Master’s Thesis in Mathematics

Development of an ADER Discontinuous Galerkin method, with a Finite Volume Limiting approach, for the sam(oa)² framework

Entwicklung eines ADER Discontinuous Galerkin Verfahrens, mit Finite Volumen Limiter Ansatz, für das sam(oa)² Framework.

Author: Leonhard Andreas Rannabauer
Supervisor: Prof. Dr. Hans-Joachim Bungartz
Advisor: Prof. Dr. Michael Bader
M.Sc. Chaulio Ferreira
Dr.rer.nat. Vasco Varduhn

Submission date: December 1st, 2016
Master’s Thesis in Mathematics

Development of an ADER Discontinuous Galerkin method, with a Finite Volume Limiting approach, for the sam(oa)$^2$ framework

Entwicklung eines ADER Discontinuous Galerkin Verfahrens, mit Finite Volumen Limiter Ansatz, für das sam(oa)$^2$ Framework.

Author: Leonhard Andreas Rannabauer
Supervisor: Prof. Dr. Hans-Joachim Bungartz
Advisor: Prof. Dr. Michael Bader
M.Sc. Chaulio Ferreira
Dr.rer.nat. Vasco Varduhn
Submission date: December 1st, 2016
I hereby declare that this thesis is my own work and that no other sources have been used except those clearly indicated and referenced.

Munich, December the 1st, 2016

Leonhard Andreas Rannabauer
Acknowledgments

I would like to express my deepest gratitude to Prof. Michael Bader for giving me the opportunity to work on a such interesting but also challenging topic, for his time, interest and guidance along my thesis.
My thanks also goes to Chaulio Ferreira, who always took the time and patience to sit down with me to help, explain and solve the problems I encountered.
In the end, I want to thank Vasco Varduhn for his advice and shared experience on the mathematical part of this thesis and Oliver Meister for his smooth introduction into sam(oa)$^2$.
I also want to take the chance and thank my parents for their unconditional support along the whole Master’s degree.
Abstract

This thesis encounters the development of an ADER Discontinuous Galerkin method within the sam(oa)² framework for the simulation of Shallow Water Equations. The key element of this scheme is a proposed predictor step, replacing the common methods for time integration. To stabilize the method a new a posteriori limiter approach, using Finite Volume schemes, will be shown. This limiter converts the solution for instabilities between polynomial representation and the representation on a patch of Finite Volume cells. A new set of criteria to detect these instabilities is introduced. They take Physical and Numerical admissibility conditions into account. Additionally a hybrid basis for the polynomial approximation of the solution, the Quasi-Nodal approach, inhabiting characteristics of nodal and modal bases, will be used. The presented concepts will finally be verified by a series of numerical tests.
# Contents

0 Introduction ................................................. 1

1 Preliminaries .............................................. 2  
  1.1 Summary of the sam(oa)² framework .................. 2  
  1.2 Shallow water equations .............................. 3  
  1.3 Geometry in sam(oa)² .................................. 4

2 A Quasi-Nodal Approach ................................. 6

3 The ADER Discontinuous Galerkin Scheme ................. 11  
  3.1 The Predictor Step .................................. 11  
     3.1.1 Divergence of the Predictor Step .............. 14  
  3.2 The Full Scheme ..................................... 15  
  3.3 Well Balanced Property .............................. 17

4 A posteriori Finite Volume Limiter Approach ............. 20  
  4.1 Converting Representations .......................... 21  
     4.1.1 Finite Volume Approximation on Subcells ...... 21  
     4.1.2 Reconstructing the polynomial Representation .. 23  
  4.2 Detection criteria .................................... 25  
     4.2.1 Physical Admissibility Detection Criteria ...... 25  
     4.2.2 Numerical Admissibility Detection Criteria .... 25

5 Realization in sam(oa)² .................................... 28

6 Numerical Analysis ........................................ 30  
  6.1 Two Dimensional Convergence Test .................. 31  
  6.2 Oscillating Lake .................................... 33  
  6.3 Radial Dam Break ................................... 36  
  6.4 Tohoku Tsunami ..................................... 39

7 Conclusion and Future Work .............................. 43

Bibliography .................................................. 44
0 Introduction

During the second world war the first attempts to numerically solve non-linear Partial Differential Equations on a large-scale where made along the Manhattan project. Task was the approximation of the pressure of blast waves arising when detonating TNT around a core of uranium. A native Finite Differences scheme was used to model the problem and evaluated by the first processing units, which where simply a bunch of desks arranged like the spacial discretisation on which the physicist’s wives performed the method’s calculations (As told in [18]). It’s been a long way from these first attempts to petascale simulations today. The desks got replaced by computer clusters, the variety of methods got widely extended and the modeled field itself broadened from military objects to civil applications.

After Several Tsunami catastrophes in the last decade, like the earthquake in the Indian ocean in 2003 where more then 230000 people were killed, eyes were laid on how to reduce the number of victims in future events. One prevention tool is the Tsunami early warning system which uses seismometers to register the position and magnitude of earthquakes in critical areas, as for example between two continental plates (e.g. the GITEWS Project is presented in [23]). A set of possible earthquakes gets simulated in advance and having registered an earthquake the measurement gets compared. An approximate prediction on how high the impact of the Tsunami is going be can be given. Though newer methods got developed, the mathematical problems for these simulations, like numerical accuracy, stability and convergence remain similar to what John von Neumann had to deal with along the Manhattan project. The standard approach models Tsunami waves by using the hyperbolic Shallow Water Equations, which can be numerically solved by Finite Volume methods, as for example in [8, 11, 17], or a Finite Element methods, as in [2, 10, 14].

In this thesis the problem is encountered by using the ADER Discontinuous Galerkin approach, as introduced in [6] by Dumbser et al.. Additionally an a posteriori Finite Volume Limiter, established in [9] also by Dumbser et al., will be implemented to stabilize the method. After preliminaries got shown in the first chapter, the second chapter will present an alternative basis to the usual nodal and modal bases to approximate the solution. The third chapter will introduce the ADER Discontinuous Galerkin approach which uses a unique way to solve integration in time. Also insights in which conditions need to be fulfilled for the scheme to converge are given. In the fourth chapter the a posteriori Finite Volume Limiter approach is explained, showing examples of successful limiting and difficulties of the approach. The fifth chapter recaptures how the whole method got integrated into sam(oa)^2. In the end, the last chapter shows numerical benchmarks performed to show convergence of the method, impact of the limiter and a real life test case.
1 Preliminaries

To be able to understand and retrace choices, decisions and conclusions in this thesis, this chapter aims to briefly summarize the environment the ADER Discontinuous Galerkin (ADER-DG) method got developed in. Intention of this chapter is not to show the whole picture but to simplify things by not taking every single detail into account. Software environment is the sam(oa)² framework, developed by Oliver Meister, Kaveh Rahnema and Michael Bader, for which the core concepts will be introduced here.

1.1 Summary of the sam(oa)² framework

sam(oa)² works on triangular grids, which are adaptively refined by newest vertex bisection, for example explained in [19] by Mitchell. On these grids sam(oa)² imposes a memory efficient Sierpinski space-filling curve which yields a traversal order on them. A high memory efficiency is realized by a so called stack and stream data structure. For deeper insights on how these concepts work the reader is referred to [17]. In this thesis the sam(oa)² framework got used from a plain developer’s view.

For each triangle, persistent data on nodes, edges and the inner cell are stored. A layer concept separates the underlying implementation and offers a simple way to use the framework. To implement functionality several hooks are provided, functions implemented in these hooks are performed in each traversal of the whole grid. The hooks represent projections between cell, edges and nodes of a single triangle, the transfer of data and communication between triangles and element local operations. For the implementation of the ADER-DG scheme the following were essentially used:

1. The Cell to Edge operator gets called for each edge of a triangle when visited the first time within a traversal. The data stored within a cell gets projected onto the edges. Even similar calculations need to be performed for each single edge of a triangle, as there is no way to share results between them.

2. The Skeleton operator gets called for each edge, shares and combines the data previously projected from the two triangles owning this edge.

3. The Cell Update operator applies changes onto the persistent data of the cell, using the data previously gathered for each edge of the triangle.

It is guaranteed that functions are always applied on each triangle in each traversal, in this order.

A simple Finite Volume (FV) method could store the approximate average value of the solution persistently within the cell data of each triangle.
As for a constant approximation the value along the edge is equal to the inner value, the Cell to Edge operator would simply copy the cell data onto the edge data. The skeleton operator is then used to evaluate the Godunov state on each edge, taking the two averages that were projected from the holding triangles. Having the Godunov states for all edges of all triangles the Cell Update operator can then calculate the solutions for the next time step.

At this point it is necessary to mention that the perspective of sam(oa)$^2$ goes way beyond this very rudimentary example. sam(oa)$^2$ also handles general tasks, like calculating the maximal time step size by gathering wave speeds from all triangles, methods for adaptive refinement of the grid are provided, as well as an interface to make the implemented application shared memory or even MPI compatible.

1.2 Shallow water equations

The set of Partial Differential Equations (PDEs) this thesis works on are the well known Shallow Water Equations. They are preferably used to model scenarios where the vertical scale is way less than the horizontal. In this case velocities in vertical direction can be neglected. They are defined, in conservative form, as

$$Q_t + \nabla \cdot F(Q) - S(Q) = 0,$$

where

$$Q = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \quad F(Q) = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ hw \\ hvu \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}, \quad S(Q) = \begin{pmatrix} 0 \\ -ghb_x \\ -ghb_y \end{pmatrix},$$

and $\nabla$ is the divergence operator for vector fields.

Physical quantities are the water height displacement $h$, the momentum in $x$ direction $hu$ and the momentum in $y$ direction $hv$. The flux is defined by the conservation laws of mass and momentum, where $\frac{1}{2}gh^2$ takes the hydrostatic pressure into account. The source term is used to consider bathymetries, it’s concrete derivation and functionality will be shown in section 3.3.

