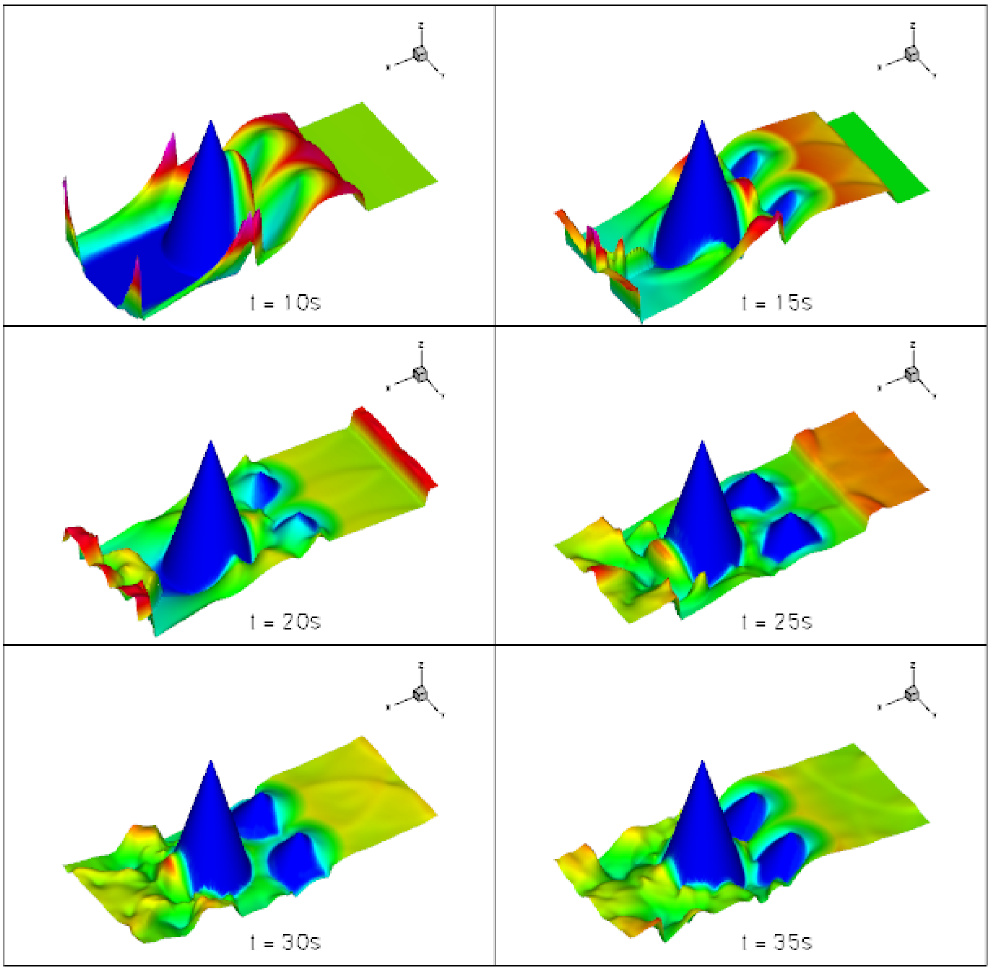

Figure 3.14: Dam break in a canal with cone-shaped obstacles at the times $t = 10, 15, 20, 25, 30, 35$, third order with reduction of order if necessary.

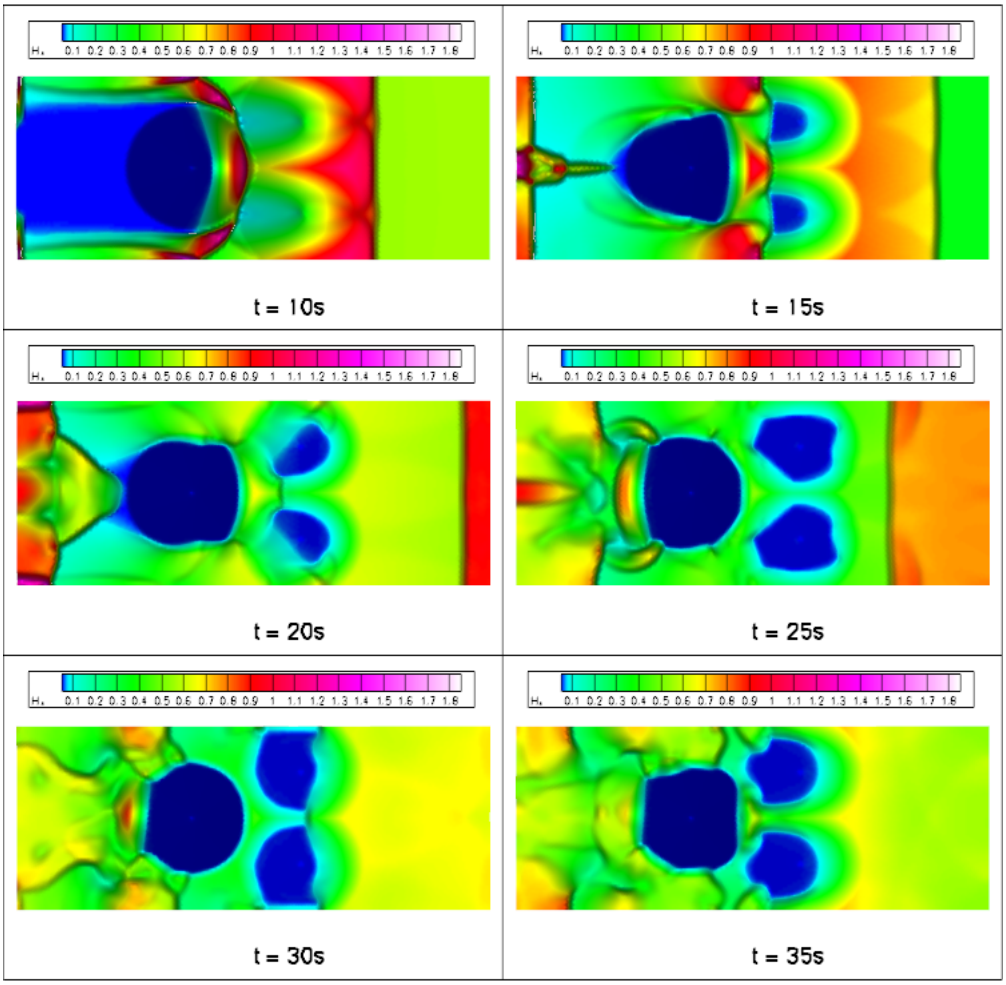

Figure 3.15: Top view of a dam break in a canal with cone-shaped obstacles at the times $t = 10, 15, 20, 25, 30, 35$, third order with reduction of order if necessary.

