SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Implementation and Validation of a Modified Shallow Water Model with Topography for Landslide Simulations in ExaHyPE 2

Alexander Leon Heinz Wachenfeld

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Implementation and Validation of a Modified Shallow Water Model with Topography for Landslide Simulations in ExaHyPE 2

Implementation und Validierung eines modifizierten Shallow Water Modells mit Topographie für Landslide Simulationen in ExaHyPE 2

Author:	Alexander Leon Heinz Wachenfeld
Supervisor:	Univ.-Prof. Dr. Michael Georg Bader
Advisor:	Mario Wille, M.Sc.
Date:	17.03.2025

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

A. Wachenfeld

Munich, 17.03.2025 Alexander Leon Heinz Wachenfeld

Abstract

The topic of this thesis is landslide and water simulations with a focus on their numerical aspects. This thesis starts by concisely summarizing the relevant governing equations in the form of the basic two-dimensional shallow water equations. In the case of landslides, this thesis expands these basic equations by incorporating a Coloumb-type friction term. Thus, it treats landslides as dry one-layer granular flows. This work then pays further attention to approximate methods to solve these equations. Therefore, both the finite volume method and the Riemann problem are introduced. After this theoretical summary, the thesis starts evaluating different numerical methods, namely approximate Riemann solvers. This work investigates the simple but robust Rusanov solver, the more sophisticated HLLC Solver, and the more complicated augmented Riemann solver. Afterwards, the thesis considers common problems with these solvers, e.g., wetting and drying mechanisms and well-balancedness, and numerical solutions to handle them robustly. After evaluating these numerical aspects, this thesis will demonstrate their feasibility and correctness through a series of different numerical validation cases. These cases, as well as the respective solvers, are implemented in ExaHyPE 2. This implementation comprises both shallow-water and landslide scenarios. The numerical results will be compared to either the analytical solution or be evaluated based on physical feasibility. The outllok will describe further developments and experiments to improve the respective solvers and scenarios.

Contents

1 Introduction

There are many different natural and man-made phenomena that are endangering surrounding landscapes. Such phenomena, like landslides or tsunamis, are natural disasters that can have serious consequences for closely located areas surrounding the location of occurrence, including the humans living there [4]. These events can behave quite differently depending on the topographical structure beneath them, their composition, and the relevant physical forces. As such, the whole physical process is quite complex. Therefore, their dynamic behaviour is difficult to predict, and complete analytical analysis is not feasible. Thus, it is paramount to utilize simplified models and integrate them and the corresponding theoretical knowledge into numerical simulations. This can help to better understand the respective physics, which is important, as understanding both their occurrences and their flows' evolution is very important for human and environmental safety. Motivated by both the complexity and importance of these processes, many different scientific disciplines cooperate in order to research and model them. Applied mathematicians and physicists develop mathematical models that are supposed to explain these phenomena' behavior, often making assumptions to make the analytical investigations feasible. Engineers are interested in areas like the structural safety of dams and the environmental effects of corresponding catastrophes like dam breaks. Computer scientists are especially interested in numerical methods to solve the equations posed by the respective model and the computational simulations. This thesis focuses on phenomena based on water and granular material flows that are assumed to be of shallow height relative to their horizontal extent. A simplified model that describes shallow water flows is the shallow water equations. These can then comprise the basis for computational simulations of diverse scenarios. Extending these formulae with further physical terms, like friction coefficients, can also make them more applicable to material flows like landslides. Another basic assumption for all flows investigated is that they consist of a single layer, making their analysis much more feasible. However, even these simplified equations can often not be solved analytically for complex scenarios, at least not in a feasible way. This difficulty necessitates approximate methods that can handle them. The finite volume method is a feasible approach to execute these simulations. It is constructed based on conservation laws. It discretizes both the respective time and space dimensions, leading to the general finite volume formula. This formula always computes a subsequent time step, starting at the initial condition of the respective scenario. The finite volume formula incorporates a numerical flux and source term. The flux describes the physical interaction between the respective faces of the control volumes. The source term incorporates physical terms like geometric sources caused by topography and friction. In order to approximate the numerical flux, so-called Riemann solvers are used. These assume constant initial states and compute the resulting numerical flux updates of the neighbouring cells. As will be explained in the following chapters, the Riemann solvers can also incorporate the source terms. If not, it will be treated separately. This thesis will discuss different Riemann solvers, their properties, and how they can handle different numerical problems, like inundation or numerical instabilities. Especially the topic of inundation with wetting and drying fronts will be investigated, as the granular material of landslides is often in a constant inundation process. The thereafter following part of the thesis evaluates different scenarios based on these numerical methods. The scenarios will be described as Initial-Value-Boundary problems. Both artificial simplified scenarios, like idealized dam breaks or oscillatory movements, as well as real-world scenarios, like the Tafjord landslide, are investigated. This thesis evaluates both water flows without friction as well as material flows with a Coulomb friction term involved. All the solvers and different scenarios will be implemented in the ExaHyPE 2 framework, which allows specifying the scenarios as Python scripts and employing different individual Riemann solvers. The assessment of the quality of these numerical solutions uses either the analytical solution, experimental results, or physical feasibility as the relevant benchmark. Finally, further improvements and extensions that can improve the numerical solutions are discussed in outlook.

2 Related Work

The topic of this thesis is the simulation of granular, and water flows over varying topographies with and without friction. As stated before, the simplified governing equations investigated in this work are hyperbolic partial differential equations of the first order. In this thesis, all implementations will be made and simulations executed utilizing the ExaHyPE 2 software. The ExaHyPE 2 engine allows for solving first-order hyperbolic differential equations. More information about it can be found in [20]. For this work, it is relevant that the system allows for numerically solving first-order FV schemes on Cartesian meshes. Furthermore, it enables the implementation of custom Riemann solvers that can compute the respective fluctuations. The validation cases are implemented through the Python interface of ExaHyPE 2. It also enables the robustness of the simulation of non-linear hyperbolic systems, like the nonlinear shallow-water equations investigated in this work. Also, the arising discontinuities can be resolved numerically. It allows the user to specify numerical properties like the patch size or the size of the finite volume cells, which provides good scalability and variety.

There are also other alternatives that deal with the simulation of such flows. The first is the open-source *Clawpack* software, which also enables solving first-order partial differential equations. Recent developments and its general project structure can be found in [16]. Just like ExaHyPE 2, its grids are based on Cartesian meshes. Clawpack consists of a series of different projects. In the context of this work, especially *Geoclaw* is of interest, as it deals with the simulation of depth-averaged geophysical flows [16]. Clawpack also provides the user with a wide range of different Riemann solvers, which will be a focal point of this thesis. Clawpack provides both normal and transverse Riemann solvers, which allows for great accuracy in the respective simulations. As mentioned in [16], Clawpack is utilized for the simulation of the two-dimensional shallow water equations with topography. Another interesting feature is that its Riemann solvers employ a wave propagation algorithm invented by LeVeque. This is slightly different than algorithms utilizing numerical fluxes. Both the general finite volume formula and the wave propagation algorithm will be mentioned in this work.