To calculate the maximal time step size it is necessary to know the maximal wave speed. The analytic wave speed is obtained as the maximal eigenvalue of the derivation of the flux field $F(Q) = (F_1(Q), F_2(Q))$

$$\frac{\delta F_1(Q)}{\delta Q} = \begin{pmatrix} 0 & 1 & 0 \\ -u^2 + gh & 2u & 0 \\ -uv & v & u \end{pmatrix}, \text{ and } \frac{\delta F_2(Q)}{\delta Q} = \begin{pmatrix} 0 & 0 & 1 \\ -vu & u & v \\ -v^2 + gh & 2v & 0 \end{pmatrix}.$$ 

It’s eigenvalues are

$$\lambda_{1,2} = u \pm \sqrt{gh} \text{ and } \lambda_{3,4} = v \pm \sqrt{gh}.$$  

3
Figure 1: All eight triangle orientations possible in sam(oa)$^2$

For a single triangle the maximal wave speed is then obtained by evaluating the wave speed with the approximated solution. The maximal possible time step size for Finite Volume and Finite Element methods are then calculated by taking the maximal wave speed $u_{max}$ of all triangles and using the Courant-Friedrichs-Lewy condition.

\[
\Delta t_{max} = \frac{c \cdot \Delta x}{u_{max}},
\]

where $c$ is the method depending Courant number. For Finite Volume methods the Courant number is $c_{FV} = 1$, for a Discontinuous Galerkin method of order $N$ it equals $c_{DG} = \frac{1}{2N+1}$.

1.3 Geometry in sam(oa)$^2$

A common method to reduce computational effort and keep the implementation simple is the transformation of element local integrals to a reference element. To be able to perform this transformations the basic geometric properties of grids in sam(oa)$^2$ need to be discussed. For the Transformation theorem itself an affine transformation function needs to be encountered. The ADER-DG method, which will be introduced in section 3, works with integrals on triangles $T^{(p)}$ of the grid and intervals in time $[t_n, t_{n+1}]$. The reference element is thus chosen as a prism in space and time $T \times [0, 1] \subset \mathbb{R}^3$, where $T = [0, 1] \times [0, 1 - x]$ is the unit triangle.

The transformation itself can be decomposed into these two parts. As the domain is triangulated by the newest vertex bisection algorithm, triangles fulfill two conditions which allow conclusions on the transformation in space. First, triangles are rectangular with both legs of equal length, which concludes that the transformation is non-deforming. Second, the algorithm allows only 8 different orientations which are sketched in Fig. 1.
The transformation from the reference triangle $T$ to an arbitrary triangle $T(p)$ is denoted as $\tau(p)$ and consists of three parts: A translation vector determining the position of the triangle, one out of eight rotation matrices determined by an angle $\alpha_i$ and a scaling parameter $\Delta x_p$, which is equal to the size of both legs of the triangle. The transformation in time is way easier as it only projects $[0, 1]$ onto two points in time $[t_n, t_{n+1}]$.

The whole transformation in space and time can then be defined.

**Definition 2. Transformation from the reference prism to $T^{(n,p)}$**

$$
\tau_{(n,p)} : T \times [0, 1] \rightarrow T^{(n,p)}
$$

$$
\tau_{(n,p)}(x, y, t) := \begin{pmatrix}
\Delta x_p \cos (\alpha_i) & -\Delta x_p \sin (\alpha_i) & 0 \\
\Delta x_p \sin (\alpha_i) & \Delta x_p \cos (\alpha_i) & 0 \\
0 & 0 & \Delta t_n
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
t
\end{pmatrix}
+ \begin{pmatrix}
\hat{x} \\
\hat{y} \\
t_n
\end{pmatrix},
$$

where $\Delta t_n = t_{n+1} - t_n$ and $(\hat{x}, \hat{y})$ is the position of the right angle of the triangle.

Transformed integrals need to use the Jacobian $J_{(n,p)}$, which is equal to the single matrix of the transformation and its determinant $|J_{(n,p)}| = \Delta x_p^2 \cdot \Delta t_n$. 

5
2 A Quasi-Nodal Approach

The approximation of the solution and the set of test functions in Discontinuous Galerkin (DG) methods rely on the choice of a nodal or a modal basis. Both have their very own advantages and disadvantages. For nodal bases the used set of points can yield computational advantages as the right choice leads to sparse or even diagonal matrices. Also the evaluation of non linear flux and source terms can be simplified as the approximate values on the single nodes are directly known by the degrees of freedom. Non linear flux and source terms are in fact one of the disadvantages of modal bases. Here evaluation is required to be done by L2-Projections or numeric quadrature. On the other hand modal bases have advantages in showing properties of the approximated function, like the average value, or finding derivatives.

This thesis works with a hybrid of both bases, the so called Quasi-Nodal basis, as named by Beisiegel in [2]. While the approximation is done in a nodal fashion on an equidistant set of nodes, the set of used basis polynomials is the modal Bernstein basis. Foundation of this approach is the Bernstein approximation in one and two dimensions, for which convergence for functions of bounded variation is shown in [3] by Bojanic in the one dimensional case and in [5] by Derriennic for the multivariate case.

This chapter encounters the general mathematical structure of Bernstein polynomials in two and three dimensions as they are needed by the ADER-DG method of section 3 and examines properties which become useful for the concrete implementation. Start is the definition of the one and two dimensional Bernstein polynomials.

\textbf{Definition 3. One-Dimensional Bernstein polynomials}

On the interval $[0, 1]$ the one-dimensional Bernstein polynomials $iB_i^N(x)$ of degree $N$ are defined as

$$iB_i^N(x) = \frac{N!}{i \cdot (N - i)!} x^i (1 - x)^{N-i},$$

where $0 \leq i \leq N$. 

6
Definition 4. Two-Dimensional Bernstein polynomials in space
On the reference triangle $T := [0,1] \times [0,1-x]$ the two-dimensional Bernstein polynomials $2B_{i,j}^N(x,y)$ of degree $N$ are defined as

$$2B_{i,j}^N(x,y) = \frac{N!}{i! \cdot j! \cdot (N-i-j)!} x^i y^j (1-x-y)^{N-i-j},$$

where $0 \leq i, j \leq N$, $i+j \leq N$.

Fig. 2 shows an exemplary plot of all two dimensional Bernstein polynomials with degree $N = 3$. Here one can already see the similarity to the nodal basis for equidistant points. For each point a certain polynomial has
its maximum on it. In difference to the nodal basis its value on this node is not equal to one.

The solution $Q$ on a triangle $T^{(p)}$ is then approximated by assigning each of the basis polynomials the function’s value at this corresponding node.

**Definition 5. Approximated DG solution**

on triangle $T^{(p)}$ at time $t_n$:

\[
Q(x, y, t_n)_{|T^{(p)}} \approx q_{h}^{(n,p)} := \sum_{i=1}^{M} 2B_i^N(x, y)^{(p)} \cdot q_i^{(n,p)} = \phi(p) \cdot q^{(n,p)},
\]

where

\[
q_i^{(n,p)} := Q(\tau(p) \left( \begin{array}{c} i \\ j \\ k \end{array} \right) \frac{1}{N}, t_n)
\]

and

\[
2B_i^N(x, y)^{(p)} = 2B_i^N(x, y) \circ \tau^{-1}(p).
\]

As the polynomials where previously only defined on the reference element the transformation $\tau(p)$ from section 1.3 needs to be used to project basis and equidistant nodes onto $T^{(p)}$. To simplify notation the multiindex $i$ was defined for $i, j$. The last equal sign converts the summation to a vector matrix multiplication where $\phi(p)$ is the row vector holding all basis polynomials and $q^{(n,p)}$ the matrix holding the coefficients or degrees of freedom of the approximation.

The ADER-DG scheme also requires a local approximation on the space and time reference prism. Therefor the two dimensional polynomials on the reference triangle get multiplied by the one dimensional Bernstein polynomials on the $[0, 1]$ interval in time, to obtain a basis in three dimensions.

**Definition 6. Three-Dimensional Bernstein polynomials**

On the reference prism $T \times [0, 1]$ the three-dimensional Bernstein polynomials $3B_{i,j,k}^N(x, y, t)$ of degree $N$ are defined as

\[
3B_{i,j,k}^N(x, y, t) = 2B_{i,j}^N(x, y) \cdot 1B_k^N(t),
\]

where $0 \leq i, j, k \leq N$ and $i + j \leq N$.

Using the same techniques for notation an simplification, as for the two dimensional approximation, by defining a multiindex $m$ for $i, j, k$ and vectors $\phi_{(n,p)}$ and $p^{(n,p)}$ for the basis in space and time and the degrees of freedom, yields the three dimensional element local approximation.
Definition 7. Local space-time approximation

on a space time prism \( T^{(n,p)} \)

\[
Q(x, y, t) \bigg|_{T^{(n,p)}} \approx p_h^{(n,p)} := \sum_{m=1}^{M} 3B_{m}^{N}(x, y, t)^{(n,p)} \cdot p_{m}^{(n,p)} = \phi_{(n,p)} \cdot p^{(n,p)},
\]

where

\[
p_{m}^{(n,p)} := Q \left( \tau^{(n,p)} \left( \frac{j}{N}, \frac{k}{N}, \frac{l}{N} \right) \right),
\]

and

\[
3B_{m}^{N}(x, y, t)^{(n,p)} := 3B_{m}^{N}(x, y, t) \circ \tau_{(n,p)}^{-1}.
\]

As one will see in chapter 3 the Discontinuous Galerkin scheme will use the approximated solution along the surfaces of the domain. On a nodal basis using only inner nodes (e.g. for orthogonal polynomials which are based on Gauss-Legendre nodes) the approximation would need to be extrapolated. The previous definition has the advantage to be zero or decompose into the multiplication of one dimensional Bernstein polynomials along the space time surfaces and into the two dimensional Bernstein polynomials at \( t = 0 \)

\[
3B_{i,j,k}^{N}(0, y, t) = \begin{cases} 
1B_{j}^{N}(y) \cdot 1B_{k}^{N}(t), & \text{if } i = 0 \\
0, & \text{else}
\end{cases}
\]

\[
3B_{i,j,k}^{N}(x, 0, t) = \begin{cases} 
1B_{i}^{N}(x) \cdot 1B_{k}^{N}(t), & \text{if } j = 0 \\
0, & \text{else}
\end{cases}
\]

\[
3B_{i,j,k}^{N}(x, 1-x, t) = \begin{cases} 
1B_{i}^{N}(x) \cdot 1B_{k}^{N}(t), & \text{if } i = N - j \\
0, & \text{else}
\end{cases}
\]

\[
3B_{i,j,k}^{N}(x, y, 0) = \begin{cases} 
2B_{i,j}^{N}(x, y), & \text{if } k = 0 \\
0, & \text{else}
\end{cases}
\]

The first three cases allow to obtain the solution and share it between elements along the surface, by simply taking the coefficients of the non-zero two dimensional polynomials. The fourth case is useful for the predictor step, as mentioned in 3.1 the space time solution at \( t = 0 \) is simply the spatial solution.