3.8 Water Flowing Down a Winding Channel

The setup of this test case represents the flowing of water from a reservoir into a channel that is winding down a hillside. The computational domain $\Omega = [0,75] \times [0,30]$ is open at the right boundary and is surrounded by impermeable walls elsewhere. The topography consists of a flat plane that

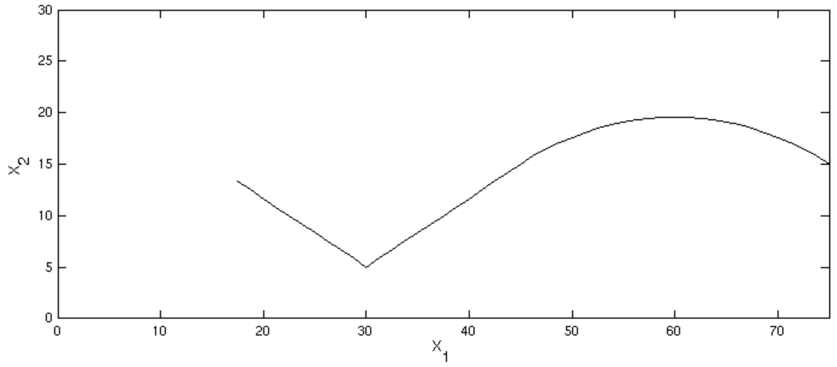

Figure 3.16: Function $a(x_1)$ that was used to create the topography of the channel.

later holds the water and the channel. The channel is defined by the function $a(x_1)$ that is depicted in figure 3.16. For each set $x_p \times [0,30]$ and every position $x_p \in [15,75]$ of the x_1-axis, it holds that that the apex of the cross section is at

$$(x_p, a(x_p), -0.25(x_p - 15)).$$

To the left and to the right, branches of parabolas that are continuous in $a(x_p)$ and have the value $15 - 0.25(x_p - 15)$ at the lower and upper boundary of Ω form the walls of the channel.

The computation was carried out on a grid with 10,712 cells. The initial conditions are

$$\Phi(\mathbf{x},0) = \begin{cases} 14.5 & \text{if } x_1 < 15 \\ 0 & \text{else} \end{cases}$$

$$\mathbf{v}(\mathbf{x},0) = \begin{pmatrix} 0 \\ \\ 0 \end{pmatrix}$$

$$a(x_1) = \begin{cases} -\frac{2}{3}x + 25 & \text{if } 15 \leq x_1 < 30 \\ \frac{2}{3}x - 15 & \text{if } 30 \leq x_1 < 45 \\ \sqrt{731.25 - (x-60)^2} + 37.5 & \text{if } 45 \leq x_1 \leq 75 \end{cases}$$

$$chan(\mathbf{x}) = \begin{cases} 15\left(\frac{x_2 - a(x_1)}{a(x_1)}\right)^2 - 0.25(x_1 - 15) & \text{if } 0 \leq x_2 < a(x_1) \\ 15\left(\frac{x_2 - a(x_1)}{30 - a(x_1)}\right)^2 - 0.25(x_1 - 15) & \text{else} \end{cases}$$

$$top(\mathbf{x}) = \begin{cases} 0 & \text{if } x_1 < 15 \\ chan(\mathbf{x}) & \text{else} \end{cases}$$

The results of the computation that are depicted in the figures 3.17 to 3.22 seem plausible and in accordance with the topography. Due to the parabolic opening of the reservoir to the channel the resulting flow rates Hv_1, Hv_2 lead to small shock waves in the reservoir. A close up view of the reservoir showing the evolution of these waves in the beginning of the computation is depicted in figure 3.23. These small waves vanish through smearing in the first order computation, as can be seen in figure 3.17. The second and third order computations conserve these waves, but in a film that shows a sequence of the pictures, it can be seen that only the third order computation shows the traveling of these waves down the channel.

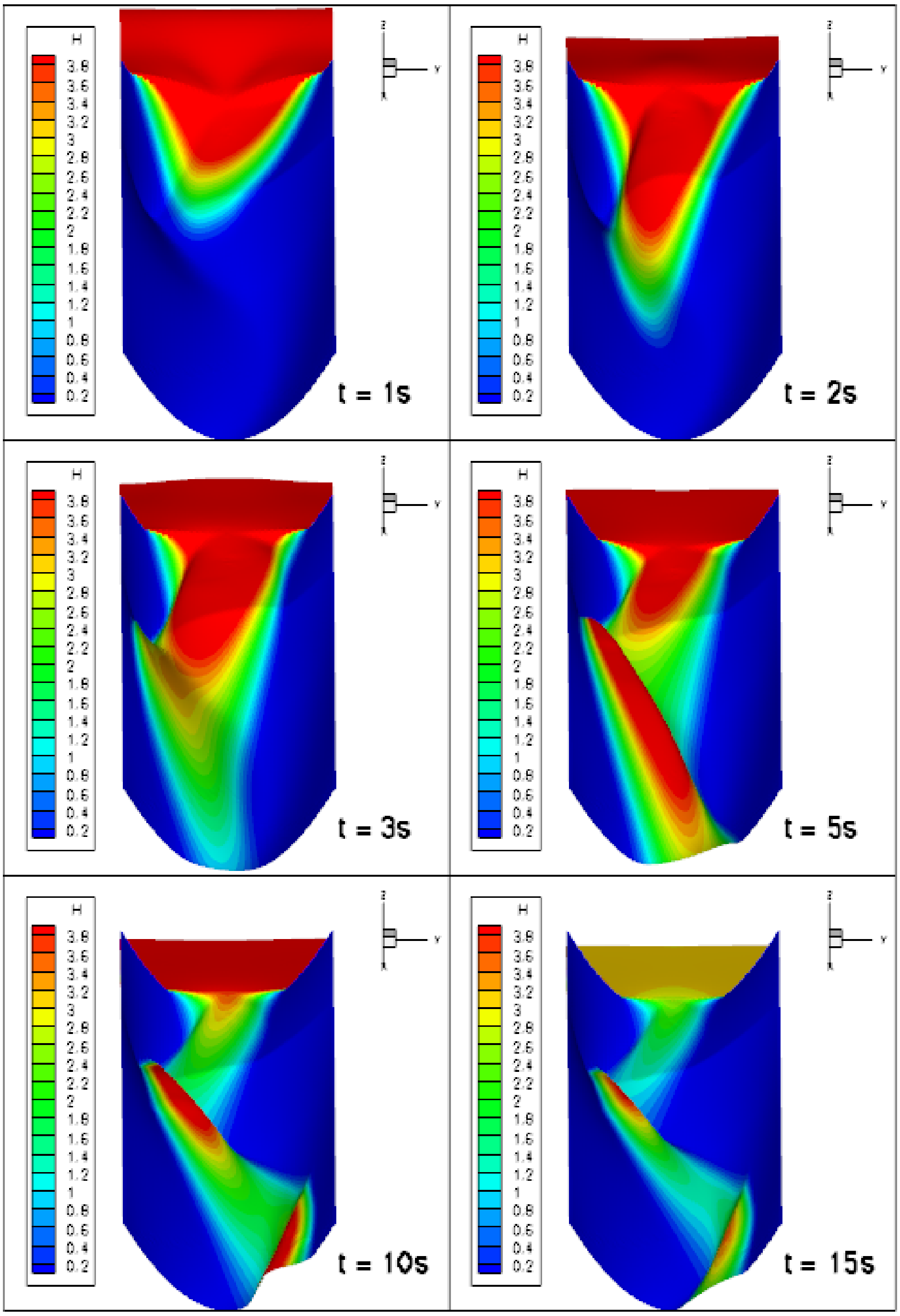

Figure 3.17: Water flowing down a winding channel at the times $t = 1, 2, 3, 5, 10, 15$, first order.