Another relevant software project is the HySEA model described in [14]. It deals with granular and solid flow-induced tsunamis. For this purpose, it employs the coupling of different models for the respective different flow types. Its ability has been shown by simulating a variety of well-accepted benchmarks with it. Even though it employs the coupling of different models and utilizes a different set of governing equations, it deals with the simulation and validation of geophysical flows, which will also be investigated in this work.

This thesis will also deal with the implementation of approximate Riemann solvers for the two-dimensional shallow water equations. There are many different approaches and papers within this area of research, especially concerning the HLLC solver family. There, interesting inventions are described in [18], which incorporate special mechanisms dealing with source terms.

3 Theory

The basic model for fluid flows is the Navier-Stokes equations [2]. One can reduce their complexity by making certain assumptions, as in the case of the shallow water equations [19]. Thus, scientific simulation frameworks consult simplified models that are often based on the Navier-Stokes equations. The differences between such models are mainly different physical assumptions they make, e.g., whether they assume hydrostatic pressure or whether they incorporate dispersion effects. Such models are, for example, given by the Boussinesq [11] or the Korteweg-De Vries equations [15], which describe the dynamics of waves. In this work, the shallow water equations will be used to describe the evolution of shallow water flows with satisfactory accuracy in many cases.

3.1 Physical Models

3.1.1 The Shallow Water Equations

The theoretical basis of the following equations is the conservation laws of both mass and momentum. Based on them, the Navier-Stokes equations can be formulated for fluid dynamics. These equations can describe the evolution of the flow of various types of fluids. [2] If one further simplifies these equations, one can derive the so-called shallow water equations. Scientific literature, like [23] and [10] analyzses the shallow water model and its properties in detail. These equations can describe free-surface flows and are incorporating a series of simplifications. As the name suggests, the water flow is assumed to be shallow, so the horizontal dimension is the dominant factor, while the vertical dimension is relatively small. Based on this premise, various simplifications are embedded in this model. As such, the vertical acceleration of the fluid is omitted in the model, as one presumes that its influence is rather irrelevant for shallow water regions. The horizontal velocities are then integrated along the vertical z-dimension, leading to depth-averaged velocities u and v. Furthermore, on the free surface, the atmospheric pressure is set to be zero [23]. Additionally, one assumes that the pressure distribution is hydrostatic. The shallow water equations can be expressed in both conservative and physical variables. Here, the conservative variable formulation will be used, as they are also the variables employed in the numerical simulations. They can also be expressed in different forms, namely the differential and the integral forms. Equations

3.1-3.3 describe different versions of the 2-dimensional SWE in differential form.

$$\partial_t \begin{pmatrix} h \\ hu \\ hv \end{pmatrix} \quad +\partial_x \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix} \quad +\partial_y \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix} \quad = \begin{pmatrix} 0 \\ -gh\partial_x b \\ -gh\partial_y b \end{pmatrix} \tag{3.1}$$

$$\partial_t \begin{pmatrix} h \\ hu \\ hv \end{pmatrix} \quad +\partial_x \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix} \quad +\partial_y \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix} \quad = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.2}$$

$$\partial_t \mathbf{Q} \quad +\partial_x \mathbf{F}(\mathbf{Q}) \quad +\partial_y \mathbf{G}(\mathbf{Q}) \quad = \mathbf{S}(\mathbf{Q}) \tag{3.3}$$

Both Equation 3.1 and Equation 3.2 consist of three scalar partial differential equations each. The respective first equations, describing the time change of h ($\partial_t h$), comprise the mass conservation. The respective second and third PDEs, describing the time change of hu and hv respectively, describe the x- and y-momentum conservation. Equation 3.1 constitutes the differential SWE in inhomogeneous form. The inhomogeneous here describes that they incorporate a source term; here, the geometric source is caused by varying bathymetry. By incorporating a source term, the equation depicts a balance law. In contrast, Equation 3.2 poses the formulation of the homogeneous SWE, where the source term is set to zero. Thus, it is a conservation law. At last, equation (3.3) displays the vector form of equation (3.1) or (3.2), depending on whether the $S(Q)$ vector contains the geometric source or not. The syntax is the following: Q is the vector of conserved variables, $F(Q)$ and $G(Q)$ are the physical flux vectors in x- and y-direction, and $S(Q)$ is the source term. In the equations 3.1 and 3.2 $h(x, y, t)$ is the material height, which could be water or another material. $u(x, y, t)$ and $v(x, y, t)$ are the depth-averaged horizontal flow velocities in x- and y-direction. g is the gravitational constant set to be $9.81\frac{m}{s^2}$ on Earth. $b(x, y)$ describes the bottom bathymetry of the domain. Subsequently, the terms $\partial_x b$ and $\partial_y b$ are the slopes of the bathymetry in x- and y-direction. In this thesis, the bottom bathymetry is constant over time. Another parameter not included in the standard formula but will become relevant afterward is the surface elevation s. It is defined in Equation 3.4.

$$s(x, y, t) = b(x, y) + h(x, y, t) \tag{3.4}$$

A simplistic exemplary setup with the corresponding variable conventions is depicted in Figure 3.1. The equations mentioned above describe hyperbolic partial differential equations. The hyperbolicity can be assessed by examining the flux Jacobians $\frac{\partial \mathbf{F}(\mathbf{Q})}{\partial \mathbf{Q}}$ and $\frac{\partial \mathbf{G}(\mathbf{Q})}{\partial \mathbf{Q}}$, further information can be found in [23].

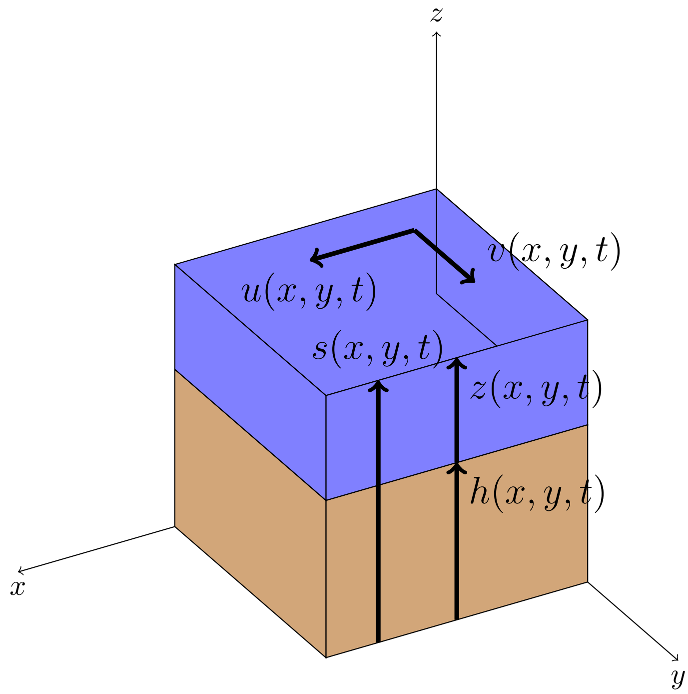


Figure 3.1: An exemplary setup of the shallow water model. (inspired by [4])

