DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics: Games Engineering

Tsunami simulation in the ExaHyPE framework

Lisa Scheller

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics: Games Engineering

Tsunami simulation in the ExaHyPE framework

Tsunamisimulation innerhalb des ExaHyPE-Frameworks

Author:	Lisa Scheller
Supervisor:	Prof. Dr. Michael Bader
Advisor:	MSc Leonhard Rannabauer
Submission Date:	18.12.2017

I confirm that this bachelor's thesis in informatics: games engineering is my own work and I have documented all sources and material used.

Munich, 18.12.2017 Lisa Scheller

Acknowledgments

Abstract

Cell size, $x_{i+1/2} - x_{i-1/2}$

List of Abbreviations

ODE ordinary differential equation

PDE partial differential equation

Nomenclature

$f'(q)$ Jacobi matrix of $f(q)$

q vector of unknowns

q_x partial derivative $\frac{\delta q}{\delta x}$

Contents

1 Introduction

1.1 Section

1.1.1 Subsection

2 The shallow water equations

2.1 Hyperbolic partial differential equations

2.1.1 Definition

Partial differential equations (PDEs) are used to describe a multitude of problems. In contrast to ordinary differential equations (ODEs) PDEs contain partial derivatives, whereas ODEs only contain derivatives of one variable. There are a multitude of phenomenons that can be described by PDEs, ranging from fluid mechanics and traffic simulation over thermodynamic problems to quantum mechanics. In this bachelor's thesis we restrict ourselves to a certain subclass of PDEs, the so-called hyperbolic PDEs. The core equations of this thesis, the shallow water equations, which will be introduced in detail in the next section,as well as the standard advection equation, are hyperbolic PDEs.

Initially let

$$q_t + Aq_x = 0 \tag{2.1}$$

be a one-dimensional, homogenous, first-order system of PDEs. q is a vector from $\mathbb{R}^m$ that contains the unknowns of the equations and A a $m \times m$ matrix. If this equation shall describe a system of hyperbolic PDEs, the matrix A has to meet certain conditions. Firstly, the matrix has to be diagonizable [Lev04], thus it must be possible to describe the matrix in the following way:

$$A = S * D * S^{-1} \tag{2.2}$$

, where D is a matrix that is zero for all entries not on the diagonal, and S is a transformation matrix. Secondly, the eigenvalues of A have to be real. This leads to the following definition:

Definition 2.1.1. "A linear system of the form

$$q_t + Aq_x = 0$$

is called hyperbolic if the $m \times m$ matrix A is diagonizable with real eigenvalues."[Lev04]

Other types of PDEs are elliptic or parabolic PDEs.

2.1.2 Classification

There are several sub-groups of hyperbolic PDEs, in regard to the eigenvalues of A as well as in regard to the structure of the equations.

Eigenvalues

Since a symmetric matrix is always diagonizable with real eigenvalues, all symmetric matrices can be used to describe a hyperbolic problem [Lev04]. These equations are called symmetric hyperbolic.
If the matrix has distinct eigenvalues and is diagonizable, the system of PDEs is called strictly hyperbolic. [Lev04]
In contrast, a matrix that is not diagonizable but has real eigenvalues is called weakly hyperbolic.[Lev04]

Structure of equations

In the simplest case of a hyperbolic PDE A is not a matrix, but a constant scalar. Equation (2.1) simplifies to

$$q_t + aq_x = 0 \tag{2.3}$$

and a has to be real for the system to be hyperbolic.
If the scalar a or the matrix A are time-dependent or depend on another value (e.g. $a(x,t)$, $A(x,p)$), they have to fulfill the criteria for a system of hyperbolic PDEs for each value of t (or in the examples also of x and p).

Conservation laws A special group of hyperbolic PDEs are called conservation laws. In the simplest case (1 dimension) they have the following structure:

$$q(x,t)_t + f(q(x,t))_x = 0 \tag{2.4}$$

or in the quasilinear form

$$q_t + f'(q)q_x = 0 \tag{2.5}$$

[Lev04] $f(q)$ can be linear as well as non-linear in regard to q.
However, it is often necessary to use the integral formulation instead of (2.4). This is caused by the formation of discontinuities, which are often seen in the solutions of hyperbolic PDEs.

The integral formulation The main benefit of using the integral formulation is that its validity is not impacted by discontinuities. Since handling discontinuities is an important part of solving hyperbolic PDEs, it is vital to always be certain if the differential formulation can be used or not.