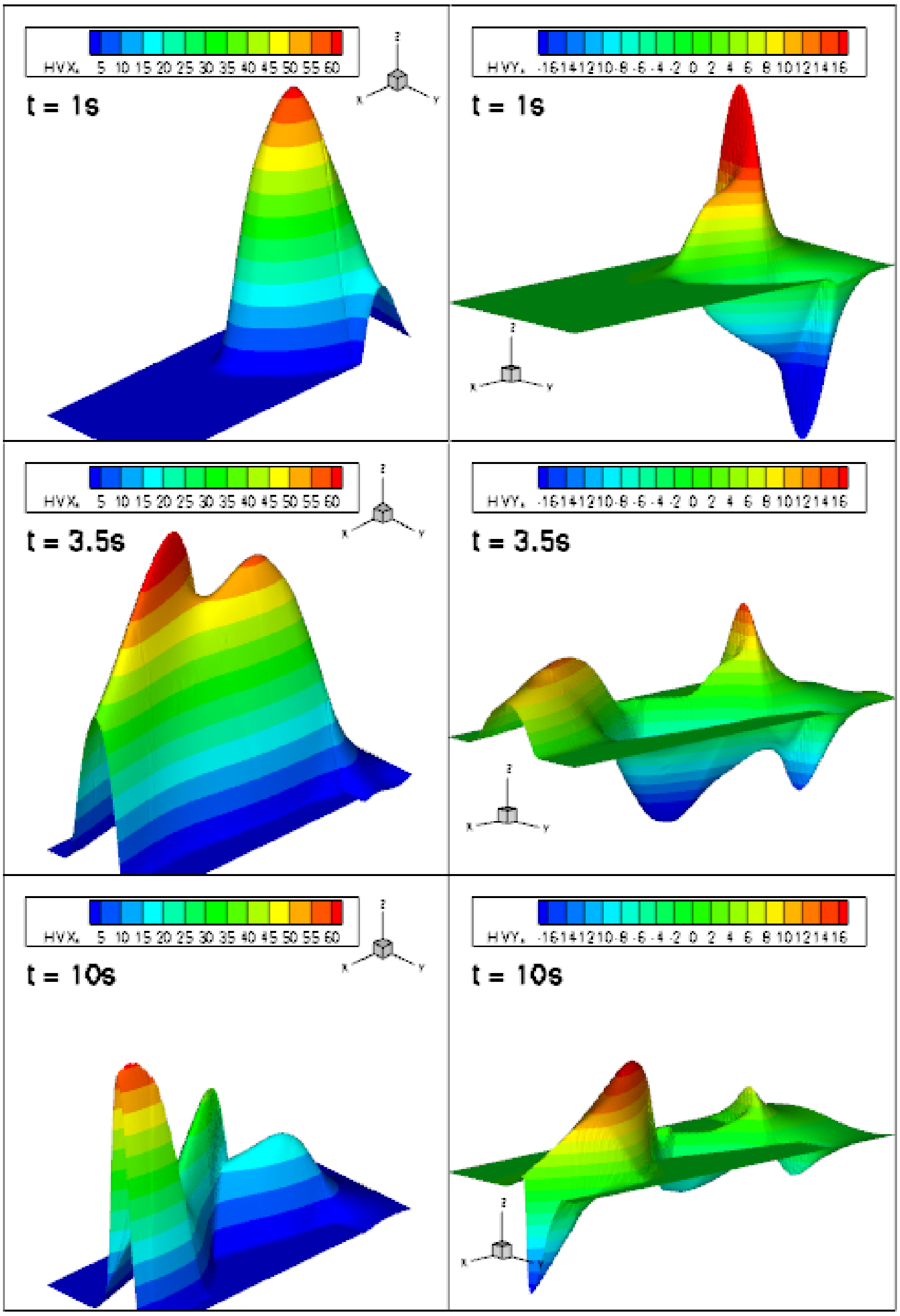

Figure 3.18: Flow rates Hv_1, Hv_2 of water flowing down a winding channel at the times $t = 1, 3.5, 10$, first order.