3.1.2 The Landslide Model

One can extend the shallow water equations with a friction term to model dry, granular, gravity-driven flows with friction. There are different friction terms and compositions of friction terms one can consider, but in this work, only the Coulomb friction term will be used. Equation (3.5) displays its general formula. The following formulae will replace the bathymetry variable with a topography variable z, as the model is used for landslides.

$$f = -\mu ghc\frac{u}{|v|} \tag{3.5}$$

$$\mu = tan(\phi) \tag{3.6}$$

$$c = \frac{1}{\sqrt{1 + (\partial_x z)^2 + (\partial_y z)^2}} = cos(\theta) \tag{3.7}$$

$$|v| = \sqrt{u^2 + v^2} \tag{3.8}$$

Equation (3.5) displays the friction f and thus introduces several new parameters. The first new parameter is c, the cosine of the local elevation slope, displayed in (3.7). The local elevation slope is represented by the slope angle θ. Another new parameter is ϕ, the basal friction angle, a material parameter. μ is then the tangens of ϕ (3.6). $|v|$ describes the magnitude of the two horizontal depth averaged velocities (3.8). Incorporating the Coulomb friction law into equation (3.1) results in equation (3.9), which represents a simplified model for granular material flows.

$$\partial_t \begin{pmatrix} h \\ hu \\ hv \end{pmatrix} + \partial_x \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix} + \partial_y \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix} = \begin{pmatrix} 0 \\ -gh\partial_x z - \mu ghc\frac{u}{|v|} \\ -gh\partial_y z - \mu ghc\frac{v}{|v|} \end{pmatrix} \tag{3.9}$$

3.2 Finite Volume Methods

The previous Section 3.1 introduced the two physical models this thesis evaluates. These will represent the basis for the numerical simulations of various scenarios described in Chapter 4. Before the numerical simulations can be executed, one needs both a numerical framework and a corresponding implementation. Therefore, this subsection introduces the numerical methods applied in the simulations. There are various approaches one can take in order to discretize the computational domain and time. In numerical simulations of the shallow water equations, there are three well-established discretization strategies, namely the finite differences methods, finite element methods, and the finite volume methods [3]. In order to choose a feasible method for the respective simulations, one has to carefully evaluate their respective advantages and disadvantages, especially concerning properties like the conservation of physical quantities, the accuracy of the computation, and the resulting computational costs.

This work will focus on finite volume methods, as they incorporate many desirable numerical properties. For example, they are able to work with discontinuous data, unlike the classical finite differences approach. Furthermore, employing them for solving hyperbolic systems of conservation laws is a reasonable decision, as they inherently conserve physical quantities like the accumulated mass over the computational domain, presuming one applies feasible boundary conditions so that the quantities do not escape. The aforementioned desirable properties of finite volume methods are due to their origin, as they are based on the integral form of conservation laws, which are in general defined in one space dimension as displayed in (3.10) [10].

$$\frac{d}{dt}\int_{x_1}^{x_2} q(x,t)dx \; = f(q(x_1,t)) - f(q(x_2,t)) \tag{3.10}$$

This states that the time change of the placeholder quantity q in the space interval $[x_1, x_2]$ depends on the net fluxes of the boundary points. Without these net boundary fluxes, the integral of the quantity q remains constant. The finite volume approach then embeds these conservation laws in its formula after discretizing the domain in time and space, resulting in so-called control volumes. Accordingly, these control volumes span over a space-time-domain. Equation (3.11) defines the two-dimensional case for a control volume centered around point (x_i, y_j).

$$CV_{i,j} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}] \times [t_n, t_{n+1}] \tag{3.11}$$

After applying the two-dimensional extension of the conservation formula in (3.10) on the control volumes defined by (3.11), and averaging over time and space, one can deduce the finite volume formula for two dimensions on a control volume CV_i, displayed in (3.12).

$$\mathbf{Q_{i,j}^{n+1}} = \mathbf{Q_{i,j}^{n}} - \frac{\Delta t}{\Delta x} * (\mathbf{F_{i+\frac{1}{2},j}} - \mathbf{F_{i-\frac{1}{2},j}}) - \frac{\Delta t}{\Delta x} * (\mathbf{G_{i,j+\frac{1}{2}}} - \mathbf{G_{i,j-\frac{1}{2}}}) + \Delta t \mathbf{S_{i,j}} \tag{3.12}$$

Equation (3.12) describes the computation of the new values of the vector of conserved variables $\mathbf{Q_{i,j}^{n+1}}$ after one time step, based on the old vector $\mathbf{Q_{i,j}^{n}}$, the numerical fluxes $\mathbf{F_{i+\frac{1}{2},j}}$ and $\mathbf{F_{i-\frac{1}{2},j}}$ in normal x-direction, the numerical fluxes in $\mathbf{G_{i,j+\frac{1}{2}}}$ and $\mathbf{G_{i,j-\frac{1}{2}}}$ in normal y-direction, and the numerical source $\mathbf{S_{i,j}}$. This work only considers first-order finite volume schemes, so the variable values of the conserved vectors $\mathbf{Q_{i,j}}$ are assumed constant per control volume. The unknown terms of (3.12) are the numerical fluxes at the interfaces and the numerical source, as the old conserved available vector is given at the time of evaluation and the values of Δt and Δx can be chosen by the developer, presuming one chooses them in a feasible manner. Computing the numerical fluxes and source exactly is not computationally feasible, so numerical methods are used to determine them approximately. One such method, the Godunov upwind method, utilizes a part of the solution of so-called Riemann problems at the control volume interfaces in the numerical flux computation for computing time steps via the general finite volume formula [23]. Figure 3.2 visualizes a control volume $CV_{i,j}$ and its relations with the neighbouring volumes via the numerical flux functions. The next section will investigate these approximate methods further.