By the remarks of chapter 3.1.1 it is crucial to be able to easily detect drying areas. One and two dimensional Bernstein polynomials share two
handy features, being positive and summing up to one on any point

\[ \forall x \in [0, 1] \]
\[ \sum_{i=0}^{N} B_i^N(x) = 1, \forall x \in [0, 1] \]
\[ \sum_{i=0}^{N} \sum_{j=0}^{N-i} a_{i,j} \cdot B_{i,j}^N(x, y) \leq \sum_{i=0}^{N} \sum_{j=0}^{N-i} \min a_{i,j} \cdot B_{i,j}^N(x, y) = \min a_{i,j}. \] (1)

For a polynomial, having the two dimensional Bernstein basis, these properties allow to conclude that as long as all coefficients of the polynomial are positive the polynomial is nowhere less then zero. In fact this conclusion can be generalized to

\[ \sum_{i=0}^{N} \sum_{j=0}^{N-i} a_{i,j} \cdot B_{i,j}^N(x, y) \geq \sum_{i=0}^{N} \sum_{j=0}^{N-i} \min a_{i,j} \cdot B_{i,j}^N(x, y) = \min a_{i,j}. \] (1)

The approximated water height displacement is thus at least as high, as the smallest coefficient. The same conclusion can be made for an upper boundary, which will not be used in this thesis.
3 The ADER Discontinuous Galerkin Scheme

Foundation of this thesis is the ADER-DG scheme as first proposed by Dumbser et al. in [7]. The scheme considers a general hyperbolic Partial Differential Equation as defined by

\[ \frac{\partial Q}{\partial t} + \nabla \cdot F(Q) - S(Q) = 0 . \]  (2)

\( \nabla \cdot F(Q) \) is the called the conservative term of the equation. For two space dimensions \( x \) and \( y \) it equals \((F(Q)_1)_x + (F(Q)_2)_y\) and can be written in quasi-linear form \( \delta F(Q) / \delta Q \cdot Q \). For an arbitrary normalized vector \( \vec{n} \), \( \delta F(Q) / \delta Q \cdot \vec{n} \) has only real eigenvalues. \( S(Q) \) is called source term and used to handle non conserving parts of the equation.

The main point distinguishing the ADER-DG approach from other Discontinuous Galerkin methods is the way integration in time is solved. Classical approaches build the weak form on a set of test polynomials in space and evolve this equation system into a so called semi-analytic scheme by discretizing it in space only. The resulting formulation is equivalent to an Ordinary Differential Equation in time which can then be solved by the usual numerical solvers, as for example Runge-Kutta methods. A more detailed insight into this approach is given in [14] by Hesthaven andWarburton. These methods hold two disadvantages, for orders higher than four the formulation of the algorithm becomes cumbersome and, as for each intermediate solution in time the flux terms have to be evaluated, multiple traversals of the whole domain are required.

The ADER-DG method omits the semi-analytic scheme and solves the integration in time by the result of a nonlinear equation system. The equation system itself is solved by a unique fix-point iteration, yielding the so called DG-Predictor. It can be generated locally for every triangle within a single traversal.

3.1 The Predictor Step

The predictor step is the successor of a scheme using the Cauchy-Kowalewskaja method to solve integration in time. The old method is very detailed introduced in [24] by Toro. The nomination ADER was composed from the usage of Arbitrary high order DERivatives by this predecessor. A detailed explanation of the predictor scheme can be found in the original derivation by Dumbser et al. in [6]. In this chapter it is revisited and combined with the geometric conclusions about triangulation and transformation in sam(oa)\(^2\), as discussed in chapter 1.3.

To get the predictor step between time steps \( t_n \) and \( t_{n+1} \) on a triangle \( T^{(p)} \), ansatz is the weak formulation of the Partial Differential Equation (2). Over the prism \( T^{(n,p)} = T^{(p)} \times [t_n, t_{n+1}] \) in space and time it is
\[
\int_{T^{(n,p)}} \phi^{(n,p)}_T \cdot \left( \left( p^{(n,p)}_h \right)_t + \nabla \cdot F \left( p^{(n,p)}_h \right) - S \left( p^{(n,p)}_h \right) \right) \, dxyt = 0 ,
\]

where \( Q \) is approximated by \( p^{(n,p)}_h \) on a set of element local space-time basis polynomials, which are ordered in a column vector \( \phi^{(n,p)} \). Note that from this point on all integrals and functions evaluated on vectors are considered element wise. As usual for Discontinuous Galerkin approaches the same set of polynomials is used for the approximation as well as the test functions. The original formulation uses a nodal basis, in this thesis the Quasi-nodal approach, as described in section 2 by definition 7, was used.

Partial integration in time of the first term, transformation to the reference space-time prism \([0, 1] \times T\) and approximation of the flux and source terms yields the nonlinear equation system

\[
\begin{aligned}
&\left( \int_T \phi^T \cdot \phi \, dxy \bigg|_{t=1} - \int_T \phi^T \cdot \phi \, dxy \bigg|_{t=0} - \int_{[0,1]} \int_T (\phi)_x^T \cdot \phi \, dxy \, dt \right) \cdot p^{(n,p)} + \\
&\frac{\Delta t_n}{\Delta x_p} \left( \int_{[0,1]} \int_T \phi^T \cdot (\phi) \, dxy \, dt \cdot \hat{F}_1 \left( p^{(n,p)} \right) \right) + \\
&\int_{[0,1]} \int_T \phi^T \cdot (\phi)_y \, dxy \, dt \cdot \hat{F}_2 \left( p^{(n,p)} \right) - \\
&\int_{[0,1]} \int_T \phi^T \cdot \phi \, dxy \, dt \cdot S \left( p^{(n,p)} \right) \right) = 0 ,
\end{aligned}
\]

where \( \left( \hat{F}_1 \left( p^{(n,p)} \right), \hat{F}_2 \left( p^{(n,p)} \right) \right)^T = \left( F^{-1}_{(n,p)} \cdot \left( F_1 \left( p^{(n,p)} \right), F_2 \left( p^{(n,p)} \right) \right) \right)^T \), by the chain-rule. \( p^{(n,p)} \) is the vector of degrees of freedom of the approximation in space and time and the unknown value to compute. It is called **DG-Predictor** and will replace integration in time.

By using the approximation on flux and source terms and transforming integrals to the reference element, the resulting integral-matrices can be precalculated and inversions realized as LU-Decomposition. The solution to this system turns into matrix-matrix multiplications and application of LU-Factorized matrices. More expensive L2-Projections and a repeating numerical quadrature are avoided this way, for the price of lost accuracy. To maintain a well-balanced scheme in the final realization a more developed approach, than the here presented, needed to be used on the source term. A detailed description will be given in section 3.3.
By defining notations for the space time mass matrix $M$, the stiffness matrices $K_x, K_y, K_t$ and the matrix $W$, which is the mass matrix evaluated along the surface in time, the expression gets abbreviated

$$p^{(n,p)} = (W - K_t)^{-1} \frac{\Delta t_n}{\Delta x_p} \left( -K_x \cdot \hat{F}_1 \left( p^{(n,p)} \right) - K_y \cdot \hat{F}_2 \left( p^{(n,p)} \right) + M \cdot S \left( p^{(n,p)} \right) \right) = 0,$$

where the unknown $p^{(n,p)}$ got separated from all matrix multiplications and non-linear terms. For a deeper insight into the next step the reader is again referred to [6, 7]. As the coefficients of the approximation are build on the evaluation on a set of equidistant nodes in space and time, the unknown can be split into a part concerning $t = 0$, $p^{(n,p)}_0$ and the remaining degrees of freedom $p^{(n,p)}_1$. At $t = 0$ the solution to the equation system is already known, as it is equivalent to the full ADER-DG scheme at time step $t_n$, which will be introduced in the next subsection 3.2. For a matrix $M = \begin{pmatrix} M_{0,0} & M_{0,1} \\ M_{1,0} & M_{1,1} \end{pmatrix}$, where $M_{0,1}$ and $M_{0,0}$ are the part matrices regarding the solution at $t = 0$, only matrices $M_{1,1}$ and $M_{1,0}$ have to be considered. Splitting all matrices and discarding the already solved parts shortens the number of equations in the nonlinear system

$$p^{(n,p)}_1 = (W - K_t)_1^{-1} \frac{\Delta t_n}{\Delta x_p} \left( -K_{x,1} \cdot \hat{F}_1 \left( p^{(n,p)}_1 \right) - K_{y,1} \cdot \hat{F}_2 \left( p^{(n,p)}_1 \right) + M_{1,1} \cdot S \left( p^{(n,p)}_1 \right) \right) - \left( (W - K_t)_{0,0} \cdot p^{(n,p)}_0 + K_{x,0} \cdot \hat{F}_1 \left( p^{(n,p)}_0 \right) + K_{y,0} \cdot \hat{F}_2 \left( p^{(n,p)}_0 \right) - M_{1,0} \cdot S \left( p^{(n,p)}_0 \right) \right).$$

It turns out that the right side of this equation system is a self contraction with respect to $p^{(n,p)}_1$. This allows to solve the system by a fix point iteration, having the solution $p^{(n,p)}_1$ as limit value. Using $i$ as iteration-index.
\[ i+1 p_1^{(n,p)} = (W - K_t^{(n,p)})^{-1} \cdot \frac{\Delta t_n}{\Delta x_p} \]

\[
\left( -K_{1,1}^{x} \cdot \hat{F}_1 \left( p_1^{(n,p)} \right) - K_{1,1}^{y} \cdot \hat{F}_2 \left( p_1^{(n,p)} \right) \right) \\
+ M_{1,1} \cdot S \left( p_1^{(n,p)} \right) \\
\left( (W - K_t^{(n,p)})_{0,0} \cdot p_0^{(n,p)} + K_{1,0}^{x} \cdot \hat{F}_1 \left( p_0^{(n,p)} \right) + K_{1,0}^{y} \cdot \hat{F}_2 \left( p_0^{(n,p)} \right) \right) \\
- M_{1,0} \cdot S \left( p_0^{(n,p)} \right) \right).
\]

\[ p_0^{(n,p)} \] is the spacial solution at \( t_n \), the properties of the space time basis polynomials, shown in section 2, allows to simply take the degrees of freedom of the spatial solution \( q_{0}^{h} \) at time \( t_n \).

The original paper [6] suggests to set the initial value \( 0 p_1^{(n,p)} \) to a steady state solution, e.g. for shallow water equations \( 0 p_1^{(n,p)} = 0 \). After a number of iterations equal to the number of degrees of freedom it is shown that the limit value is reached with a precision of at least \( 10^{-9} \).