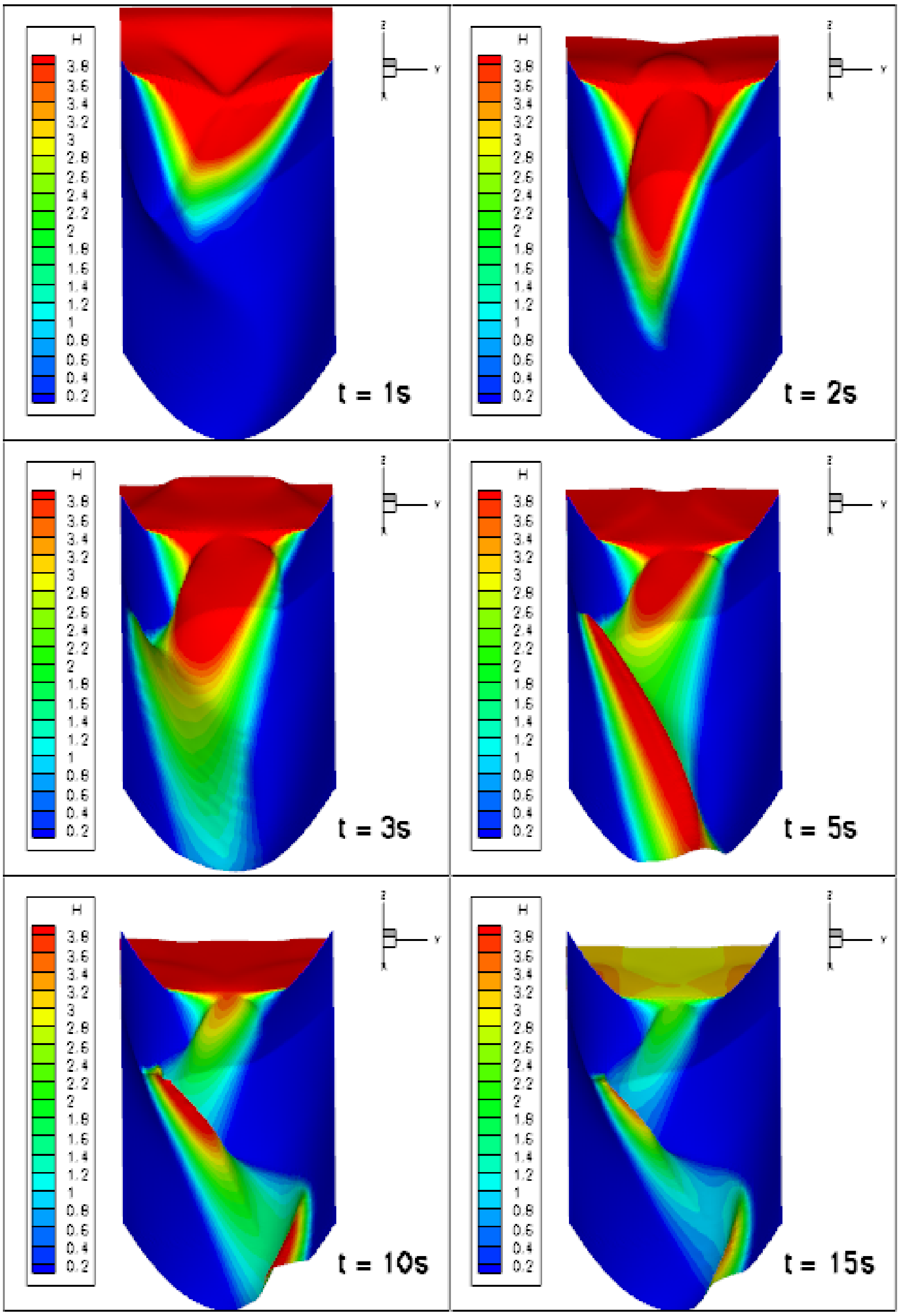

Figure 3.19: Water flowing down a winding channel at the times $t = 1, 2, 3, 5, 10, 15$, second order with reduction to first order if necessary.

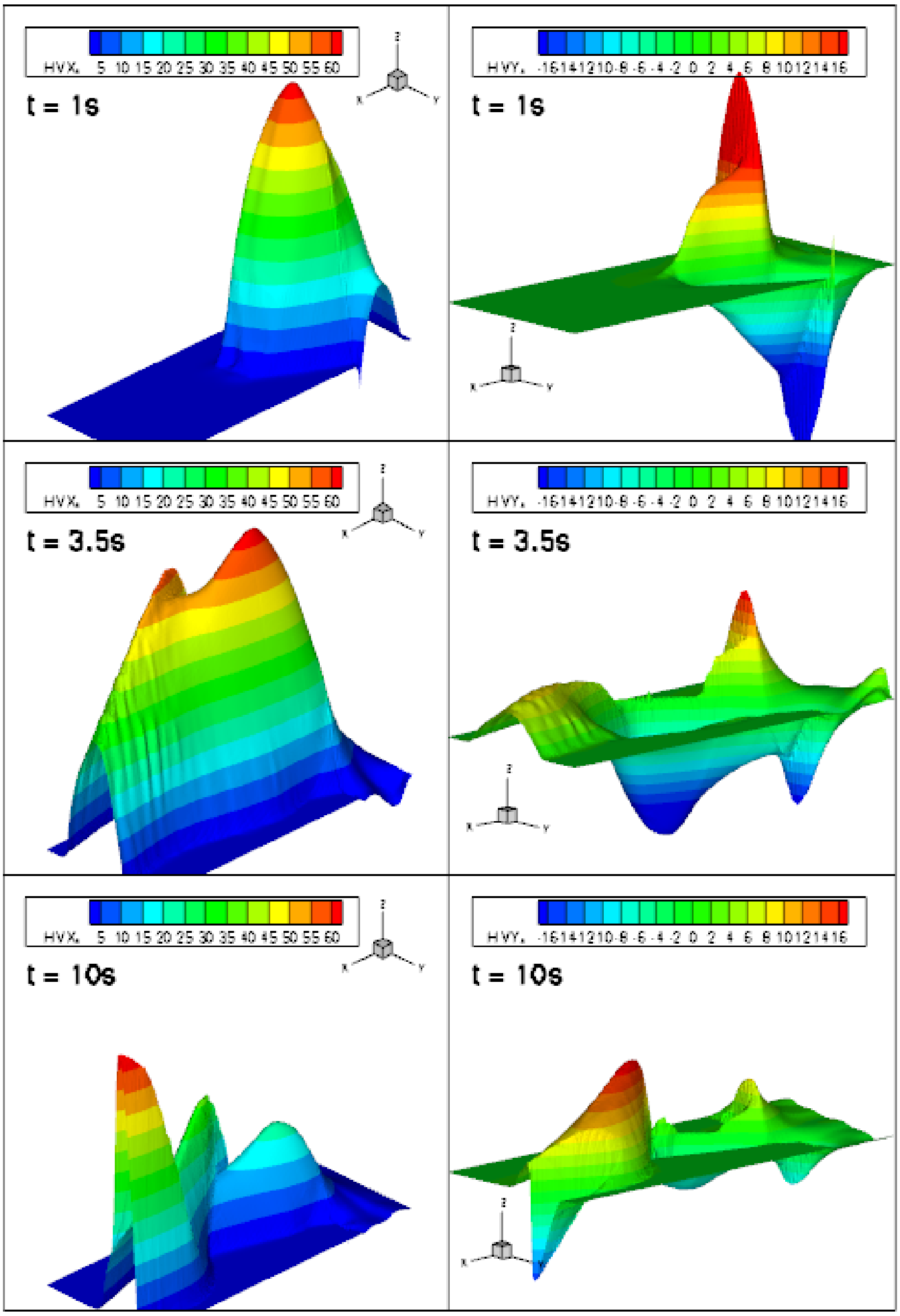
HVX: 5 10 15 20 25 30 35 40 45 50 55 60
t = 1s
HVY: -16 -14 -12 -10 -8 -6 -4 -2 0 2 4 6 8 10 12 14 16
t = 1s
HVX: 5 10 15 20 25 30 35 40 45 50 55 60
t = 3.5s
HVY: -16 -14 -12 -10 -8 -6 -4 -2 0 2 4 6 8 10 12 14 16
t = 3.5s
HVX: 5 10 15 20 25 30 35 40 45 50 55 60
t = 10s
HVY: -16 -14 -12 -10 -8 -6 -4 -2 0 2 4 6 8 10 12 14 16
t = 10s
x
y
z

Figure 3.20: Flow rates Hv_1, Hv_2 of water flowing down a winding channel at the times $t = 1, 3.5, 10$, second order with reduction to first order if necessary.