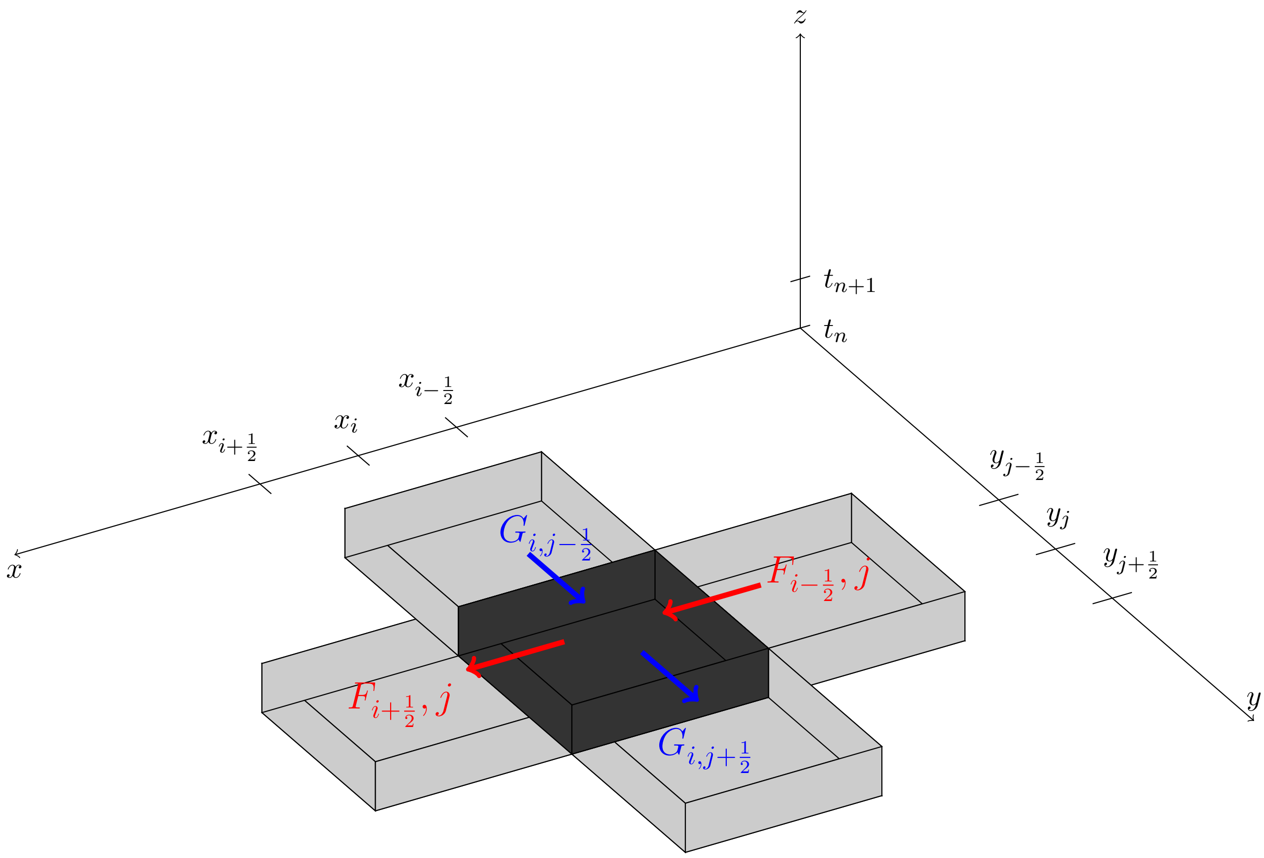
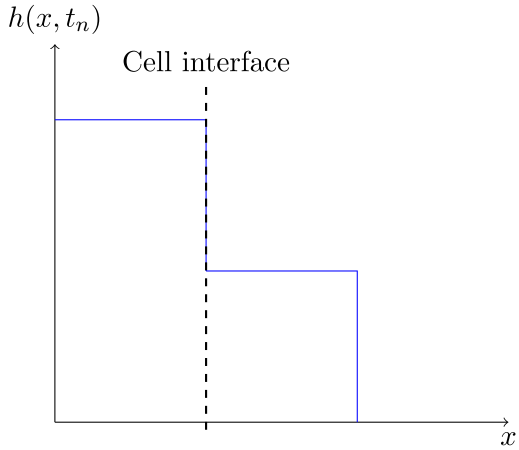


Figure 3.2: A classical Finite Volume in one space dimension

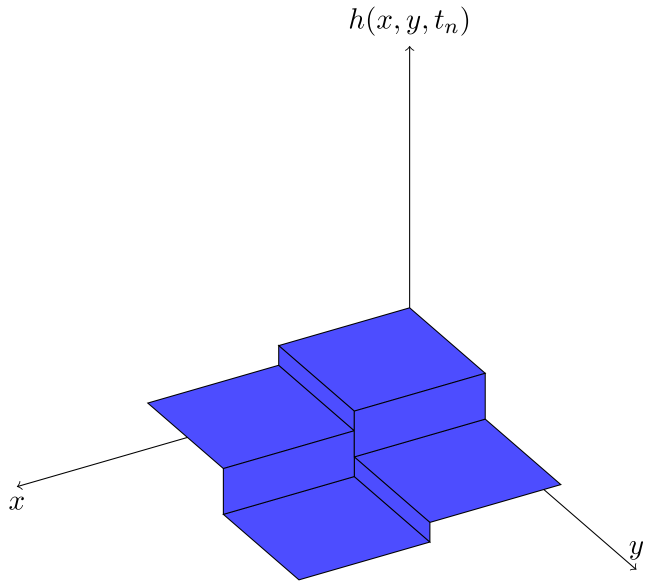
3.3 The Riemann Problem

This section will introduce the aforementioned Riemann problem and its formula and depict why it is a paramount method for solving the shallow water equations approximately.

3.3.1 Riemann Problem and Interface Fluxes



(a) Riemann problem for the water height $h(x, t_n)$ for an arbitrary time t_n for one-dimensional control volumes (inspired by [23]).



(b) Discontinuous data for the water height $h(x, y, t_n)$ for an arbitrary time t_n for two-dimensional control volumes. At the interfaces between these control volumes, local Riemann problems occur.

Figure 3.3: Exemplary setup of discontinuous data entailing local Riemann problems illustrated in one and two space dimensions.

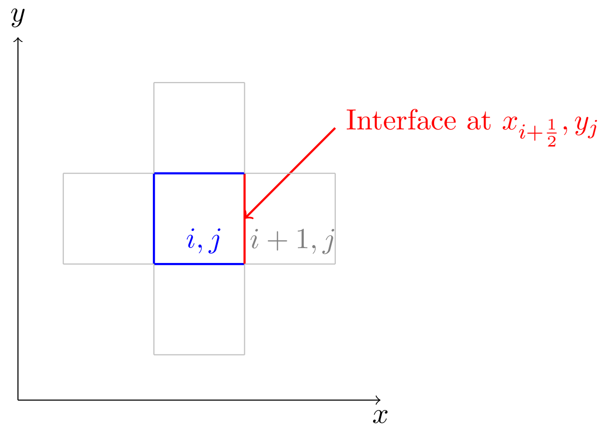
A Riemann problem is an initial value problem for partial differential equations. If one considers two volumes, a Riemann problem entails that the initial data is piecewise constant, with a discontinuity at the interface between the two volumes. Thus, the basic formula for the initial condition of a 1D Riemann problem with one discontinuity is shown in (3.13). In this formula, the discontinuity is assumed to be at $x = 0$. Combining this Riemann initial condition with a PDE like (3.1) or (3.2) then comprises the complete Riemann problem.

$$\mathbf{Q}(\mathbf{x}, \mathbf{0}) = \begin{cases} \mathbf{Q}^{\mathbf{Left}} & if\ x < 0 \\ \mathbf{Q}^{\mathbf{Right}} & if\ x > 0 \end{cases} \tag{3.13}$$