The integral formulation of a hyperbolic PDE looks as follows:

$$\frac{d}{dt}\int_{x_1}^{x_2} q(x,t)\,\mathrm{d}x = F(x_2,t) - F(x_1,t) \tag{2.6}$$

$f(x,t)$ represent fluxes going either into or out from the element. The differential formlation ((2.4)), which is used mostly, can be derived from the integral formulation. [Lev04]

2.1.3 Examples

The complexity of hyperbolic PDEs ranges from relatively simple, with known exact solutions, to very complex with no known analytic solutions. One example of a very basic hyperbolic PDE is the one-dimensional advection equation with constant speed:

$$q_t + uq_x = 0 \tag{2.7}$$

, where u is the constant velocity. This equation is solved by

$$q(x,t) = q_0(x - ut) \tag{2.8}$$

, where $q_0(x) = q(x,0)$ are the known initial conditions.

Even though this equation is very simple, many problems can be put into a formulation resembling it. So it plays a role even at solving more difficult problems.[Lev04]
Another, more complex example are the Maxwell equations

$$\begin{aligned} E_t &= \nabla \times B \\ B_t &= -\nabla \times E \end{aligned} \tag{2.9}$$

([Stu13]), which are used to describe electromagnetic fields. Most problems which are solved by waves can be described by hyperbolic PDEs. Of course this also includes the description of water waves. Because of this the shallow water equations, which will be covered in detail in the next chapter, are also hyperbolic.

2.1.4 Numerical difficulties

The main problem with solving hyperbolic PDEs is that the solutions (or even the initial conditions) often contain discontinuities, which are difficult to solve. Especially methods that assume the solution is sufficiently smooth often fail at providing the correct solution. Another problem is that the solutions often can be described as several waves of different types. The composition of types depends on the problem and has to be computed first. If the waves interact which each other, it complicates the solution further.

2.2 The shallow water equations

This thesis focuses on solving the shallow water equations in one dimension, which describe water height and momentum of a fluid. Their main characteristic is that the vertical scale is much smaller than the horizontal scale. This is true for ocean waves and even tsunami waves, which are also described by the shallow water equations.

2.2.1 Equations in one dimension

The one-dimensional shallow water equations consist of two equations. The first one is:

$$h_t + (uh)_x = 0 \tag{2.10}$$

and the second one, which is a little bit more complicated, is:

$$(hu)_t + (hu^2 + \frac{1}{2}gh^2)_x = 0 \tag{2.11}$$

If $q(\vec{x}, t) = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} h \\ hu \end{pmatrix}$ is introduced and $f(q) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} q_2 \\ (q_2)^2/q_1 + \frac{1}{2}gq_1 \end{bmatrix}$ is defined, then the equations can be written as

$$q_t + f(q)_x = 0 \tag{2.12}$$

For smoot solutions, the formulation

$$q_t + f'(q)q_x = 0 \tag{2.13}$$

is also valid.[Lev04]

2.2.2 Properties

The eigenvalues and eigenvectors of $f'(q)$ are important properties of the equation. They have to be known to be able to compute a numeric or exact solution to any problem. Fortunately, they are quite simple to compute. They are:

$$\lambda_1 = u - \sqrt{gh} \tag{2.14}$$

and

$$\lambda_2 = u + \sqrt{gh} \tag{2.15}$$

This leads to the eigenvectors

$$\vec{v_1} = \begin{pmatrix} 1 \\ \lambda_1 \end{pmatrix} \tag{2.16}$$

and

$$\vec{v_2} = \begin{pmatrix} 1 \\ \lambda_2 \end{pmatrix} \tag{2.17}$$

[Lev04].
Because the eigenvalues (and thus the eigenvectors) depend on h and u, which are the unknowns of the equation system and can be different for every point in the domain, every of these points has different eigenvalues. It is not sufficient to compute one set of eigenvalues at the beginning and then never change it again, like with the constant velocity advection equation. The eigenvalues have to be computed for each point or cell in the domain and for each timestep.
If a comparison is made between the advection equation and the shallow equations (both for a hump of water in gaussian form), the main difference is that using the advection equation the wave doesn't change shape and just travels around, whereas using the shallow water equations a change of shape can be observed.

2.2.3 Example problems

There are to types of problems that will be approximated by applying the shallow water equations. The first one has continuous starting conditions and is the already descriped gaussian water hump. The second one is the dam break problem, a so called Riemann problem with initial velocities zero and a difference in inital height.