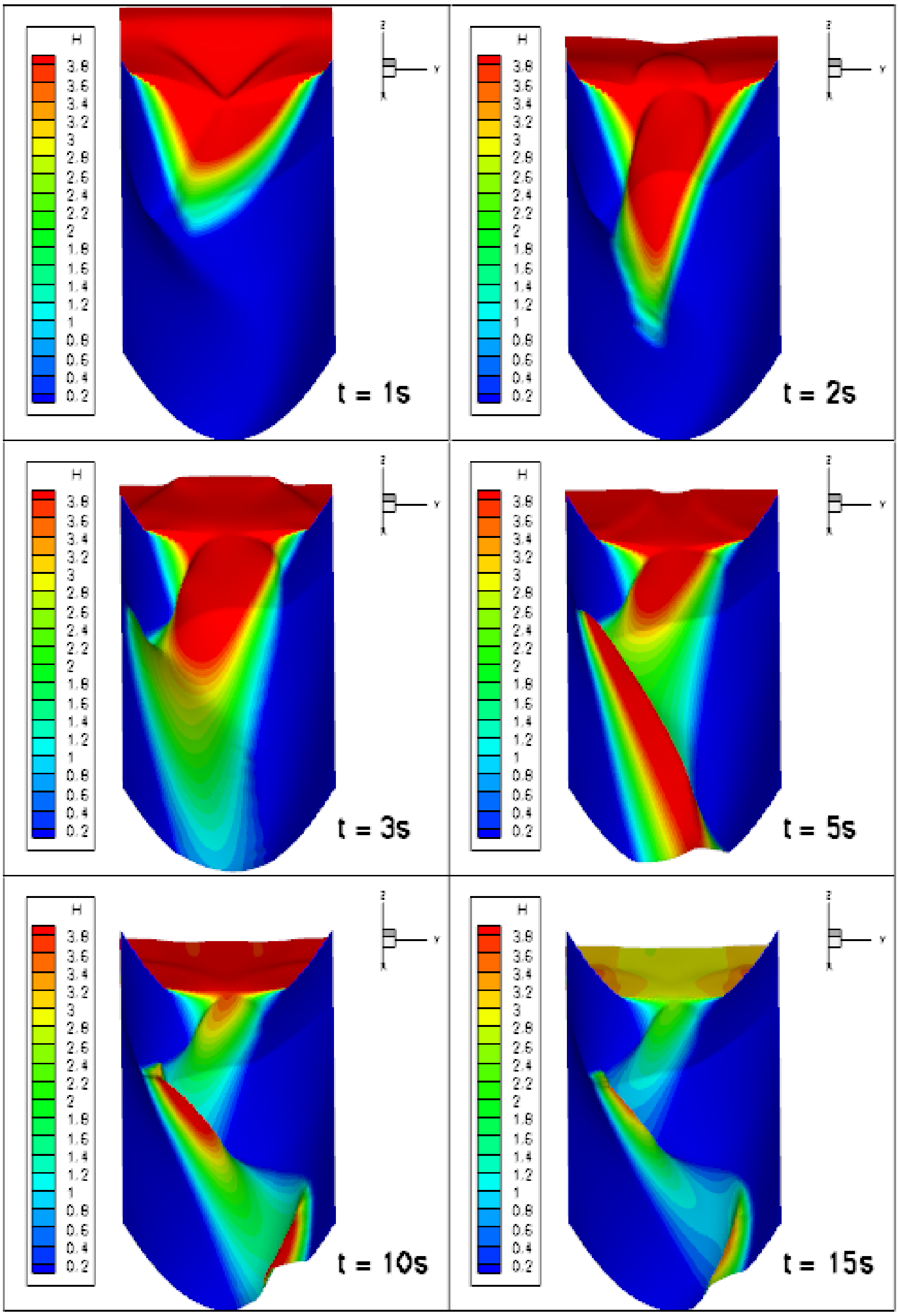

Figure 3.21: Water flowing down a winding channel at the times $t = 1, 2, 3, 5, 10, 15$, third order with reduction to second and first order if necessary.