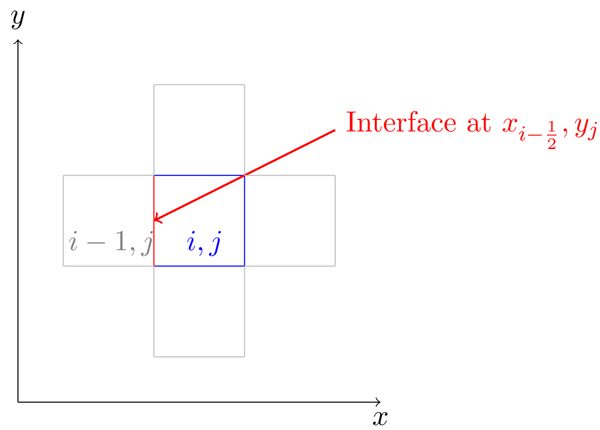
As (3.13) shows, there are two distinct vectors $\mathbf{Q}^{\mathbf{Left}}$ and $\mathbf{Q}^{\mathbf{Right}}$, with a discontinuity at $x = 0$, where the values of $\mathbf{Q}$ might change abruptly. Figure 3.3a illustrates such an initial condition of a one-dimensional Riemann problem for the water height $h(x, t_n)$. Figure 3.3b shows how piecewise initial data for $h(x, y, t_n)$ might look like in an arbitrary two-dimensional setup. The relevant volumes for the finite volume approach are the control volumes $CV_{i,j}$ as described in (3.11); the conserved variables are constant within these volumes. The Riemann problems then arise from the discontinuities of these conserved variable values at the control volume interfaces between neighbouring CVs. One control volume $CV_{i,j}$ in a two-dimensional domain, therefore, has four relevant interfaces, where discontinuities of the values of $\mathbf{Q}$ occur; at $x_{i-\frac{1}{2}}, y_j$ and $x_{i+\frac{1}{2}}, y_j$ for interfaces normal to the x-direction, and at $x_i, y_{j-\frac{1}{2}}$ as well as $x_i, y_{j+\frac{1}{2}}$ for interfaces normal to the y-direction. The local Riemann problems for one $CV_{i,j}$ are then given by the following formulas.

The initial conditions of Riemann problems occurring at interfaces normal to the x-axis are:

$$\mathbf{Q}(\mathbf{x}, \mathbf{y}, \mathbf{t_n}) = \begin{cases} \mathbf{Q_{i,j}} & x < x_{i+\frac{1}{2}} \\ \mathbf{Q_{i+1,j}} & x > x_{i+\frac{1}{2}} \end{cases}$$

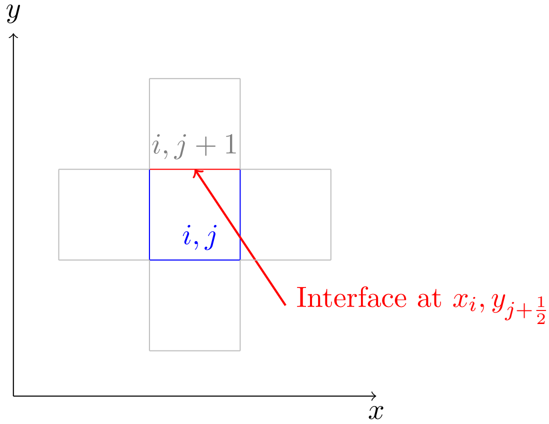


$$\mathbf{Q}(\mathbf{x}, \mathbf{y}, \mathbf{t_n}) = \begin{cases} \mathbf{Q_{i-1,j}} & x < x_{i-\frac{1}{2}} \\ \mathbf{Q_{i,j}} & x > x_{i-\frac{1}{2}} \end{cases}$$

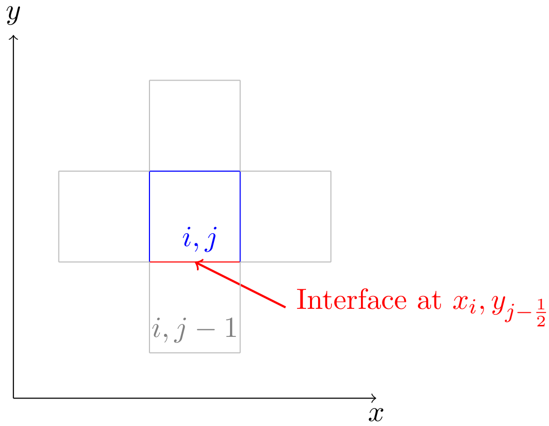


The initial conditions of Riemann problems occurring at interfaces normal to the y-axis are:

$$\mathbf{Q}(\mathbf{x},\mathbf{y},\mathbf{t_n}) = \begin{cases} \mathbf{Q_{i,j}} & y < y_{j+\frac{1}{2}} \\ \mathbf{Q_{i,j+1}} & y > y_{j+\frac{1}{2}} \end{cases}$$



$$\mathbf{Q}(\mathbf{x},\mathbf{y},\mathbf{t_n}) = \begin{cases} \mathbf{Q_{i,j-1}} & y < y_{j-\frac{1}{2}} \\ \mathbf{Q_{i,j}} & y > y_{j-\frac{1}{2}} \end{cases}$$



The relevant interfaces for each Riemann problem are marked in red. The solution to these problems can then be incorporated in the approximation of the numerical interface fluxes, e.g., the solution of the Riemann problem at interface $x_{i+\frac{1}{2}}, y_j$ is used for the computation of the numerical flux $F_{i+\frac{1}{2},j}$. Solving each of these local Riemann problems incorporates computing the resulting wave structure. The number of resulting waves and the corresponding wave types depend on the hyperbolic System. Subsection 3.3.2 will concisely introduce the three wave types relevant to the two-dimensional shallow water equations as well as its whole wave structure.

3.3.2 Wave Structure for the Shallow Water Equations

As mentioned before, the solution of a Riemann problem results in a specific wave structure, that is both dependent on the specific hyperbolic system and the respective initial values. These waves propagate at certain speeds and each have specific properties concerning their propagation and the change of values they entail. Before looking at the explicit wave structure, one should investigate the quasi-linear formulation of the conservation law, as it is used to deduce certain relevant speeds. (3.14) shows this specific form of the homegeneous shallow water equations (3.2), as it has been described by [23].

$$\partial_t \mathbf{Q} + \mathbf{A}(\mathbf{Q})\partial_x \mathbf{Q} + \mathbf{B}(\mathbf{Q})\partial_y \mathbf{Q} = \mathbf{0} \tag{3.14}$$

Here $\mathbf{A}(\mathbf{Q}) = \frac{\partial \mathbf{F}(\mathbf{Q})}{\partial \mathbf{Q}}$ and $\mathbf{B}(\mathbf{Q}) = \frac{\partial \mathbf{G}(\mathbf{Q})}{\partial \mathbf{Q}}$ are the aforementioned flux Jacobians, displayed in (3.15) and (3.16) respectively.

$$\mathbf{A}(\mathbf{Q}) = \begin{pmatrix} 0 & 1 & 0 \\ a^2 - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix} \quad (3.15)$$