### 3.1.1 Divergence of the Predictor Step

At this point I want to emphasize one crucial remark. For a water-height near to zero the predictor scheme often diverged. I assume this behavior originates in the flux function being unbound on a dried surface. A short idea why this happens can be given by looking at the self contractivity the scheme has to fulfill to ensure its convergence

\[
\frac{\Delta t_n}{\Delta x_p} \left( (W - K_t^{(n,p)})^{-1} \cdot \left( K_{1,1}^{x} \cdot \left( \hat{F}_1 (x) - \hat{F}_1 (y) \right) \right) + K_{1,1}^{y} \cdot \left( \hat{F}_2 (x) - \hat{F}_2 (y) \right) - M_{1,1} \cdot (S (x) - S (y)) \right) \right)
\]

\[ := G(x, y) \leq \|x - y\|, \]

where \( x, y \) are two reachable states within a single time step. Using the triangle inequality and operator norms the left side can be bounded

\[
G(x, y) \leq \frac{\Delta t_n}{\Delta x_p} \left( \left\| (W - K_t^{(n,p)})^{-1} \cdot K_{1,1}^{x} \cdot \hat{F}_1 \right\| + \left\| (W - K_t^{(n,p)})^{-1} \cdot K_{1,1}^{y} \cdot \hat{F}_2 \right\| + \left\| (W - K_t^{(n,p)})^{-1} \cdot M_{1,1} \cdot S \right\| \cdot \|x - y\| \right) := L \|x - y\|. \]

That \( L \leq 1 \) is shown in the original paper. This boundary however fails as long as it is not ensured that the flux functions are finite, which can be seen by the definition of their operator norm

\[
\|F_i\| = \sup_{\|x\| \neq 0} \frac{\|F_i (x)\|}{\|x\|} = \infty, \]

14
where the last equation holds for the flux terms as \( \lim_{h \to 0} \frac{1}{2} (hu)^2 / h = \infty \) for \( hu \neq 0 \).

As consequence one must take care that a water-height of zero can not be reached within a run of the iteration. In the final implementation the problem got solved by switching to a Finite Volume scheme as defined in section 4.2.1 as soon as the initial water-height drops below a predefined threshold.

### 3.2 The Full Scheme

To now obtain the full ADER-DG Scheme the weak form is again build in space and time. Now only the element local spacial basis, as row vector \( \phi_{(p)} \), is used as set of test functions

\[
\int_{T^{(n,p)}} \phi_{(p)}^T \cdot (Q_t + \nabla \cdot F (Q) - S (Q)) \, dxyt = 0.
\]

By using partial integration in time on the first summand and then approximating the solution in space the proceeding in time of the numerical solution is obtained

\[
\int_{T^{(n,p)}} \phi_{(p)}^T \cdot Q_t \, dxyt = \int_{T^{(p)}} \phi_{(p)}^T Q \, dx \bigg|_{t=t_n}^{t=t_{n+1}} \approx \int_{T^{(p)}} \phi_{(p)}^T \left( q^{(n+1,p)} - q^{(n,p)} \right) \, dx
\]

\[
\int_{T^{(p)}} \phi_{(p)}^T \phi_{(p)} \left( q^{(n+1,p)} - q^{(n,p)} \right) \, dx.
\]

Note, that the volume integral in time, which would normally occur when applying partial integration, equals zero, as the set of test functions is constant in time. The flux integral gets treated similar, by applying partial integration in \( x \) and \( y \) direction and using the previously calculated predictor step to approximate the solution in space and time. As the test functions are not constant in space, the volume integrals remain
\[ \int_{T(n,p)} \phi^T_{(p)} \cdot \nabla \cdot F(Q) \, dxyt \approx \]
\[ \sum_{i=1}^{3} \int_{\delta T_i^{(n,p)}} \phi^T_{(p)} \cdot D - \left( p_h^{(n,p)}, p_h, \gamma \right) \cdot \mathbf{n} \, dxy - \]
\[ \int_{T(n,p)} \nabla \phi^T_{(p)} \cdot F \left( p_h^{(n,p)} \right) \, dxyt \approx \]
\[ \sum_{i=1}^{3} \int_{\delta T_i^{(n,p)}} \phi^T_{(p)} \cdot \phi_{(n,p)} \, dxyt \cdot F \left( p_h^{(n,p)} \right) . \]

Along the space time boundaries \( \delta T_i^{(n,p)} \) a Riemann solver \( D^- (\cdot, \cdot) \cdot \mathbf{n} \) needs to be applied to obtain the Godunov state. A short insight on the options for Riemann solvers for Shallow Water Equations is given in the next chapter.

By again defining matrices \( M \) as spatial mass, \( K_x \) and \( K_y \) as stiffness, \( B_i \), for \( i = 1, \ldots, 3 \) as boundary and \( \hat{M} \) as mass matrix on the reference element the full scheme can be written as

\[ q^{(n+1,p)} = q^{(n,p)} - M^{-1} \frac{\Delta t}{\Delta x_p} \left( \sum_{i=1}^{3} B_i \cdot D \left( p_h^{(n,p)}, p_i^{(n,p)} \right) \right) \]
\[ - K_x \cdot \hat{\mathbf{F}}_1 \left( p_h^{(n,p)} \right) - K_y \cdot \hat{\mathbf{F}}_2 \left( p_h^{(n,p)} \right) - \hat{M} \cdot S \left( p_h^{(n,p)} \right) , \]

where \( \hat{\mathbf{F}}_1 \) and \( \hat{\mathbf{F}}_2 \) are again the transformed fluxes. The source accordingly got approximated by the Bernstein polynomials. This formulation also shows that the only value which needs to be shared between cells is the DG-Predictor.
3.3 Well Balanced Property

One indicator on how numerical stable Finite Element methods are is the preservation of equilibria. Benchmark scenario for this so called well-balanced property is a resting lake. The initial conditions are

\[
\begin{align*}
  h_0(x, y) + b(x, y) &= H = const. \\
  u_0(x, y) &= 0 \\
  v_0(x, y) &= 0,
\end{align*}
\]

where \(b(x, y)\) is an arbitrary bathymetry. The solution should stay constant in time and preserve water height and velocities,

\[
\begin{align*}
  h(x, y, t) + b(x, y) &= H = const. \\
  hu(x, y, t) &= 0 \\
  hv(x, y, t) &= 0.
\end{align*}
\]

In the analytic case the source term, from section 1.2 is defined by this condition

\[
Q_t + \nabla \cdot F(Q) - S(Q) = 0 \iff \begin{pmatrix} 0 \\ \frac{1}{2}gh^2 \\ 0 \end{pmatrix}_x + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2}gh^2 \end{pmatrix}_y - S(Q) = 0.
\]

With \(h = H - b\) and thus \(h_x = -b_x\) for the derivative in \(x\)- and in \(y\)-direction analogously yields the source term

\[
S(Q) = \begin{pmatrix} 0 \\ -ghb_x \\ -ghb_y \end{pmatrix}.
\]

This term fulfills the function of balancing out the flux terms for a steady state solution. Within the numeric scheme the source integral has to fulfill the same task.

To achieve a well-balanced scheme, or in other terms a scheme with exact C-property, it is crucial to ensure that the solution of the Riemann problem along the domain boundaries preserves the equilibrium. A popular approach is the modification of the flux term by slope limiting. The flux term gets adjusted to exactly fit the source term for a constant solution. A good explanation with application can be found in [10]. Roe-solvers, as for example presented in [11], state a set of well-balanced Riemann solvers which are also able to handle wetting and drying. In this thesis the scheme relies on a set of Riemann-solvers as proposed by George in [13] which extend Roe-solvers in many properties. Mainly used are two quite similar solvers
called HLLE and augmented Riemann solver, which both maintain the non-negativity of the water height displacement. Along constant bathymetries both solvers are equivalent while for a variable bathymetry the augmented Riemann solver uses a more accurate approach. For schemes not using polynomial approximation on the source and flux terms, and evaluating volume integrals by numeric quadrature, these solvers are sufficient to ensure the exact C-property.

As the Bernstein approximation is used on the flux term to omit a repeating numeric quadrature, the source term needs to be treated similar. To be well-balanced both volume integrals need to be equal for steady state solutions

\[
\int \int_{[0,1] \times [0,1]} \phi^T \cdot \nabla \phi \, dx \, dy \, dt \cdot F \left( p^{(n,p)} \right) = \int \int_{[0,1]} \phi^T \cdot S \left( p_h^{(n,p)} \right) \, dx \, dy \, dt.
\]

Using the native approach of also approximating the source term by Bernstein polynomials leads to an unstable scheme, which is easily seen by the fact that for the Bernstein basis in general

\[
\phi_x(f(x_i)) = \phi(f_x(x_i))
\]
does not hold. Thus, the source integral must not be approximated before evolved into a form, fitting the flux integral. As approach, the total approximated water height is defined as \( H_h = h_h + b_h \), where \( h_h = (p_h^{(n,p)})_1 \) and \( b_h \) is a suitable approximation of the bathymetry. By splitting the division, the part of the integral only concerning non constant total water heights can be separated

\[
\int \int \phi^T \cdot S \left( p_h^{(n,p)} \right) \, dx \, dy \, dt =
\int \int_{[0,1]} \phi^T \left( 0, -gh_h(H_h - h_h)x, -gh_h(H_h - h_h)y \right) \, dx \, dy \, dt =
\int \int_{[0,1]} \phi^T \left( 0, -gh_h(H_h)x, -gh_h(H_h)y \right) \, dx \, dy +
\int \int_{[0,1]} \phi^T \left( 0, gh_h(h_h)x, gh_h(h_h)y \right) \, dx \, dy.
\]

For a constant total water height the first integral vanishes, leaving the second integral which can now be transformed, by using Green’s formula into a form equivalent to the flux integral.
\[ \int_{\text{T} \times [0,1]} \phi^T \left( 0, gh_h(h_h)_x, gh_h(h_h)_y \right) dxyt = \]
\[ \int_{\delta \text{T} \times [0,1]} \phi^T \cdot \left( 0, \frac{1}{2} gh_h^2 \cdot n_2, \frac{1}{2} gh_h^2 \cdot n_3 \right) dxyt - \]
\[ \int_{\text{T} / \delta \text{T} \times [0,1]} \phi_x^T \cdot \left( 0, \frac{1}{2} gh_h^2, 0 \right) + \phi_y^T \cdot \left( 0, 0, \frac{1}{2} gh_h^2 \right) dxyt. \]

If the integrands get approximated at this point the resulting term exactly cancels out the corresponding flux terms for a resting lake scenario. The solution stays constant.
4 A posteriori Finite Volume Limiter Approach

Through the polynomial approximation of solutions, Discontinuous Galerkin methods have been proved to be efficient to model smooth scenarios, as for example by Johnson et al. in [15]. Though, due to the lack of appropriately representing non-smooth functions, these lead to oscillations and instabilities, which is also known as Gibbs phenomenon. There have been various proposals on how to limit these instabilities. One of the main ideas considers Total Variation Diminishing (TVD), where the computed solution at each time step gets manipulated (by the well known minmod function) in a way that the total variation stays bounded. This famous approach is for example presented in [4] by Cockburn and Shu. Other methods use ENO/WENO-Reconstruction and retrieve damped degrees of freedom, as Qui et al. in [20].