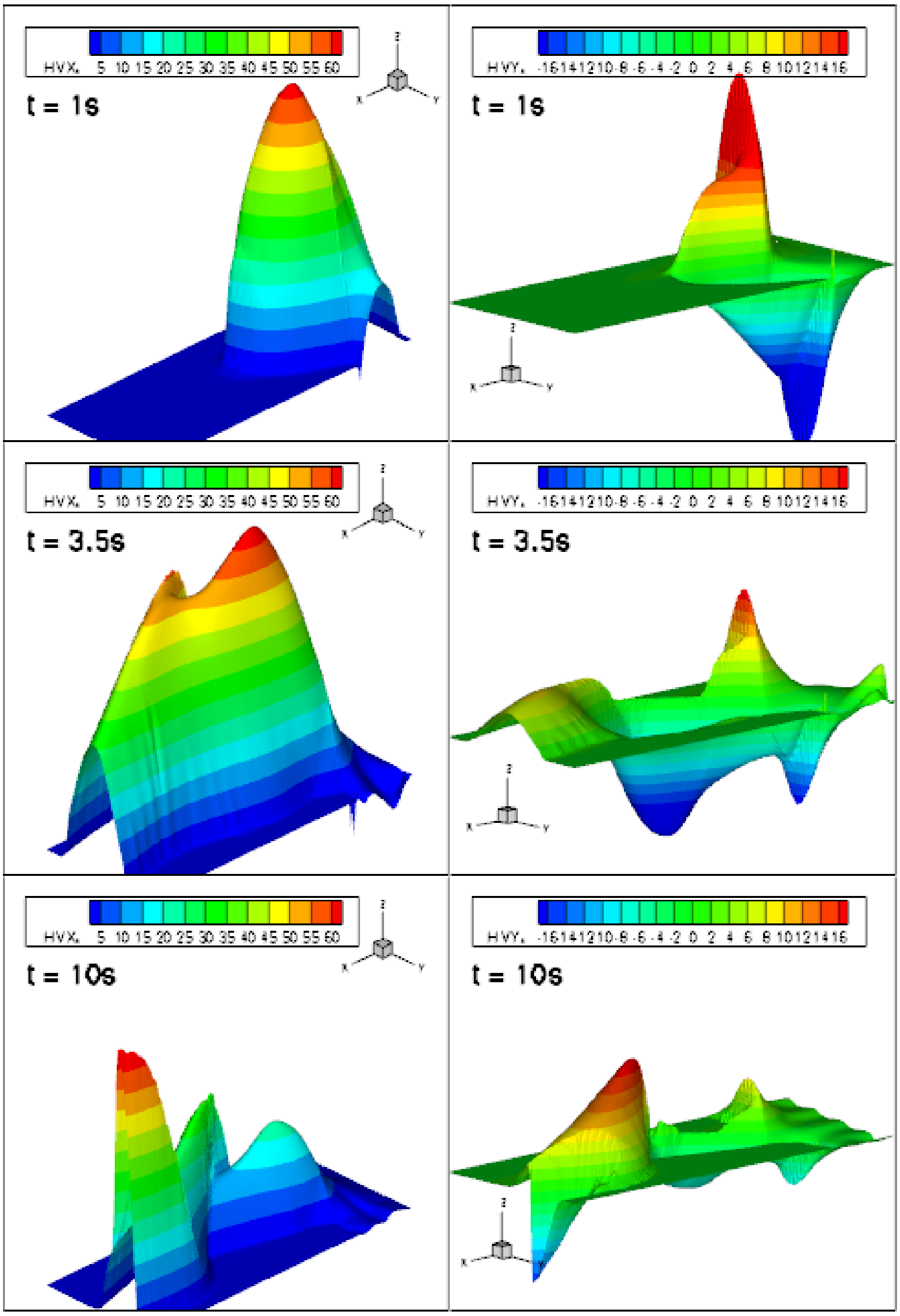

Figure 3.22: Flow rates Hv_1, Hv_2 of water flowing down a winding channel at the times $t = 1, 3.5, 10$, third order with reduction to second and first order if necessary.

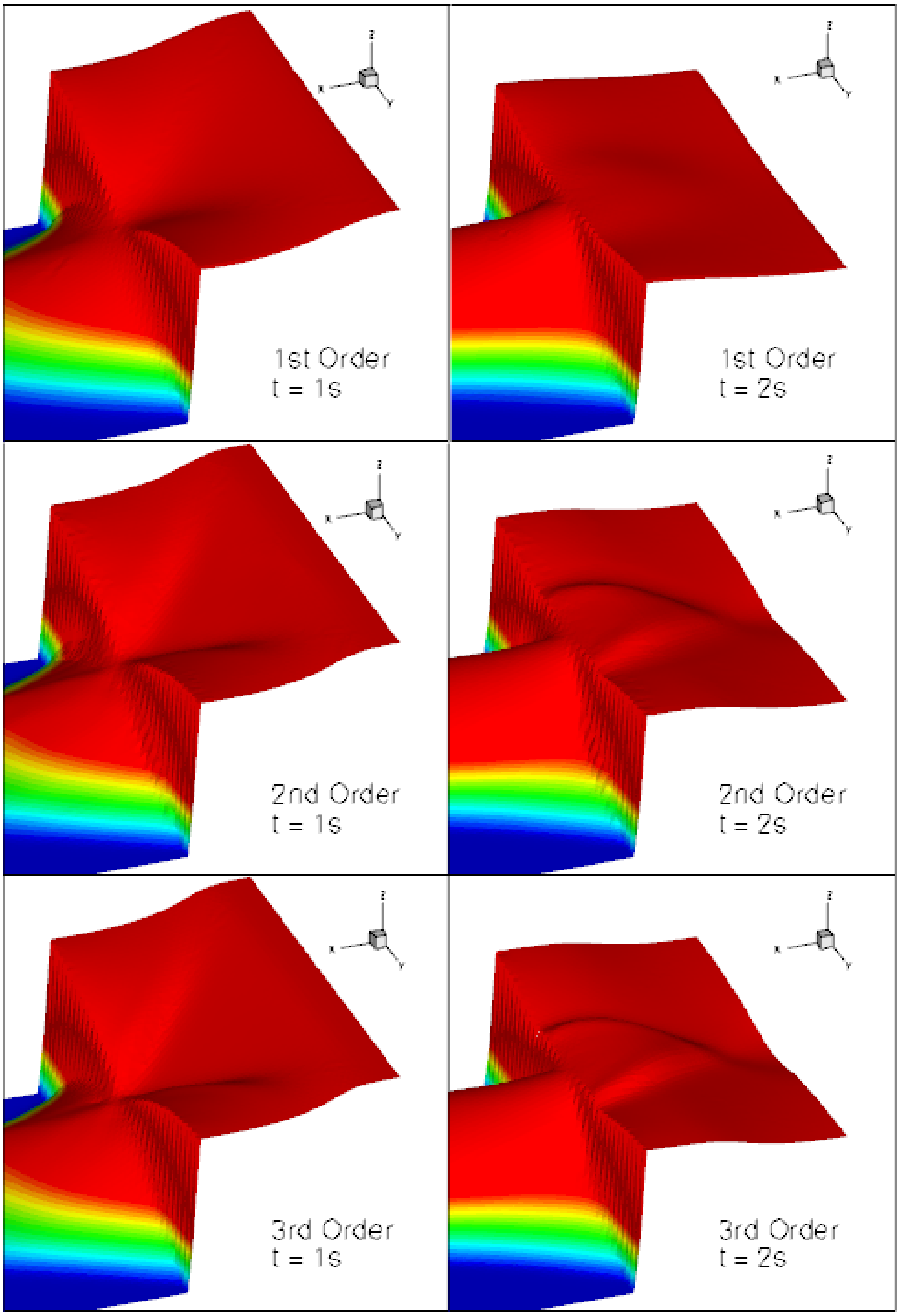

Figure 3.23: Close up view of the evolution of the shock waves in the reservoir for the first, second and third order computation at the times $t = 1, 2$.

Chapter 4

Summary and Prospects

In this thesis, a numerical scheme for the two dimensional shallow water equations was presented that is of arbitrary high order in sufficiently wet, smooth regions of the solution. The scheme can cope with source terms induced by topography, and with dry bed regions. Moreover, it is well balanced for still water steady states.

Due to its properties, the scheme developed in this work can be used for simulations in the context of, for example, ebb and flow. It could be used to investigate the influence of newly planned wind farms on the tidal currents in the Wadden Sea. Another application are artificial modifications of riverbeds in urban development and landscape building. Especially the prediction of the behavior during seasonal flash floods is an interesting point.

The similarity of these fields is that they are all characterized by a combination of the existence of water of a certain depth where waves and currents can be computed with a high order of accuracy, of influences of the topography that are important for the behavior of those currents and waves, and of an area where wetting, dry falling or very shallow water may occur, depending on the currents and waves. In contrast to other schemes, the use of unstructured grids in this scheme allows to consider complicated geometries to be prescribed in the grid, which otherwise might be difficult to describe in terms of topography.

In the course of developing the scheme, a least squares WENO method was described that makes use of the minimizing problem and that provides a set of reconstruction polynomials. It was possible to prove an important approximation property for the coefficients of the polynomials obtained by this method that is the basis of many results further obtained in this work. The approximation order of the spatial reconstruction polynomials follows from this result. Moreover, using the result concerning the coefficients of the spatial polynomial and the Cauchy-Kovalewskaja procedure, for the coefficients of the space time Taylor polynomials a similar result be could proven. Again the approximation order, this time of the space time Taylor polynomial, follows.