Riemann problems

Definition 2.2.1. A problem with inital conditions of the form

$$q(x,0) = \begin{cases} q_l & x \leq 0 \\ q_r & 0 \leq x \end{cases}$$

is called a Riemann problem[Lev04].

The main characteristic of a Riemann problem is the discontinuity. To solve such a problem analytically, it has to be fragmented into several states. Two of them are the outer left and right states, where the solution takes the value q_l or q_r respectively. The middle states depend on the exact composition of the problem, the possibilities for the cases viewed here are either 2 shock waves, 2 rarefaction waves or one of each. In every case a middle value q_m has to be computed. The exact way to solve a Riemann problem as well as an explanation of the different types of waves will be given in chapter 3. The analytical solution for a dam-break problem will also be computed.

2.2.4 Two-dimensional equations

In two dimension the shallow water equations can be written as

$$\begin{aligned} h_t + (hu)_x + (hv)_y &= 0 \\ (hu)_t + (hu^2 + \frac{1}{2}gh^2)_x + (huv)_y &= 0 \\ (hv)_t + (huv)_x + (hv^2 + \frac{1}{2}gh^2)_y &= 0 \end{aligned} \tag{2.18}$$

[Lev04] This differs from the one-dimensional form by the addition of a additional dimension y. Also the velocity is now described by a vector, $\vec{u} = \begin{pmatrix} u \\ v \end{pmatrix}$. The equations look very similar to and can be solved using similar techniques as for the one-dimensioal case, however for the sake of complexity this thesis will only cover the one-dimensional equations.

3 Methods for solving hyperbolic partial differential equations

There are a multitude of numerical methods to solve PDEs. However, some of them are more suited to the task of solving hyperbolic PDEs than others. As pointed out earlier especially discontinuities break some methods. So it's important to choose a method that handles these discontinuities well if a hyperbolic PDE has to be solved numerically.

3.1 Finite Differences

3.1.1 General description

The finite differences method is one of the better known numerical schemes. It approximates derivatives by replacing them with a difference quotient of the form

$$\frac{u(x+h) - u(x)}{h} \tag{3.1}$$

and solving the resulting equation for $u(x+h)$. The result can then be integrated by using an integration scheme such as explicit Euler etc. The domain is divided into intervals of length h, for which the equation is evaluated. Higher accuracy can be achieved by choosing a smaller h or using a higher order integration scheme such as Runge-Kutta.

The computational domain is divided so that the method can be evaluated at certain pre-defined grid points.

3.1.2 Advantages and disadvantages

Advantages The method is easy to understand and since only one value is computed per interval it can also be very efficient[JSH08]. Also it is possible to modify the scheme to get high order accuracy.

Disadvantages However since the finite difference becomes quite nonsensical at a discontinuity it is not well suited for solving hyperbolic PDEs. Also it doesn't allow more complex geometry because it only supports equidistant intervals [JSH08]. This becomes even more of a problem for solving higher dimensional problems.

3.1.3 Example

3.2 Finite Volumes

3.2.1 General description

The main difference between the method of finite volumes and the method of finite differences is that the first is based on the integral form, whereas the latter is based on the differential form. Using the integral form makes it possible to handle discontinuities. The computational domain is divided in cells, and for each cell the integral over this cell is approximated. For each cell, the cell average is computed, which is the quotient of the approximated integral and the cell volume [Lev04]. This means that each cell stores only this average and the changing values inside a cell are not stored.

To evaluate the change over time now also the fluxes from the neighbor cells have to be considered. There are several possibilities to compute those fluxes, some of which will be explained later.

From now on, the description will be restricted to only one dimension.
The basic idea is to describe the ith grid cell as

$$C_i = (x_{i-1/2}, x_{i+1/2}) \tag{3.2}$$

[Lev04] The cell average of the ith cell in the nth iteration can now be approximated as:

$$Q_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} q(x, t_n) dx = \frac{1}{\Delta x} \int_{C_i} q(x, t_n) dx \tag{3.3}$$

[Lev04], where Δx is the cell size.
If this is applied to the integral formulation of a conservation law, it leads to the following equation:

$$\frac{d}{dt} \int_{C_i} q(x, t) dx = f(q(x_{i-1/2}, t)) - f(q(x_{i+1/2}, t)) \tag{3.4}$$