All common methods share one characteristic. The DG solution is first calculated, then tested and finally manipulated to fulfill a certain indicator on stability. The new limiter approach varies the last step by repeating the calculation \textit{a posteriori} with Finite Volume methods, which are more efficient in solving non smooth systems. Additionally a new set of criteria to detect troubled cells is proposed.

The new a posteriori Finite Volume Limiter approach was introduced by Dumbser and Loubère in [9]. It can be summarized in four steps which are performed at every time step \( t_n \). The single steps will be discussed in detail in a very own subsection.

1. **Computation of a Candidate Solution** Using the ADER-DG scheme as described in section 3 for each cell a candidate solution \( \mathbf{q}_h^{(n+1,p)} \) at time \( t_{n+1} \) is computed.

2. **Detecting troubled Cells** The resulting values get checked by two admissibility conditions, the Physical Admissibility Detection criteria (PAD) and Numerical Admissibility Detection criteria (NAD). Cells that are detected get marked as troubled.

3. **Corrector Step** For troubled cells the candidate solution is discarded. The solutions at time \( t_n \) for Triangle \( T(p) \) and all its adjacent neighbors \( T^{(p_i)} \) get converted to Finite Volume representations \( v^{(n,p)} \) and \( v^{(n,p_i)} \), on a patch of cells. The solution for \( v^{(n+1,p)} \) gets computed.

4. **Reconstructing the DG-Solution** For troubled cells the final approximation \( \mathbf{q}_h^{(n+1,p)} \), at time \( t_{n+1} \), gets reconstructed from the new computed Finite Volume solution \( v^{(n+1,p)} \). For untroubled cells the candidate solution is taken.

Fig. 8 on page 29 shows the whole process as it is implemented in sam(oa)^2.
Figure 3: Reference element with patch subcells for N=4

4.1 Converting Representations

One of the main tools of this limiter is the conversion between both representations. This chapter treats the task of switching between a polynomial basis within a cell, as used in DG schemes and a Finite Volume approximation on a set of subcells or in other words on a patch of cells. For the original approach the reader is referred to [8], where the conversion is introduced on a set of orthogonal basis polynomials. This chapter will encounter the same approach for a general polynomial basis.

4.1.1 Finite Volume Approximation on Subcells

To convert the polynomial representation \( q_h^{(n,p)} \) on a Triangle \( T^{(p)} \) to a Finite Volume patch, \((N_s + 1)^2\) sub-triangles \( t_i \) are generated by taking \( N_s \) equidistant intersections along each edge and connecting them diagonal, vertical and horizontal. The original paper suggests to take \( N_s = 2N + 1 \) where \( N \) is the order of the DG scheme. Intention is to keep the CFL condition \( c \) for DG solution and FV patch constant. By considering definition (1) from section 1.2

\[
c_{DG} = u \cdot (2N + 1) \cdot \frac{\Delta t_n}{\Delta x_p} = u \cdot \frac{\Delta t_n}{\Delta x_p/(2N + 1)} = c_{FV},
\]

where \( u \) is the wave speed. Fig. 3 shows an example for the reference element \( T := [0, 1] \times [0, 1-x] \). With help of the Transformation theorem the conversions for all triangles will be reduced to a reference element having the same number of intersections. Within the reference element all vertices are given by \( \xi_{k,j} = \left( \frac{k}{N_s}, \frac{N_s-j}{N_s} \right) \), \( 0 \leq k \leq j \leq N_s \). Here one can see that each subtriangle \( t_i \) on the element has the same area \( |t_i| = N_s^{-2}/2 \). For an arbitrary triangle \( T^{(p)} \) the Transformation theorem yields for all subtriangles \( t_i^{(p)} \) an area of \( |t_i^{(p)}| = \Delta x_p^2 |t_i| \). To convert the DG solution \( q_h^{(n,p)} \) to a set of
Figure 4: Conversion of a Bernstein polynomial of degree four on the left to a Finite Volume approximation on 81 subtriangles on the right.

finite volume solutions \( v_{i}^{(n,p)} \) on these subtriangles the standard averaging approach is used

\[
\hat{v}_{i}^{(n,p)} := \frac{1}{|t_{i}^{(p)}|} \int_{t_{i}^{(p)}} q_{h}^{(n,p)} \, dxy.
\]

Again using the Transformation theorem yields

\[
\frac{1}{|t_{i}^{(p)}|} \int_{t_{i}^{(p)}} q_{h}^{(n,p)} \, dxy = \frac{1}{|t_{i}^{(p)}|} \int_{t_{i}^{(p)}} \phi^{(p)} \, dxy \cdot q^{(n,p)} = \frac{1}{|t_{i}|} \int \phi \, dxy \cdot q^{(n,p)},
\]

where integrals over vectors and matrices are considered element wise. Defining matrices \( \Phi \) and \( v^{(n,p)} \) over all subtriangles states the conversion

\[
v^{(n,p)} = \frac{1}{|t_{i}|} \Phi \cdot q^{(n,p)}.
\]

Note that \( |t_{i}|^{-1} \Phi \) can be calculated preliminary as \( |t_{i}| = |t_{j}|, \forall i, j \). The conversion becomes a simple matrix vector multiplication on each subtriangle. Fig. 4 shows the result of a conversion of an arbitrary polynomial of degree four. The polynomial has 15 degrees of freedom the resulting Finite Volume solution is defined on 81 subtriangles. The general structure of the polynomial remains in the Finite Volume solution. In fact it is distinct, the polynomial can be exactly reconstructed from the FV representation.
4.1.2 Reconstructing the polynomial Representation

Starting with a patch of Finite Volume solutions $v^{(n,p)}$ on a Triangle $T^{(p)}$ the reconstructed polynomial $q_h^{(n,p)}$ needs to fulfill the reverse requirement of the previous section, which means, using the same notation

$$\frac{1}{|t_i|} \Phi \cdot q^{(n,p)}_h \upharpoonright = v^{(n,p)}.$$  

As $N_s^2 = (2N + 1)^2$ subtriangles where defined, but the polynomial only has $M = (N + 1) \cdot (N + 2)/2$ degrees of freedom, the equation system is over determined and can only be solved approximately. To get a distinct solution a constraint least squares approximation for each $j$-th physical quantity $q^{(n,p)}_s$ and $v^{(n,p)}_s$ is performed. The original paper suggests constraining solutions to the ones keeping the integral on the whole triangle constant,

$$\int_{T^{(p)}} \left( q_h^{(n,p)}_j \right) \, dx \, dy = \frac{1}{|t_i|} \sum_{i=1}^{N_s^2} q_i^{(n,p)}_j,$$

where the fact that all subtriangles have the same size got used.

Again using the Transformation theorem both sides decompose to

$$\int_{T^{(p)}} \left( q_h^{(n,p)}_j \right) \, dx \, dy = \Delta x_p^2 \int_T \phi \, dx \, dy \cdot q^{(n,p)}_s = \Delta x_p^2 \Phi \cdot q^{(n,p)}_s$$ and

$$|t_i| \sum_{i=1}^{N_s^2} v_i^{(n,p)}_j = \Delta x_p^2 |t_i| \|v^{(n,p)}_s\|_1,$$

where the elements of $\Phi \in \mathbb{R}^{1 \times M}$ are defined as the integral of each basis function on the reference element and $\| \cdot \|_1$ is the sum of all vector elements without taking the absolute value, in contrast to the common definition. The constraint least squares solution is then obtained by finding the minimum of the function

$$f = \left( \frac{1}{|t_i|} \Phi \cdot q^{(n,p)}_s - v^{(n,p)}_s \right) \cdot \left( \frac{1}{|t_i|} \Phi \cdot q^{(n,p)}_s - v^{(n,p)}_s \right) - \lambda \left( \frac{1}{|t_i|} \Phi \cdot q^{(n,p)}_s - |v^{(n,p)}_s|_1 \right).$$

As the function is convex the minimum is the single root of the derivative.

$$\frac{\partial f}{q^{(n,p)}_s} = \frac{2}{|t_i|} \Phi^T \cdot \Phi \cdot q^{(n,p)}_s - (v^{(n,p)}_s)^T \cdot \Phi - \Phi^T \cdot v^{(n,p)}_s - \lambda \Phi^T \upharpoonright = 0,$$

$$\frac{\partial f}{\partial \lambda} = \frac{1}{|t_i|} \Phi \cdot q^{(n,p)}_s - |v^{(n,p)}_s|_1 \upharpoonright = 0.$$
This is a linear equation system

\[
\begin{pmatrix}
2\Phi^T \Phi & \Phi^T \\
\Phi & 0
\end{pmatrix}
\begin{pmatrix}
q_{s,j}^{(n,p)} \\
-\lambda |t_i|
\end{pmatrix}
= |t_i|
\begin{pmatrix}
2 \cdot \Phi^T v_{s,j}^{(n,p)} \\
|v_{s,j}^{(n,p)}|_1
\end{pmatrix},
\]

where the matrix on the left side again can be computed and LU factorized in advance. As \(\lambda\) has no further importance its calculation can be dropped. For Finite Volume representations, which originate from a DG polynomial, the reconstruction is simply the inversion. Fig. 5 shows an exemplary conversion of a discontinuity from a Finite Volume to a polynomial representation. For this type of functions the reconstruction is not exact, recovering the Finite Volume representation would have a completely different result. The discontinuity gets smoothed and oscillations occur around the edges.
4.2 Detection criteria

The decision which cells are limited and which are considered as stable is made based on two types of criteria. The Physical Admissibility Detection criteria (PAD) and Numerical Admissibility Detection criteria (NAD).

4.2.1 Physical Admissibility Detection Criteria

PADs take physical restrictions into account. For particle simulations a velocity above light speed would violate laws of relativity theory, simulations considering car fluxes can not allow negative car density along a road. Also NaNs are taken into account which could happen for too high values or a non physical division through zero.