The resulting eigenvalues of $\mathbf{A}(\mathbf{Q})$ are:

$$\lambda_1 = u - a, \lambda_2 = u, \lambda_3 = u + a$$

$$\mathbf{B}(\mathbf{Q}) = \begin{pmatrix} 0 & 0 & 1 \\ -uv & v & u \\ a^2 - v^2 & 0 & 2v \end{pmatrix} \quad (3.16)$$

The resulting eigenvalues of $\mathbf{B}(\mathbf{Q})$ are:

$$\lambda_1 = v - a, \lambda_2 = v, \lambda_3 = v + a$$

The three respective eigenvalues of the matrices $\lambda_i \; for \; i = 1, 2, 3$ are the so-called characteristic speeds of the system. In the case of the shallow water equations, they influence the wave speeds. With this knowledge, one can now also inspect the wave structure of the two-dimensional shallow water equations. There, each wave system, that is, the solution to the corresponding Riemann problem, consists of three waves, influenced respectively by the three characteristic speeds. Figure 3.4 shows how an exemplary wave system at the interface of two control volumes might look like in a so-called x-t coordinate system. In this figure, only one local Riemann problem at an interface normal on the x-axis is shown, and the y-coordinate is assumed to be at a fixed value. For the y-direction, the computation is analogous at the respective interfaces normal to the y-axis.

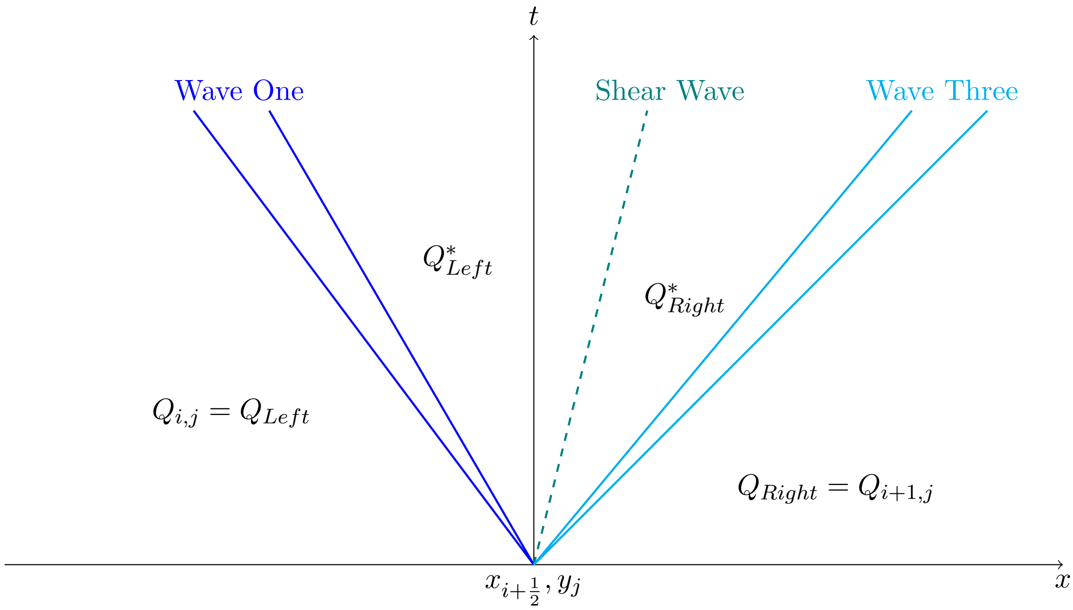


Figure 3.4: Wave system that poses the solution of a Riemann problem at an interface normal to the x-axis. At the left-hand side, one can see the control volume $CV_{i,j}$; at the right-hand side, $CV_{i+1,j}$. The y coordinate is fixed at y_j (inspired by [23]).

This wave system shows the solution to the Riemann problem. There are four different vectors of conserved variable values, which are separated through the three waves. The values of the different Q vectors that pose the solution of the initial value problem are shown in (3.17) and (3.18).

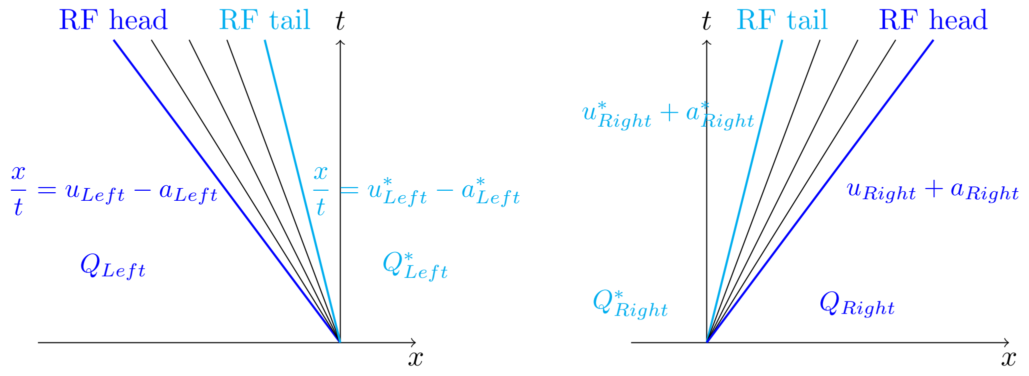
$$\text{Initial Condition values:} \quad Q_{Left} = \begin{pmatrix} h_{Left} \\ hu_{Left} \\ hv_{Left} \end{pmatrix}, \; Q_{Right} = \begin{pmatrix} h_{Right} \\ hu_{Right} \\ hv_{Right} \end{pmatrix} \tag{3.17}$$

$$h^* \text{ region values:} \quad Q^*_{Left} = \begin{pmatrix} h^* \\ hu^* \\ hv^*_{Left} \end{pmatrix}, \; Q^*_{Right} = \begin{pmatrix} h^* \\ hu^* \\ hv^*_{Right} \end{pmatrix} \tag{3.18}$$

On the one hand, the values of Q_{Left} and Q_{Right} from Equation 3.17 are given through the initial condition of the problem. They don't have to be computed. On the other hand, the values Q^*_{Left} and Q^*_{Right} from Equation 3.18 depend on solving the Riemann problem, thus are of unknown nature. The area on the x-t coordinate system in which $Q(\frac{x}{t})$ takes on a value of such an unknown vector is known as the *star region*, which is delimited by *Wave One* and *Wave Three* as illustrated in Figure 3.4. If one further investigates the scalar values inside the vectors Q^*_{Left} and Q^*_{Right}, one can note that both the water height as well as the normal velocity are constant in this star region but still unknown. The transverse velocity, v^*, changes between Q^*_{Left} and Q^*_{Right}. The change is caused by a so-called shear wave. Mathematical computation yields that $v^*_{Left} = v_{Left}$ and $v^*_{Right} = v_{Right}$, so these values can be easily inserted into the respective vectors. So there are remaining unknown scalar values, namely the physical variables h^* and u^*. These can be calculated through different mathematical techniques depending on the type of the respective waves, the initial values, and their relation if one were to solve the problem exactly. So, the wave types are an important part of the solution; there are three relevant types for the shallow water equations [23].