The topography including Riemann solver of Chinnayya, Le Roux and Seguin was introduced. Using the latter, under the condition of an adapted reconstruction and Cauchy-Kovalewskaja procedure, the well balanced-ness of the scheme was proven for arbitrary high numerical order.

For the treatment of wet/dry fronts and states containing very shallow water, a method to preserve a positive water height throughout the flux computation was developed. This method consists of a cell-wise reduction of the order of the scheme in two steps. Indicators when to reduce the order depending on the water height, the size of the cell and the expected oscillation of the polynomial on the cell were given.

Several numerical test were accomplished to verify these properties. The quality of the spatial reconstruction was verified numerically up to fourth order.

The full scheme was validated for the two dimensional linear advection equation using a simple (but exact) numerical upwind flux, and for the two dimensional shallow water equations using the exact source terms and the HLLC Riemann solver. In both cases, the numerical matching the theoretical order was demonstrated up to fourth order.

The well balanced-ness of the scheme using the topography Riemann solver was documented for a still water steady state problem. Again, the computation was carried out for a numerical order up to four.

It was numerically verified that the treatment of wet/dry fronts copes with the problem of preserving their correct speed in the case of wetting a former dry area as well as in the case of the dry falling of a former wet area.

As a last validation problem, the test case of an oscillating lake for which the analytical solution is available was computed. This problem contains the treatment of topography as well as wetting and dry falling of areas. The results of this computation for different orders of accuracy were shown.

Finally, two applications of the scheme to more complex geometries were given. Firstly, a dam break with a topography consisting of three cone shaped obstacles was computed using the schemes of first to third order. The resulting figures show very nicely that the schemes of second and third order conserve traveling waves much better than the first order scheme.

Secondly, the emptying of a water reservoir in a channel that represents a riverbed winding down a hill was computed. Again, great improvements in the conservation of waves, and thus in the exactness of the obtained results, can be observed.

At present, the computation of the time step is not fit for the requirements of the scheme. The evolution of the velocity accounted for in the numerical flux due to the space time expansion is not taken into account yet but was covered in the computations by a smaller CFL-number. Moreover, for the topography Riemann solver exists an extension that allows the bottom friction to be taken into account by considering a roughness coefficient for the bottom for each cell. This extension is not implemented yet in the presented scheme but it seems to be a very interesting feature.

To sum up, for the scheme developed in this work it can be stated that the results obtained by several computations that are documented in chapter 3 are very promising. Especially the computation of the dam break with obstacles and the emptying of a water reservoir show the scheme's capability in order to simulate complex problems.

Bibliography

[Abg94] R. Abgrall: *On Essentially Non-oscillatory Schemes on Unstructured Meshes: Analysis and Implementation.* J. Comput. Phys. **114** (1994), pp 45-58.

[AuB05] E. Audusse, M.-O. Bristeau: *A well-balanded positivity preserving 'second-order' scheme for shallow water flows on unstructured meshes.* J. Comput. Phys., **206** (2005), pp 311-333.

[BCK98] M. van Berg, O. Cheong, M. van Kreveld, M. Overmars: *Computational Geometry. Algorithms and Applications.* Springer, 3rd Edition (2008).

[Bjö96] Å. Björck: *Numerical Methods for Least Squares Problems.* SIAM (1996).

[BoM10] F. Bouchut, T. Morales de Luna: *A subsonic-sell-balanced reconstruction scheme for shallow water flows.* SIAM J. Numer. Anal. **48** No. 5 (2010), pp 1733-1758.

[CGL08] M. Castro, J.M. Gallardo, J.A. López-García, C. Parés: *Well-balanced high order extensions of Godunov's method for semilinear balance laws.* SIAM J. Numer. Anal. **46** No. 2 (2008), pp 1012-1039.

[ChL99] A. Chinnayya, A.-Y. LeRoux: *A new general Riemann Solver for the Shallow-Water Equations with Friction and Topography.* http://math.ntu.no/conservation (1999).

[CLS04] A. Chinnayya, A.-Y. LeRoux, N. Seguin: *A well balanced numerical scheme for the approximaton of the shallow-water equations with topography: the resonance phenomenon.* International Journal on Finite Volume Methods, **1** Nr 1 (2004), pp 1-33.

[ChM98] A. J. Chorin, J. E. Marsden: *A Mathematical Introduction to Fluid Mechanics.* Springer, 3rd Edition (1998).

[CoF44] R. Courant, K. O. Friedrichs: *Supersonic flow and shock waves, a manual on the mathematical theory of non-linear wave motion.* (1944).

[Dav88] S. F. Davis: *Simplified Second-Order Godunov-Type Methods.* SIAM J. Sci. Stat. Comput. **9**, pp 445-473 (1988).

[Del34] B. Delaunay: *Sur la Sphère Vide.* Bulletin de l'Académie des Sciences de l'URSS **6** (1934), pp 793-800.

[DuK07] M. Dumbser, M. Käser: *Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems.* J. Comput. Phys., **221** Issue 2, pp 693-723 (2007).

[DKT07] M. Dumbser, M. Käser, V. A. Titarev, E. F. Toro: *Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems.* J. Comput. Phys., **226** Nr 1 (2007), pp 204-243.

[DuM07] M. Dumbser, C.-D. Munz: *On source terms and boundary conditions using arbitrary high order discontinuous galerkin schemes.* Int. J. Appl. Math. Comput. Sci. **17** No. 3 (2007), pp 297-310.

[Dys01] R. W. Dyson: *Technique for Very High Order Nonlinear Simulation and Validation.* NASA/TM–2001-210985 (2001).

[Fri98] O. Friedrich: *Weighted Essentially Non-Oscillatory Schemes for the Interpolation of Mean Values on Unstructured Grids.* J. Comput. Phys. **144**, 194-212 (1998).

[Fri99] O.Friedrich: *Gewichtete wesentlich nicht-oszillierende Verfahren auf unstrukturierten Gittern.* Dissertation am Fachbereich Mathematik der Universität Hamburg (1999).

[FrT93] L. Fraccarollo, E. F. Toro: *A Shock-Capturing Method for Two Dimensional Dam-Break Problems.* Proceedings of the fifth international Symposium in Computational Fluid Dynamics, Sendai, Japan, 1993.

[FrT95] L. Fraccarollo, E. F. Toro: *Experimental and Numerical Assessment of the Shallow Water Model for Two-Dimensional Dam-Break Type problems.* J. Hydraul. Res. **33**, pp 843-864 (1995).

[GPC07] J.M. Gallardo, C. Parés, M. Castro: *On a well-balanced high-order finite volume scheme for shallow water equations with topography and dry areas.* J. Comput. Phys. **227** (2007), pp 574-601.

[GLM07] G. Gassner, F. Lörcher, C.-D. Munz: *A discontinuous Galerkin scheme based on a space-time expansion. I. Inviscid compressible flow in one space dimension.* J. Sci. Comput. **32** No 2 (2007), pp 175-199.