For the Shallow Water Equations the main point of interest is the water height displacement as it is forbidden to drop below zero. Furthermore, as shown in 3.1.1, it needs to be ensured that the water height displacement does not drop below a certain threshold, else the predictor step diverges. In the final implementation the simple lower-bound from equation (1) section 2 got used to detect cells below this threshold. If a cell violates this restriction it gets detected and marked as troubled.

4.2.2 Numerical Admissibility Detection Criteria

NADs check the simulation for instabilities originating in numerical errors. The main indicator used is the so called relaxed Discrete Maximum Principle which states boundaries for the solution. For triangle $T(p)$ the solution at time $t_{n+1}$ must lie between the maximal and minimal values of its own and its whole Voroni neighborhood’s solution at $t_n$. All considered triangles are summarized by $V(p)$.

**Definition 8. Relaxed Discrete Maximum Principle**

$$\min_{p_i \in V(p)} \left( q_{h}^{(n,p_i)} \right) - \delta \leq q_{h}^{(n+1,p)} \leq \max_{p_i \in V(p)} \left( q_{h}^{(n,p_i)} \right) + \delta,$$

where

$$\delta := \max \left( \epsilon_0, \epsilon \left( \max_{p_i \in V(p)} \left( q_{h}^{(n,p_i)} \right) - \min_{p_i \in V(p)} \left( q_{h}^{(n,p_i)} \right) \right) \right),$$

and $\epsilon_0$ and $\epsilon$ are defined constants (normally $\epsilon_0 = 10^{-4}$ and $\epsilon = 10^{-3}$).

The principle is called relaxed because the solution does not have to lie strictly between these boundaries, as it is necessary for its analytic counterpart for parabolic and elliptic PDEs, but within a certain tolerance defined by $\delta$. Note that the comparison is done for each physical quantity.
The idea behind the relaxed Discrete Maximum Principle can be explained by the one-dimensional Shallow Water Equations. The linear dam break, as introduced and solved in [16] by Leveque, consists of a simple discontinuity in the water height at $t = 0$ (Fig. 6 left). Jumps in functions are a known source of oscillations for polynomial representations. Because of this, instabilities are expected to occur along the discontinuity for the next time step. These instabilities get detected by the relaxed maximum principle (Fig. 6 right), for suitable definitions of the constants. The limiter checks if the cell’s extreme values at the next time step lie between the extreme values of the cell and its neighbors of the previous time step, enlarged by a buffer.

The buffer is intended to allow new occurring extreme values which are possible for nonlinear hyperbolic PDEs. However this criterion is not fail save. For a completely smooth scenario, as a Gaussian curve, shown in Fig. 7, cells might violate the relaxed maximum principle. With the initial velocity being zero the admitted interval for the momentum reduces to $[-\delta, \delta]$. Due to acceleration new maximal and minimal values occur on the next time step. As soon as the acceleration is too high, in this example it only depends on the water height, the new momentum can overshoot the interval. The cell gets marked as troubled despite having no instability. This example states the question if it is possible to prevent this unintended limiting by only considering water height and neglect the momentum. To test this idea a numerical comparison was performed in 6.3.

A second solution might be found in elementary analysis by making the
buffer depending on the $\epsilon - \delta$ definition of continuous functions

$$\forall \epsilon > 0 \exists \delta > 0 : \| t_n - t_{n+1} \| < \delta \implies \| f(\vec{x}, t_n) - f(\vec{x}, t_{n+1}) \| < \epsilon.$$ 

Finding a constant $\delta$, depending on $\epsilon$ and the PDE could yield a suitable buffer size to not erroneously detect smooth solutions.
5 Realization in sam(oa)$^2$

Having encountered all necessary topics to implement the method, it is now possible to show how the scheme got realized in sam(oa)$^2$. Fig. 8 shows how the numerical solution is calculated for each single time step.

For each time step three traversals of the whole domain are performed. The first is used to calculate the DG-Predictor (Predictor Traversal), the second the Discontinuous Galerkin solution (DG Solver Traversal) and detect troubled cells, the third to apply the Finite Volume solution to previously detected cells (Patch Solver Traversal).

Solving for the DG-Predictor (Predictor Step) and calculating the solution (DG Solver) was realized in two separate traversals. The full scheme from section 3.2 needs the DG-Predictor values along all adjacent edges to compute the solution of the next time step. By recapturing section 1.1 calculations done at the projection onto the edge can’t be stored. If the Predictor step were to be realized before the DG solver in a single traversal, the Predictor would need to be calculated for each single edge of a triangle, three times. The size of the next time step can only be determined after all wave speeds are gathered. It is thus not possible to simply generate the DG-Predictor right after calculating the next solution as there is no way the new time step can be known at that point. In fact it only can be generated after all troubled cells got corrected, as only then all wave speeds are known.

Before performing the predictor step it is necessary to check (Cell Drying ?) that the water height displacement is higher than the threshold from section 3.1.1. If it is below that threshold it must get marked as troubled, to avoid the predictor to diverge. As soon as a cell has neighbors that where marked as troubled in the first traversal, the ADER-DG scheme can not be applied, as no DG-Predictor for the neighboring cell exists (Any neighbor drying ?). The cell itself needs to be evaluated by the FV scheme and gets marked as troubled. On the other hand cells that have neighbors that got detected by the previous test can proceed normally, DG-Predictors exists as the cells where not troubled at the previous traversal. After evaluating the DG solution it gets checked by the NADs and PADs (Limiter), the original proposal suggests to convert the DG solution to a FV solution to simplify finding maximal and minimal values. Solutions that passed are not manipulated any more. For detected cells the old state needs to be recovered (Recover old state) and can for example be extracted from the DG-Predictor.

In the last traversal all detected cells get converted to a Finite Volume solution. As the DG-Solution of all neighbors is known, they can also be converted to a FV solution. The solution for the next time step are computed (Patch solver) and then converted back to a DG polynomial.
Figure 8: Flow chart for the whole scheme implemented with limiter. Dotted boxes represent single traversals of the whole domain.
6 Numerical Analysis

In the following chapter the ADER-DG scheme, introduced in the previous sections and implemented in sam(oa)$^2$, is validated and analyzed numerically. Core aspects are the numerical convergence, how the limiter behaves and what impact it has on the accuracy of the solution. To show how well the scheme behaves the error-revealing oscillating lake was simulated. In the end the Tohoku Tsunami as occurred in 2011 was modeled as a real life application.

All analysis will be done on uniform static grids, where the grid size $h$ is given by the length of a single triangle leg. The number of cells and $h$ correlate, on a domain $L \times L$ it holds $\#\text{cells} = 2 \cdot L \cdot (h^{-1})^2$.

Solutions in cells that got detected by the limiter are always replaced by a Finite Volume Euler scheme. Furthermore the limiter is used for drying and wetting in all benchmarks. In scenarios where wetting and drying needs to be considered the water height threshold will be given. When the limiter is said to be switched off only the Numerical Admissibility Detection criteria are meant. The constants $\epsilon$ and $\epsilon_0$ from chapter 4 are set to $10^{-3}$ and $10^{-4}$, similar to the introductory paper [9]. Time step size is always the maximal possible as allowed by the CFL condition.

Whenever the accuracy of the method is evaluated the errors for a discrete $p$-norm will be calculated on a set of nodes which is described when applied. The whole numerical errors are then evaluated as the sum of the single nodal errors

$$E_p = \sum_{i=1}^{N} \left\| q_h(x_i, y_i, t_i) - q_{\text{ref}}(x_i, y_i, t_i) \right\|_p,$$

where $q_{\text{ref}}$ is the analytic solution or if no analytic solution is known a numeric reference solution.

sam(oa)$^2$ provides a very friendly environment to develop shared memory and even MPI capable simulations. Unfortunately in this thesis the DG scheme only worked with shared memory. All simulations were performed on the MAC-Cluster (http://www.mac.tum.de/wiki/index.php/MAC_Cluster) on Intel SandyBridge-EP Xeon compute-nodes, where for each core one thread was chosen. As no deeper optimization of the simulation was done computation times are neglected in this analysis. Benchmarks like time to solution need to be analyzed in further studies.
6.1 Two Dimensional Convergence Test

As explained in section 3 one of the main advantages of the ADER-DG method is the convenience to reach high orders of convergence. To avoid disturbances through instabilities a benchmark with completely smooth solution, as introduced in [11] by Gallardo, was performed. The initial conditions in this benchmark are defined as

\begin{align*}
    h_0(x, y) &= 10 + e^{\sin(2\pi x) \cos(2\pi y)} \\
    hu_0(x, y) &= \sin(\cos(2\pi x)) \sin(2\pi y) \\
    hv_0(x, y) &= \cos(2\pi x) \cos(\sin(2\pi y)) \\
    b(x, y) &= 2 - \sin(2\pi x) - \cos(2\pi y),
\end{align*}

where the computational domain is the unit square \([0, 1]^2\) and periodic Dirichlet boundary conditions are assumed. For these initial conditions the solution is guaranteed to remain smooth until time \(T = 0.05\). The generic traversal order in \texttt{sam(oa)}^2 hinders an easy implementation of periodic boundaries, which is why the condition got artificial modeled by periodically extending the simulation domain up to a length where the condition boundary has no more influence on the solution. The benchmark considers a non-constant bathymetry and together with the smoothness of the solution it is the perfect test case to analyze the order of convergence of the unlimited ADER-DG method. The solutions for orders one to four were each compared to a result of the same order on a grid, having the number of maximal possible cells on a single compute node. The solutions got compared on a set of 2500 points at \(T = 0.05\), equidistantly distributed over the domain. Due to memory limitations these maximal number of cells were \(2^{20}, 2^{18}, 2^{16}\) and \(2^{14}\). All methods used the previously introduced HLLE-Riemann solver. To explicitly avoid the limiter taking action it was shut off.