Rarefaction Waves

The first wave type that can occur is a so-called "Rarefaction." It is a continuous wave type. Figure 3.5 illustrates how such a wave appears on the x-t coordinate system. In the investigated Riemann problems of the shallow water equations, it can be the λ_1 wave ("Wave One"), then it would be a left rarefaction as shown in Figure 3.5a. In this case, the left rarefaction connects the states Q_{Left} and Q^*_{Left}. A rarefaction can also be the λ_3 wave ("Wave Three"), which would then be a right rarefaction that connects the corresponding right-hand states.



(a) Left rarefaction

(b) Right rarefaction

Figure 3.5: Illustration of left-going and right-going rarefaction waves. They connect two states with piecewise constant data respectively. Their propagation speed depends on the characteristic fields (inspired by [23]).

The rarefaction wave type spans a fan of characteristic curves delimited by two curves, the *rarefaction head* and the *rarefaction tail*. The speeds of these delimiting curves depend on the states left and right of the rarefaction, as well as on which characteristic field the rarefaction depends. How they are defined for these cases is also shown in Figure 3.5. The exact mathematical values for the values $Q(\frac{x}{t})$ inside the rarefaction fan and their derivation are further investigated in scientific literature like [23]. For the rarefaction, equations (3.19)-(3.21) display the most important mathematical relations.

$$v^*_{Left} = v_{Left} \qquad v \text{ is constant here in general} \tag{3.19}$$

$$u^*_{Left} + 2 * a^*_{Left} = u_{Left} + 2 * a_{Left} \qquad u + 2a \text{ is constant here in general} \tag{3.20}$$

$$a = \sqrt{g * h} \qquad \text{celerity } a \tag{3.21}$$

Shock Waves

The second wave type that can occur is a *shock wave*. Unlike a rarefaction, it is a discontinuous wave that causes an abrupt change of values at its front. It progresses with a shock wave speed S, and the explicit value of that speed is bounded through the "Rankine-Hugoniot" condition. The shock wave also doesn't alter the value of the velocity v, meaning that the value of v in the shock wave separated states is equal behind and in front of the shock. That is similar to the rarefaction wave. As for the rarefaction, in the shallow water wave system, the shock wave can be either the λ_1 (*Wave One*) or the λ_3 wave (*Wave Three*). Figure 3.6 displays examplary shock waves and their speeds. The parameters q_{Left} and q_{Right} are defined in (3.23).

$$q_{Left} = \sqrt{\frac{1}{2}\frac{(h_{Left} + h^*_{Left})h^*_{Left}}{h^2_{Left}}} \tag{3.22}$$

$$q_{Right} = \sqrt{\frac{1}{2}\frac{(h_{Right} + h^*_{Right})h^*_{Right}}{h^2_{Right}}} \tag{3.23}$$

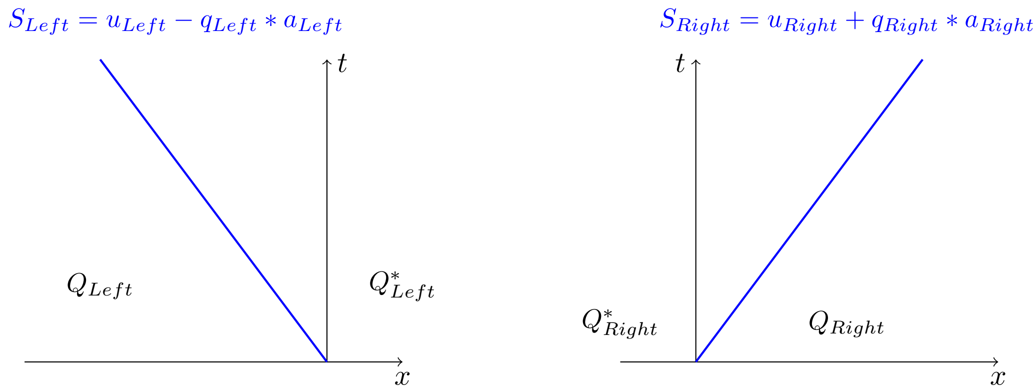


Figure 3.6: Shock waves connecting the two states it separates. The shock can be either left-going or right-going, depending on to which characteristic field it belongs. The respective shock speeds are shown as S_{Left} and S_{Right} (inspired by [23]).

Shear Waves

The last relevant wave type is "shear waves." Here, the middle wave, "Wave Two", is always a shear wave. Unlike the two aforementioned waves, it doesn't alter the water height or the normal velocity between the states it connects, but instead only the tangential velocity.

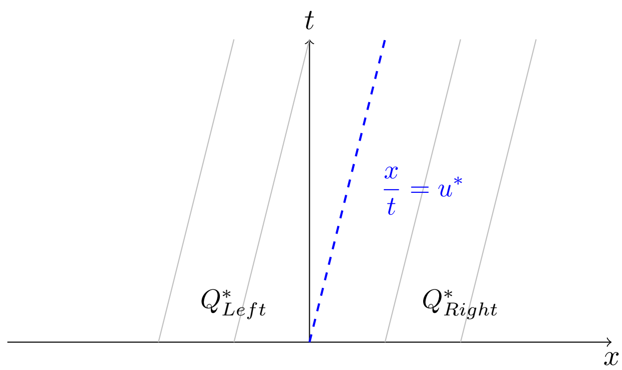


Figure 3.7: Shear Wave. (inspired by [23])

3.4 Wave System - Conclusion

This section on the wave structure and wave types assumed that the Riemann problem concerns control volumes separated through an interface normal to the x-axis. If one solves a Riemann problem lying on an interface normal to the y-axis, then the normal velocity is v, and u is the tangential velocity. The respective formulas and computations then need to take this change into account. As mentioned before, this mathematical background can be used in order to compute one step in the finite volume formula. The Godunov upwind method applies knowledge of the Riemann problem in order to compute the numerical fluxes. (3.24) shows how the Godunov upwind method computes the numerical flux at the right interface normal to the x-axis, as also shown in [22]. This $F(Q_{i+\frac{1}{2},j}(0))$ is the so-called "Godunov state", that is the exact solution of the Riemann problem at the respective interface for $\frac{x}{t} = 0$.

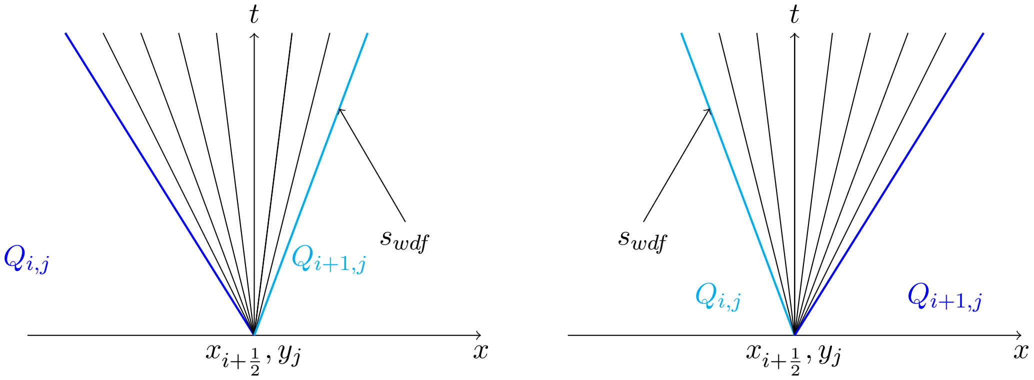
$$\mathbf{F_{i+\frac{1}{2},j}} = \mathbf{F}(\mathbf{Q_{i+\frac{1}{2},j}}(\mathbf{0})) \tag{3.24}$$

But for most cases, the necessary exact Riemann solution is not computable in a feasible manner. Therefore, other approximations of the numerical flux can be used. These are often still based on the premise of a Riemann problem.