Since the goal is to get a way to get from the known Q_i^n at time t^n to the values of Q_i^{n+1} at time t^{n+1}, the above equation has to time-integrated. If the result of this is then divided by Δx it leads to an approximation of exactly this.

$$\frac{1}{\Delta x}\int_{C_i} q(x,t^{n+1})dx = \frac{1}{\Delta x}\int_{C_i} q(x,t^{n})dx - \frac{1}{\Delta x}[\int_{t^n}^{t^{n+1}} f(q(x_{i+1/2},t))dt - \int_{t^n}^{t^{n+1}} f(q(x_{i-1/2},t))dt] \tag{3.5}$$

[Lev04]. While this equation would compute exactly what we want it to compute, it can't normally be used because the time integrals on the right side are usually not known. Usually schemes of the following form are used:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}(F_{i+1/2}^n - F_{i-1/2}^n) \tag{3.6}$$

F is a approximation to the time integrals in (3.5). The choice of this flux approximation heavily impacts the quality of the approximated solution. $F_{i-1/2} = F((Q_{i-1}, Q_i))$ is called a numerical flux function and it can now be seen that the approximation of Q_i^{n+1} depends on Q_{i-1}, Q_i and Q_{i+1}, a so called three point stencil [Lev04].

3.2.2 Advantages and disadvantages

Advantages One of the main benefits of the method of finite volumes is that it is able to handle problems with discontinuities. Also it is possible to use varying cell sizes, which is useful for more complex geometries. Since the solution depends only on the inital conditions and the fluxes one of the main tasks is to use the flux functions most well-suited for the specific problem. This depends on required accuracy, stability and also on the properties of the problem itself.

Disadvantages Since only the cell average is stored for each cell, the approximation inside of one cell is quite rudimentary. It would be more accurate to give the cell more information, e.g. a value at the left boundary and the right boundary of the cell as well as a interpolation function to link those values. The inaccuracy of the cell average can of course be lessened by making the cells smaller, but it is still not as accurate as for example the discontinuous Galerkin method, which wil be covered later.
Also if a higher order of accuracy shall be reached, the restrictions on the grid come back again, as described in [JSH08]. This means the method loses one of it's main advantages. If these disadvantages are a problem for the specific problem that should be solved, and a finite difference scheme is also not the right solution, it might be necessary to take a look at finite elements methods.

3.2.3 The CFL condition

The stability of a finite volume method depends heavily on the chosen timestep. If it is to big, the changed information has travelled too far and so the solution depends not only on the neighboring cells but on their neighbors etc. as well. Because those are not part of the numerical scheme the results will become increasingly wrong with time. So a condition to determine the timestep size is necessary.
The Courant-Friedrichs-Lewy-condition (CFL-condition) is such a condition.

Definition 3.2.1. "A numerical method can be convergent only if its numerical domain of dependence contains the true domain of dependence of the PDE, at least in the limit as Δt and Δx go to zero"[Lev04]

The CFL condition is a necessary condition for stability and convergence. It leads to the conclusion that the Courant number

$$v = \frac{\Delta t}{\Delta x} max|\lambda^p| \tag{3.7}$$

, where the λ^p are the wave speeds of the system of equations, has to stay below a certain bound, usually 1 for the finite volume methods we use. From this equation the biggest timestep that still leads to a stable solution can be computed by solving it for Δt. Depending on the problem (for problems with variable wavespeeds) it is not possible to use a stable timestep, since it would become bigger than the maximum allowed timestep over time and so the timestep for the next time iteration has to computed from the wave speeds at the respective time.

3.2.4 Flux approximations

The choice of the flux function is a major part of the finite volumes scheme. The following section will give short overview over some flux functions. Most of these fluxes where also implemented in the program.

Unstable Flux This method follows a very simple approach. The flux function is approximated by

$$F^n_{i-1/2} = \frac{1}{2}[f(Q^n_{i-1}) + f(Q^n_i)] \tag{3.8}$$

This would lead to

$$Q^{n+1}_i = Q^n_i - \frac{\Delta t}{2\Delta x}[f(Q^n_{i+1}) - f(Q^n_{i-1})] \tag{3.9}$$

However, this approach is unusable because it is unstable for hyperbolic problems and all timestep sizes [Lev04].