Johnson and Pitkäranta showed in 1986 in the article [15] that on a uniform triangular grid with mesh size \(h\) for a hyperbolic PDE modeled by a DG method of order \(n\), a convergence order in \(O(h^{n+0.5})\) is to be expected. Fig. 9 shows the \(L_2\)-errors for all four orders displayed on a logarithmic scale. For a visual comparison the \textit{ideal} graph, which is the median error-measurement with logarithmic slope of the desired order of convergence, is hinted by a fine dotted plot in the same color as the measured errors for each method. The plot shows that all orders resemble the desired numerical convergence. The exact numerical errors for the \(L_1\), \(L_2\) and \(L_\infty\) norm are displayed together with the numerical convergence orders in table 1. The order for each measurement is computed by the logarithmic slope between itself and the previous measurement. Except a few outliers the analysis reveals that the numerical convergence equals or even outgoes the theoretical expectation. (As shown in [22] it is usual to measure a numerical order of \(O(h^{n+1}).\)"
<table>
<thead>
<tr>
<th>$h^{-1}$</th>
<th>$L_1$</th>
<th>$L_2$</th>
<th>$L_\infty$</th>
<th>$L_1$ conv.</th>
<th>$L_2$ conv.</th>
<th>$L_\infty$ conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.83</td>
<td>362.12</td>
<td>290.54</td>
<td>257.96</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3.36</td>
<td>213.01</td>
<td>200.32</td>
<td>181.91</td>
<td>2.30</td>
<td>2.15</td>
<td>2.02</td>
</tr>
<tr>
<td>4.00</td>
<td>149.91</td>
<td>118.83</td>
<td>102.75</td>
<td>2.79</td>
<td>3.01</td>
<td>3.30</td>
</tr>
<tr>
<td>4.76</td>
<td>96.04</td>
<td>77.69</td>
<td>68.81</td>
<td>2.57</td>
<td>2.45</td>
<td>2.31</td>
</tr>
<tr>
<td>5.66</td>
<td>56.43</td>
<td>44.22</td>
<td>38.04</td>
<td>3.07</td>
<td>3.25</td>
<td>3.42</td>
</tr>
<tr>
<td>6.73</td>
<td>39.33</td>
<td>31.24</td>
<td>27.33</td>
<td>2.98</td>
<td>2.00</td>
<td>1.91</td>
</tr>
<tr>
<td>8.00</td>
<td>24.70</td>
<td>19.15</td>
<td>16.36</td>
<td>2.68</td>
<td>2.82</td>
<td>2.96</td>
</tr>
<tr>
<td>11.31</td>
<td>12.73</td>
<td>9.87</td>
<td>8.41</td>
<td>1.91</td>
<td>1.91</td>
<td>1.92</td>
</tr>
<tr>
<td>16.00</td>
<td>7.33</td>
<td>5.71</td>
<td>4.90</td>
<td>1.59</td>
<td>1.58</td>
<td>1.56</td>
</tr>
<tr>
<td>22.03</td>
<td>4.51</td>
<td>3.63</td>
<td>3.20</td>
<td>1.40</td>
<td>1.31</td>
<td>1.23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$h^{-1}$</th>
<th>$L_1$</th>
<th>$L_2$</th>
<th>$L_\infty$</th>
<th>$L_1$ conv.</th>
<th>$L_2$ conv.</th>
<th>$L_\infty$ conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.38</td>
<td>292.84</td>
<td>241.46</td>
<td>218.45</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2.83</td>
<td>159.93</td>
<td>128.81</td>
<td>113.63</td>
<td>3.49</td>
<td>3.63</td>
<td>3.77</td>
</tr>
<tr>
<td>3.36</td>
<td>97.55</td>
<td>77.53</td>
<td>67.91</td>
<td>2.85</td>
<td>2.93</td>
<td>2.97</td>
</tr>
<tr>
<td>4.00</td>
<td>53.83</td>
<td>42.09</td>
<td>36.17</td>
<td>3.43</td>
<td>3.52</td>
<td>3.64</td>
</tr>
<tr>
<td>4.76</td>
<td>41.75</td>
<td>32.42</td>
<td>27.44</td>
<td>1.47</td>
<td>1.51</td>
<td>1.59</td>
</tr>
<tr>
<td>5.66</td>
<td>21.08</td>
<td>16.12</td>
<td>13.42</td>
<td>3.94</td>
<td>4.03</td>
<td>4.13</td>
</tr>
<tr>
<td>8.00</td>
<td>10.62</td>
<td>8.09</td>
<td>6.68</td>
<td>1.98</td>
<td>1.99</td>
<td>2.01</td>
</tr>
<tr>
<td>11.31</td>
<td>5.05</td>
<td>3.86</td>
<td>3.20</td>
<td>2.15</td>
<td>2.13</td>
<td>2.13</td>
</tr>
<tr>
<td>16.00</td>
<td>2.62</td>
<td>2.00</td>
<td>1.67</td>
<td>1.89</td>
<td>1.89</td>
<td>1.88</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$h^{-1}$</th>
<th>$L_1$</th>
<th>$L_2$</th>
<th>$L_\infty$</th>
<th>$L_1$ conv.</th>
<th>$L_2$ conv.</th>
<th>$L_\infty$ conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.00</td>
<td>885.80</td>
<td>849.91</td>
<td>833.18</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2.38</td>
<td>509.67</td>
<td>475.93</td>
<td>459.44</td>
<td>3.19</td>
<td>3.35</td>
<td>3.44</td>
</tr>
<tr>
<td>2.83</td>
<td>257.07</td>
<td>233.41</td>
<td>224.30</td>
<td>3.95</td>
<td>4.11</td>
<td>4.14</td>
</tr>
<tr>
<td>3.36</td>
<td>192.72</td>
<td>178.16</td>
<td>172.73</td>
<td>1.66</td>
<td>1.56</td>
<td>1.51</td>
</tr>
<tr>
<td>4.00</td>
<td>125.06</td>
<td>118.97</td>
<td>116.49</td>
<td>2.50</td>
<td>2.33</td>
<td>2.27</td>
</tr>
<tr>
<td>5.66</td>
<td>38.47</td>
<td>33.69</td>
<td>31.96</td>
<td>3.40</td>
<td>3.64</td>
<td>3.73</td>
</tr>
<tr>
<td>8.00</td>
<td>14.09</td>
<td>11.13</td>
<td>9.66</td>
<td>2.90</td>
<td>3.20</td>
<td>3.45</td>
</tr>
<tr>
<td>11.31</td>
<td>6.77</td>
<td>5.46</td>
<td>4.80</td>
<td>2.11</td>
<td>2.06</td>
<td>2.02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$h^{-1}$</th>
<th>$L_1$</th>
<th>$L_2$</th>
<th>$L_\infty$</th>
<th>$L_1$ conv.</th>
<th>$L_2$ conv.</th>
<th>$L_\infty$ conv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.68</td>
<td>1273.36</td>
<td>1206.98</td>
<td>1164.92</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2.00</td>
<td>531.80</td>
<td>487.97</td>
<td>468.86</td>
<td>5.04</td>
<td>5.23</td>
<td>5.25</td>
</tr>
<tr>
<td>2.38</td>
<td>528.35</td>
<td>496.87</td>
<td>480.66</td>
<td>0.04</td>
<td>0.10</td>
<td>0.14</td>
</tr>
<tr>
<td>2.83</td>
<td>345.15</td>
<td>345.95</td>
<td>343.19</td>
<td>2.46</td>
<td>2.09</td>
<td>1.94</td>
</tr>
<tr>
<td>3.36</td>
<td>172.08</td>
<td>165.81</td>
<td>162.79</td>
<td>4.02</td>
<td>4.24</td>
<td>4.30</td>
</tr>
<tr>
<td>4.00</td>
<td>95.34</td>
<td>89.12</td>
<td>86.93</td>
<td>3.41</td>
<td>3.58</td>
<td>3.62</td>
</tr>
<tr>
<td>5.66</td>
<td>26.26</td>
<td>22.77</td>
<td>21.44</td>
<td>3.72</td>
<td>3.94</td>
<td>4.04</td>
</tr>
<tr>
<td>8.00</td>
<td>8.69</td>
<td>6.86</td>
<td>5.95</td>
<td>3.19</td>
<td>3.46</td>
<td>3.70</td>
</tr>
</tbody>
</table>

Table 1: Absolute numerical errors and numerical convergence orders for the convergence test for ADER-DG methods of order one to four. Using the $L_1$ $L_2$ and $L_\infty$ norms and $h^{-1}$ as the inverted grid size.
Figure 9: Error plots for the convergence test showing the discrete $L_2$-error norm for ADER-DG methods of order one to four. All orders show the expected numerical convergence order.

6.2 Oscillating Lake

To study how symbiotic Finite Volume solver and Discontinuous Galerkin solver behave, the Oscillating Lake benchmark, as proposed in [1], was performed. Given is a parabolic bathymetry defined on a domain $\Omega = [-2, 2] \times [-2, 2]
\begin{align*}
b(x, y) &= 0.1 \cdot (x^2 + y^2).
\end{align*}

Having this bathymetry, a periodic solution is given for
\begin{align*}
h(x, y, t) &= \max \left(0, 0.05 \left(2x \cos(\omega t) + 2y \sin(\omega t)\right) + 0.075b(x, y)\right) \\
u(x, y, t) &= 0.5\omega \sin(\omega t) \\
v(x, y, t) &= 0.5\omega \cos(\omega t),
\end{align*}

where $\omega = \sqrt{0.2g}$.

The analytic solutions for $t = 0$ and $t = \pi / \omega$ are displayed in Fig. 10. It is a circle with radial symmetric water height propagating circular counterclockwise around the origin through an area of dried cells. The time variable only occurs within periodic trigonometric functions, solutions are equal for all $t \equiv \hat{t} \mod 2\pi / \omega$, where $\hat{t} \in [0, 2\pi / \omega]$.

For the combination of Finite Volume and ADER-DG solver this is a appropriate benchmark to show how harmonic both methods work. The area within the circle is smooth, so the water height threshold is set to $2.5 \cdot 10^{-4}$ to maximize the fraction handled by the ADER-DG method. Within dry
areas and along the dry-wet boundary the Finite Volume method should be used as solver. The circle propagates through the domain and a steady amount of cells changes from being calculated as Finite Volume solution to being solved by the DG scheme and back. At this point even small reoccurring errors within the conversion should be recognized. To maintain radial symmetry within the numerical solution, velocities for both methods have to be in step, as e.g. a slightly slower moving dry-wet front would result in a displacement of the circle’s middle point. These errors could also indicate an erroneous handling of bathymetry terms as these have direct impact on the resulting velocities. Also this benchmark gives insights into how well the solver conserves the mass of the solution, as a diffusion of water height would also be measurable after a certain amount of periods.