[GLM08] G. Gassner, F. Lörcher, C.-D. Munz: *A discontinuous Galerkin scheme based on a space-time expansion. II. Viscous flow equations in multi dimensions.* J. Sci. Comput. **34** No 3 (2008), pp 260-286.

[GoR96] E. Godlewski, P.-A. Raviart: *Numerical Approximation of Hyperbolic Systems of Conservation Laws.* Springer, New York, Berlin, Heidelberg (1996).

[God59] S.K. Godunov: *Finite Difference Methods for the Computation of Discontinuous Solutions of the Equations of Fluid Dynamics.* Math. Sbornik, **47**, pp 271-306 (1959), translated US Joint Publ. Res. Service, JPRS 7226 (1969).

[HaC91] A. Harten, S. R. Chakravarthy: *Multi-Dimensional ENO Schemes for General Geometries.* ICASE Report No. 91-76 (1991).

[HaO87] A. Harten, S. Osher: *Uniformely High-Order Accurate Nonoscillatory Schemes, I.* SIAM J. Num. Anal. **24** (1987), pp 279-309.

[HEO87] A. Harten, B. Engquist, S. Osher, S. R. Chakravarthy: *Uniformly High Order Accurate Essentially Non-oscillatory Schemes, III.* J. Comput. Phys. **71**, 231-303 (1987).

[HLL83] A. Harten, P. D. Lax, B. van Leer: *On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws.* SIAM Review **25** Nr 1 (1983), pp 35-61.

[Hug87] H. Hugoniot: *Sur la Propagation du Mouvement dans les Corps et spécialement dans les Gaz parfaites. Première Partie.* J. École Polytechnique **LVII** (1887), pp 3-97. English translation.

[Hug89] H. Hugoniot: *Sur la Propagation du Mouvement dans les Corps et spécialement dans les Gaz parfaites. Deuxième Partie.* J. École Polytechnique **LVIII** (1889), pp 1-125.

[IsS96] A. Iske, T. Sonar: *On the Structure of Function Spaces in Optimal Recovery of Point Functionals for ENO-Schemes by Radial Basis Function.* Numerische Mathematik **74** 2 (1996), pp 177-202.

[JiS96] G.-S. Jiang, C.-W. Shu: *Efficient Implementation of Weighted ENO Schemes.* J. Sci. Comput. **126** (1996), pp 202-228.

[KäI04] M. Käser, A. Iske: *ADER schemes on adaptive triangular meshes for scalar conservation laws.* J. Comput. Phys. **205** (2005), pp 486-508.

[LaH87] C. L. Lawson, R. J. Hanson: *Solving Least Squares Problems.* SIAM, (1987).

[Lax57] P. D. Lax: *Hyperbolic Systems of Conservation Laws II.* Commun. Pur. Appl. Math. **10** (1957), pp 537-566.

[LaW60] P. Lax, B. Wendroff: *Systems of Conservation Laws.* Commun. Pur. Appl. Math. **13** (1960), pp 217-237.

[Lax73] P. D. Lax: *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves.* Regional Conference Series in Applied Mathematics 11, SIAM, Philadelphia, (1973).

[LeV02] R. J. LeVeque: *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, New York (2002).

[LOC94] X.-D. Liu, S. Osher, T. Chan: *Weighted essentially non-oscillatory schemes.* J. Comput. Phys. **115** (1994), pp 200-212.

[MeS02] A. Meister, J. Struckmeier: *Hyperbolic Partial Differential Equations.* Viehweg, Braunschweig, Wiesbaden (2002).

[MRT05] R. M. M. Mattheij, S. W. Rienstra, J. H. M. ten Thije Boonkkamp: *Partial Differential Equations.* SIAM (2005).

[NPP06] S. Noelle, N. Pankratz, G. Puppo, J.R. Natvig: *Well-balanced finite volume schemes of arbitrary high order of accuracy for shallow flows.* J. Comput. Phys. **213** (2006), pp 474-499.

[NXS07] S. Noelle, Y. Xing, C.-W. Shu: *High-order well-balanced finite volume WENO schemes for shallow water equation with moving water.* J. Comput. Phys. **226** (2007), pp 29-58.

[Ran70] W. J. M. Rankine: *On the Thermodynamic Theory of Waves of Finite Longitudinal Disturbance.* Phil. Trans. R. Soc. Lond. **160** (1870), pp 277-288.

[Rie60] B. Riemann: *Über die Fortpflanzung ebener Luftwellen von endlicher Schwingugnsweite.* Abhandlungen der Königlichen Gesellschaft der Wissenschaften zu Göttingen **8** (1860), pp 43-66.

[Seg99] N. Seguin: *Génération et validation de Rozavel, un code équilibre en hydraulique 2D.* http://www.-gm3.univ-mrs.fr/ leroux/publications/n.seguin.html, 1999.

[ShO88] C.-W. Shu, S. Osher: *Efficient Implementation of Essentially Nonoscillatory Shock-Capturing Schemes.* J. Comput. Phys. **77** (1988), pp 439-471.

[Smo83] J. Smoller: *Shock Waves and Reaction-Diffusion Equations.* Springer, New York, 2nd Edition (1994).

[Son96] T. Sonar: *Optimal Recovery Using Thin Plate Splines in Finite Volume Methods for the Numerical Solution of Hyperbolic Conservation Laws.* IMA J. Num. Anal. **16** (1996), pp 549-581.

[Son97] T. Sonar:*Mehrdimensionale ENO-Verfahren.* Teubner, Stuttgart (1997).

[Sto57] J. J. Stokes:*Water Waves.* Interscience Publishers, Inc., New York (1957).

[TiT02] V. A. Titarev, E. F. Toro: *ADER: Arbitrary High Order Godunov Approach.* J. Sci. Comput. **17** Nos. 1-4 (2002), pp 609-618.

[Tor99] E. F. Toro: *Riemann Solvers and Numerical Methods for Fluid Dynamics.* John Wiley & Sons, Ltd, Chichester (2001).

[Tor01] E. F. Toro: *Shock-capturing methods for free surface shallow flows.* Springer, Berlin Heidelberg (1999).

[TSS94] E. F. Toro, M. Spruce, W. Speares: *Restoration of the Contact Surface in the HLL-Riemann Solver.* Shock Waves, **4** (1994), pp 25-34.

[XiS11] Y. Xing, C.-W. Shu: *High-order finite volume WENO schemes for the shallow water equations with dry states.* Adv. Water Resour. **34** (2011), pp 1026-1038.

Hiermit versichere ich, dass ich die vorliegende Dissertation selbständig und ohne unerlaubte Hilfe angefertigt und andere als die in der Dissertation angegebenen Hilfsmittel nicht benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen sind, habe ich als solche kenntlich gemacht. Kein Teil dieser Arbeit ist in einem anderen Promotions- oder Habilitationsverfahren verwendet worden.