Lax-Friedrichs flux This flux looks very similar to the unstable flux, but it is different enough that it leads to a stable result. It has the advantage that it is still a very simple method that is easy to implement. The flux function is

$$F_{i-1/2} = \frac{1}{2}[f(Q^n_{i-1}) + f(Q^n_i)] - \frac{\Delta x}{2\Delta t}(Q^n_i - Q^n_{i-1}) \tag{3.10}$$

This can then be used in (3.6) to compute the value of Q_i at the next timestep.
While the Lax-Friedrichs flux leads to a stable scheme the order of accuracy still isn't too good, as will be outlined later when the results of the Shallow Water Equations computated with this flux will be compared to the analytic solution.

Richtmyer Two-Step Lax–Wendroff Method A better order of accuracy can be achieved by evaluating q at time $t + \frac{1}{2}\Delta t$ and using this value to compute the fluxes at time $t + \Delta t$. However, this method is less performant than the one-step methods, because for each timestep two values of Q have to be computed.

The local Lax-Friedrichs-Flux The main difference between the local Lax-Friedrichs method is that the factor $\frac{\Delta x}{\Delta t}$ in the computation of the fluxes is now replaced with another value. In particular this value is not fixed, but changes for every cell and every time. The method for determining the fluxes becomes

$$F_{i-1/2} = \frac{1}{2}(f(Q_{i-1}) + f(Q_i) - a_{i-1/2}(Q_i - Q_{i-1})) \tag{3.11}$$

and the factor a is determined by

$$a_{i-1/2} = max(|f(Q_{i-1}|, |f(Q_i)|) \tag{3.12}$$

For the shallow water equations this leads to the biggest eigenvalue (which are $u \pm \sqrt{gh}$) at both indexes (i and i-1 for the left side and i and i+1 for the right side). Using the local Lax-Friedrichs flux leads to better results especially concerning rarefaction waves [Lev04]. This method is also known as Rusanov's method. The name "local Lax-Friedrichs method" is based on the similarity to the Lax Friedrichs method and the choice of different factors for each cell, which makes the method local.

Upwind methods

3.3 Finite Elements

3.3.1 General description

The method of finite elements divides the domain in several elements. Even if the general geometry is very complex, the geometry of the elements is a lot simpler and makes it possible to approximate the solution inside of this elements. The solution inside the elements is approximated by local basis functions [JSH08]. The approximated solution is easier to compute for a lower degree of these basis functions, but the accuracy increases with the degree of those functions. A popular choice for the basis functions are the Lagrange polynomials,

$$\ell_i^k(x) = \frac{x - x^{k+1-i}}{x^{k+1} - x^{k+1-i}} \tag{3.13}$$

It is of course possible to choose other interpolation polynomials.
The global formulation is retrieved by choosing a set of test functions that are orthogonal to the residual of the problem. For a Galerkin scheme the test functions span the same spaces as the basis functions. Other choices of test functions are possible. For the standard advection equation this leads to a scheme of the form

$$M\frac{du_h}{dt} + Sf_h = 0 \tag{3.14}$$

where

$$M_{ij} = \int_\Omega N^i(x)N^j(x)dx \tag{3.15}$$

is the globally defined mass matrices (N^i are the basis functions) and

$$S_{ij} = \int_\Omega N^i \frac{dN^j}{dx}dx \tag{3.16}$$

is the globally defined stiffness matrix [JSH08].

3.3.2 Advantages and disadvantages

Advantages This approach has the advantage that it is possible to choose different orders of approximation in different cells [JSH08] and so adapt to the specificiations of the problem. Also the choice of the elements size and geometry is very flexible.

Disadvantages However, the above choice of basis functions and the requirements they must fulfill means that the matrix M has to be inverted and the scheme becomes implicit [JSH08]. Also the choice of symmetric basis functions is not helpful for problems which prefer a certain direction, e.g. wave problems [JSH08].

3.4 Discontinuous Galerkin methods

3.4.1 General description

As the previous sections show, both finite volume methods and finite element methods have some strenghts, but also some weaknesses. The discontinuous Galerkin method is a method that tries to combine these strenghts and eliminate the weaknesses as good as possible.
It uses the same elements as a finite elements method, but the inner nodes are duplicated [JSH08]. The cells/elements are then (in the one-dimensional case) limited to the left by u_i and to the right by u_{i+1}. For example the first cell stretches from u_0 to u_1, while the second cell goes from u_1 to u_2. This shows why the inner nodes have to be duplicated: The value at the right end of cell 1 doesn't have to be the same value as the left end of cell 2. This means the solution can be discontinuos at the cell interfaces. Inside an element the solution is approximated as