It is worth to mention that the choice for the augmented Riemann solver was essential, as all other solvers failed to handle velocities appropriately. The benchmark got performed for orders one to four for a grid size up to $2^{14}$. For orders higher than one small perturbations occurred after a certain amount of periods, which can be explained by the approximation of the volume flux and source integrals from chapter 3, an exemplary result for an order one and two simulation is given in Fig. 11. On the top left picture the initial value at time $T = 0$ is shown, while on the lower left the numerical solution for an order one simulation after six orbits $T = 12\pi/\omega$ is displayed. The solution conserves radial symmetry and the position of the circle. In fact the solution is almost identical to the initial value, showing periodicity of the numerical solution. On the top right the numerical solution for an order two simulation is shown. It is similar to the previous case, except the erroneous perturbations. To obtain a result as exact as for the order one
Figure 11: The top left shows the initial water height for the oscillating lake scenario. On the lower left is the numerical solution after six orbits of an order one ADER-DG scheme, remaining radial symmetry and the position of the solution. On the top right is the same solution for an ADER-DG scheme of order two, showing a similar result, except occurring perturbations. The lower right displays the limiter action plot after six orbits. Red indicates dried cells handled as Finite Volume Euler method, orange smooth areas which are unlimited and in blue the interface between both handling wetting and drying.
case a more accurate approximation of the volume integrals is needed. The lower right shows the limiter action plot after six orbits. It indicates in red, cells that are dry and got handled by the Finite Volume method, in orange, cells that where not limited and in blue, the interface between both, which are handling wetting and drying. As expected, areas within the circle are handled by the ADER-DG scheme as the solution is smooth here.

6.3 Radial Dam Break

The radial dam break scenario is the radial symmetric extension of the well known and analytically solved one dimensional linear dam break. In this thesis it is considered as a cylindrical water elevation of radius $\sqrt{4.5}$ placed on top of a cylindrical bathymetry elevation of radius $\sqrt{9.0}$.

$$h_0(x,y) = \begin{cases} 
15 & \text{for } ||(x,y)||^2_2 \leq 4.5 \\
5 & \text{for } 4.5 < ||(x,y)||^2_2 \leq 9.0 \\
10 & \text{else}
\end{cases}$$

$$b(x,y) = \begin{cases} 
5 & \text{for } ||(x,y)||^2_2 \leq 9.0 \\
10 & \text{else}
\end{cases}$$

$$hu_0(x,y) = 0, hv_0(x,y) = 0.$$ 

The computational domain is set to $[-10, 10]^2$ to avoid influence of the boundary condition. Similar to the linear dam break scenario the solution consists of a radial shock wave propagating outwards until it reaches the jump in the bathymetry approximate time $T = 0.1$ and a radial rarefaction wave propagating inwards until it hits the origin. A proof for the one dimensional case can be seen in [16] by Leveque. Using this benchmark along the development showed that discontinuities, which are usually sources of instabilities, even often caused the DG-Scheme to diverge. To test the capabilities and behavior of the limiter introduced in section 4, this benchmark is thus a suitable test.

After the jump in the bathymetry is reached, an analytically solution is no more known. This is why a Finite Volume Euler solution on $9 \cdot 2^{16}$ cells as reference was used instead. To consider the effect of both discontinuities, values were taken at time $T = 0.15$ on 2500 points equidistantly distributed within the $[-5, 5]^2$ square around the origin. The limiter is supposed to take action along the moving discontinuity and at the static discontinuity of the bathymetry, as it is known that instabilities occur at these positions due to Gibb’s phenomenon. In chapter 4.2.2 it was clarified that one point of interest is how the limiter behaves for varying considered physical quantities. As one source of unintended limiting is acceleration, the limiter got compared using all quantities, neglecting velocities and only considering water height, against the scheme being unlimited.
Figure 12: Extracted plot of the radial dam break scenario with water height on the right and the limiter action plot on the left at $T = 0.15$. In the water height plot the discontinuity of the bathymetry is hinted by a grey line. The action plot indicates for red and blue cells that the limiter took action at this time step. Blue cells where limited because of a discontinuous water height, red because of the bathymetry’s discontinuity. Orange indicates that the cell is unlimited.

Fig. 12 shows a comparison of water height and troubled cells for the inner section $[-5, 5]$ of the domain at time $T = 0.15$ for an order three simulation using limiting on all physical quantities on a grid of $2^{11}$ cells. The limiter takes action along the bathymetry’s discontinuity and the propagating shock wave. Cells having the initial states on the out- and inside are left unlimited. However also parts of the middle state get unnecessarily limited despite the area being smooth, which can be traced back to an acceleration in this cells.

The absolute $L_2$-errors for orders one to four, for different grid densities are shown in table 2. One directly sees that errors for almost every width get reduced by using one of both limiting types, though only limiting on height is not as effective as limiting on all physical quantities. The unlimited scheme is highly divergent even for high orders. This can also be seen in the plot in Fig. 13, where the $L_2$-errors for the order one scheme using both limiting types and no limiter are displayed. Using only water height as limiting indicator doesn’t stop the solution from diverging, conversely the all limiting version damps the solution enough to converge. For this benchmark a restriction of considered physical quantities is not sufficient, the previously mentioned unintended limiting needs to be tolerated.
Figure 13: Plot of the $L_2$-errors for the radial dam break benchmark. Shown are the errors for an unlimited, height limited and all limited order one ADER-DG scheme.

Table 2: Table showing the $L_2$-errors for the radial dam break benchmark. Displayed are the errors for unlimited, height limited and all limited order one to four ADER-DG schemes.
To see how well the scheme behaves for a real life application the Tohoku tsunami, as occurred on the 11th March 2011 was simulated. Reference is the data of several Deepocean Assessment and Reporting of Tsunami (DART) buoys, provided by the National Oceanic and Atmospheric Administration (https://www.ngdc.noaa.gov/hazard/dart/2011honshu_dart.html), where the water height displacement over a time of three days around the event was recorded. The external geoinformation for bathymetry and the magnitude 9.0 earthquake are included by ASAGI. The earthquake gets modeled by performing a displacement of the bathymetry in time. For more insights on ASAGI the reader is referred to [21]. The actual input for displacement and bathymetry is provided by Galvez et.al. in [12]. In [17] the same Tsunami using the same input is performed also within sam(oa)^2 using a Finite Volume Euler scheme.

As most of the domain of this scenario is at rest, a non regular static grid was used around the origin of the tsunami. A circle, having the source as center, of radius $2.0 \cdot 10^6$ was set to have a $2^8$ times finer grid then the rest of the domain. The water height threshold is set to 500m which ensures that areas along coasts are only simulated as Finite Volume scheme. Fig.
14 shows a comparison between the simulated Tsunami after one minute and the areas the limiter takes action in. Red areas indicate cells that got detected by the limiter as they fall below the water height threshold, orange areas are left unchanged and evaluated by the DG method, blue cells highlight the interface between limited and unlimited cells and need to be evaluated with both methods. One sees that the Honshu isle (in green on the left side) and its coast are completely considered as dried, the interface lies along the Japan trench (the dark blue curve on the left). The source itself lies within that interface and gets modeled by both methods.

The limiter was set off as the solution was expected to be smooth and disturbances through erroneous limiting should be avoided. It was able to perform the simulation for all orders from order one to four using a grid of up to 1724652 cells. The solution with highest accuracy (order 4 with the previously mentioned amount of cells) is displayed in Fig. 15 after 1, 5, 30 and 60 minutes for the higher resolved part of the domain. After 1 and 5 minutes one can see the initial wave height of 15 meters and the symmetric separating of the wave. The last two show the for Tsunamis characteristic leading N-wave. By comparing with Fig. 14 one sees that the Tsunami wave propagates through limited areas, the presented solution is thus a mixture of both methods. To compare the quantitative accuracy of the simulation in Fig. 16 a comparison of the recorded water height of two buoys are plotted against the simulated results at the corresponding points. The first lies 1400 km, the second 3270 km away from the earthquake. The comparison shows that the simulation is accurate in time and amplitude, though the negative peak of the N-wave in the simulation overshoots the real value. It is also necessary to mention that at the interface between the fine part and the more coarsen part of the grid, unintended waves were produced when the Tsunami passed. As the shown domain is completely within the fine grid this has no effect on the here presented solutions.
Figure 15: The simulated Tohoku Tsunami after 1, 5, 30 and 60 minutes (row wise from top left to bottom right). The red dot in the last two pictures indicates the approximate position of buoy 21418. Simulated with ADER-DG order 4 on a grid of 1724652 cells.
Figure 16: Buoy data for the Tohoku tsunami on the 11th of March 2011, the t-axis displays days in a year. Blue are the measured wave height values for two single buoys, red the wave heights as numerical calculated at their locations with an ADER-DG order four scheme on a grid of 1724652 cells.
7 Conclusion and Future Work

In this thesis a realization of the ADER Discontinuous Galerkin method got included with a new a posteriori Finite Volume Limiter approach into the sam(oa)$^2$ framework and applied on various benchmarks for the Shallow Water Equations. After the preliminaries where introduced in section 1 the mathematical theory, the implementation is based on, got developed. In section 2 an alternative polynomial basis to approximate the solution was introduced. Also useful properties of this basis for the concrete implementation were derived. After working out the full ADER-DG scheme in section 3 with its time integration by the DG-Predictor, a new a posteriori Finite Volume limiter approach was presented in section 4. This limiter approach uses a set of new detection criteria, for which examples for intended and unintended limiting were explained. After the theoretical part of this thesis results for benchmarks got shown in section 6 where tests for convergence and the effect of the limiter got performed. Additionally the oscillating lake scenario, which places high demands on the method and the limiter, got simulated. In the end the Tohoku Tsunami from 2011 got simulated as a real life test case.

It was possible to prove that the ADER-DG approach works on the set of Bernstein polynomials. Also the requirements on convergence are fulfilled by this scheme. The new limiter approach was successfully tested for two benchmark cases. The radial dam break showed that only considering water height for the relaxed discrete maximum principle was not sufficient. The simulation of the oscillating lake benchmark was able show that limiter and DG method work in step, even for a high demanding scenario.

The developed scheme states a promising base for future work. As there have not been any attempts in optimizing the computational time nor the memory usage, these need to be the first points regarded. As the number of limited cells is expected to be only a small fraction of the whole grid a full traversal, as implemented in this thesis, is very expensive. At this point an alternative approach, like a ring buffer, would be an improvement. Optimization also includes making the method compatible with the MPI-integration of sam(oa)$^2$. After the method is capable of strong scaling, benchmarks can be performed on a lager grid to validate previous results. A comparison of time to solution of the Discontinuous Galerkin method and a Finite Volume method to see in which field which method is superior to the other, would be a desirable result.

From a mathematical point of view an error- and stability comparison between the Quasi-Nodal basis and other bases is an interesting task.
Bibliography