$$u_h^k(x) = \sum_{i=0}^{1} u^{k+i} \ell_i^k(x) \tag{3.17}$$

as well as the flux, as explained in [JSH08].
To determine the test functions, the local residual (for a scalar conservation law)

$$R_h(x,t) = \frac{\delta u_h^k}{\delta t} + \frac{\delta f_h^k}{\delta x} \tag{3.18}$$

has to be orthogonal to the test functions, this means the test functions have to fulfill the following equation:

$$\int_{D^k} R_h(x,t)\ell_j^k(x)dx = 0 \tag{3.19}$$

Applying Gauss' theorem and introducing a numerical flux f^* leads to the equation

$$\int_{D^k} R_h(x,t)\ell_j^k(x)dx = [(f_h^k - f^*)\ell_j^k]_{x^k}^{x^{k+1}} \tag{3.20}$$

Like with finite volumes the choice of the numerical flux is an important part of the method. It is also possible to convert this equation so that it mimics the matrix formulation of finite element methods. The mass and stiffness matrixes are computed analogous to this.

3.4.2 Nodal and modal formulation

There a two ways to express the solution when using a discontinuous Galerkin method. These two ways are called the nodal and the modal form.

Nodal form The nodal form uses the values at the grid points and interpolates the function between them by using some form of interpolation polynomial [JSH08]:

$$u_h^k = \sum_{i=1}^{N_p} u_h^n(x_i^k, t)\ell_i^k(x) \tag{3.21}$$

In this case the interpolation polynomials are the Lagrange polynomials of order $N_p + 1$.

"The global solution u(x, t) is then assumed to be approximated by the piecewise N-th order polynomial approximation $u_h(x, t)$,

$$u(x,t) \simeq u_h(x,t) = \oplus_{k=1}^{K} u_h^k(x,t) \tag{3.22}$$

defined as the direct sum of the K local polynomial solutions $u_h^k(x, t)$."([JSH08], p.21)

Modal form The modal form uses a local polynomial basis [JSH08]:

$$u_h^k(x,t) = \sum_{n=1}^{N_p} {}_n^k(t)\psi_n(x) \tag{3.23}$$

where $\psi_n(x)$ is the polynomial basis function and N_p is the polynomial order plus one.

3.4.3 Advantages and disadvantages

Advantages The discontinuous Galerkin method makes it possible to achieve higher order accuracy even on unstructured grids. Also it can make use of different flux functions, making it possible to choose a flux tailored to the specific problem that shall be solved [JSH08]. Choosing the right flux makes it also possible to reduce oscillations in comparison with finite element methods [JSH08].

Disadvantages The computational work is a lot higher than in the simpler methods. This is caused by the doubling of the nodes but also by the more complex numerical computation. Especially with high-order interpolation and thus bigger matrices to compute this is an issue. Also, of course, for problems that don't need the advantages of the discontinuous Galerkin method and are not hit especially hard by the problems of other methods, this computational effort is often not worth it.

3.4.4 Comparison between the methods

The following figure gives an overview over the strenghts and weaknesses of each of the methods we analyzed so far. It can be seen that the discontinuous Galerkin method is a good combination of the strengths of finite volume methods and finite element methods.

Figure 3.1: Comparison of the advantages and disadvantes of different methods to solve hyperbolic PDEs, source [JSH08]

	Complex geometries	High-order accuracy and *hp*-adaptivity	Explicit semi-discrete form	Conservation laws	Elliptic problems
FDM	×	✓	✓	✓	✓
FVM	✓	×	✓	✓	(✓)
FEM	✓	✓	×	(✓)	✓
DG-FEM	✓	✓	✓	✓	(✓)

3.5 Overview over the integration methods

3.5.1 Explicit Euler

3.5.2 Implicit Euler

3.5.3 Heun's method

3.5.4 Leap-frog method

3.5.5 Runge-Kutta method

3.6 Analytical solution of the dam-break problem

3.6.1 General strategy

3.6.2 Specific solution

4 Implementation

5 Conclusion

List of Figures

List of Tables

Bibliography

[JSH08] T. W. Jan S. Hesthaven. *Nodal Discontinuos Galerkin Methods - Algorithms, Analysis, and Applications*. Texts in Applied Mathematics. Springer, 2008. ISBN: blub.

[Lev04] R. J. Leveque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2004. ISBN: bla.

[Stu13] U. Stuttgart. *Skript Partielle Differentialgleichungen II*. 2013.