3.4.1 Wetting and Drying Mechanisms

Before this work summarizes the applied numerical techniques, this subsection will briefly summarize the special problem of wetting and drying, especially the wetting process. A wetting, or dry-state Riemann problem, occurs, if one solves a Riemann problem for two neighboring control volumes, with one volume having a material height greater than zero, so $h > 0$, whilst the other cell is dry, so $h = 0$. If one applies a certain dry tolerance in the simulation, then a cell is dry if the material height is smaller than this tolerance. A dry tolerance is quite useful so that the simulation does not have to deal with values very close to zero, as that can easily lead to significant numerical instabilities. For shallow water simulations, this wetting mechanism needs to accurately model the process of inundation of dry cells. For landslides, this mechanism is even more important, as the propagating granular material constantly "wets" the dry region, that is the underlying topography. The resulting wave of a wetting Riemann problem is just one rarefaction, as [7] states. The respective speed of the resulting rarefaction head and tail depend on which state is dry and which state is wet at the initial condition. Figure 3.8 displays the two configurations of a dry-state Riemann problem incorporating the wave speed of the wet-dry front.

Both the processes of wetting cells as well as drying cells are quite difficult to accurately incorporate for approximate solvers. Nevertheless, accurately modeling them is quite substantial for both marine and biological research, as outlined in [19]. In this scientific work, another big challenge of this wetting and drying process is underlined, namely that the length scales of inundation processes may comprise horizontal and vertical length scales that don't satisfy the respective assumptions that are necessary for the shallow water equations to be applicable. From a numerical standpoint, these wetting and drying fronts often comprise very small water heights, which are always challenging to deal with in numerical methods.



(a) Single rarefaction wave structure, where the rarefaction originates from $Q_{i,j}$. The state $Q_{i+1,j}$ is the dry state with $h_{i+1,j} = 0$. The respective wave speed of the wet-dry front is $s_{wdf} = u_{i,j} + 2 * \sqrt{g * h_{i,j}}$.

(b) Single rarefaction wave structure, where the rarefaction originates from $Q_{i+1,j}$. Here, the state $Q_{i,j}$ is the dry state with $h_{i,j} = 0$. The respective wave speed of the wet-dry front is $s_{wdf} = u_{i+1,j} - 2 * \sqrt{g * h_{i+1,j}}$.

Figure 3.8: Illustration of the rarefactions posing the solutions of dry-state Riemann problems (inspired by [7]).

3.4.2 Source Term

In this section, this work will shortly summarize how the employed numerical methods and solvers will treat the respective source terms. As shown in Section 3.2, the finite volume formula comprises two different types of unknown data: the interface fluxes and the source terms. In the previous chapters, this work discussed how Riemann problems can be utilized to compute interface fluxes. Thus, the evaluation of the source term remains to be investigated. This source term has implications on the general structure of the wave system, as it can have an effect on the different wave speeds [18]. For the shallow water equations, including bathymetry, the geometric source term, from equation (3.1), has to be approximated. In this thesis, there will be two distinct approaches to include its computation. As will be shown in Section 3.5, there are approaches to include this geometric source in the computation of the intercell fluxes. How this exactly works depends on the respective approach, but in general, these approaches are useful for guaranteeing paramount properties like well-balancedness and positivity preservation. One possibility is to include this source term through an additional wave introduced in the presumed wave system [18, 7]. If one doesn't use a solver that utilizes this process, the respective numerical method consults a centered approach for this source term. This means that the geometric source equations, $-ghb_x$ and $-ghb_y$, will be evaluated separately from the computation of the intercell flux. The second relevant source term is the Coulomb friction term, cf. equation (3.5), which will be included in simulations with granular material. In this work, the finite volume method will always compute this source independently of the Riemann solver.

3.5 Approximate Riemann Solvers

This section will introduce multiple different Riemann solvers and their theoretical backgrounds concisely. Additionally, this work discusses these solvers' different important numerical properties, making them applicable to a wide range of applications. Furthermore, this section presents supplementary modifications that enable the solvers to deal with varying bathymetries and wetting and drying fronts.

3.5.1 Properties

There are many desirable properties for Riemann solvers in numerical simulations. This thesis will shortly discuss two of them: positivity preservation and well-balancedness. The following subsections will concisely summarize their meaning and importance in the context of the shallow water equations. More numerical properties that, for example, concern entropy or pressure gradients, can be found in scientific literature [19]. Ortleb et al. [19] discuss several of these numerical properties, especially with regard to wetting and drying procedures. Therefore, different wetting and drying mechanisms are presented, especially the thin-layer approximation. This thin-layer approximation is quite similar to the dry tolerance approach implemented in the following approximate solvers

Positivity Preservation

This attribute is self-explanatory. As the governing model describes flows of material above a bathymetry, the height of this material $h(x, y, t_n)$ should always be above greater or equal to zero: $h(x, y, t_n) >= 0$. If this does not hold, severe instabilities can occur, and the simulation is generally not physically feasible. Multiple parts of the shallow water equations are not mathematically defined for negative height values ($h(x, y, t_n) < 0$), like the computation of the wave celerity, which would then result in the calculation of a square root of a negative value [19]. It is especially critical in areas with relatively low material height and where a flux update decreases the height. While this property seems simple, not every solver fulfills it by default. The following solver descriptions will further explain their approach to positivity preservation.

Well-balancedness

This property classifies solvers depending on whether they can numerically preserve the steady states, meaning they do not severely deviate from a reached steady state in the time-stepping evolution. This state preservation is paramount for the simulation being physically feasible [1]. A steady state comprises the balance of different parts of the shallow water equations. Equations (3.25)-(3.27) show the simplest steady state, the so-called lake-at-rest steady state, described in [1].

$$h(x, y, t_n) + b(x, y, t_n) = s(x, y, t_n) = C \quad \text{with } C = constant \tag{3.25}$$

$$u = 0 \tag{3.26}$$

$$v = 0 \tag{3.27}$$

As one can observe in (3.26) and (3.27), both the horizontal velocities are zero in this case (at rest). One can investigate how such a lake-at-rest state influences the balance laws of the inhomogeneous shallow water equations (3.1). Equation (3.28) displays the balance law for the momentum in the x-direction.

$$\partial_t(hu) + \partial_x(hu^2 + \frac{1}{2}gh^2) + \partial_y(huv) = -gh\partial_x b \tag{3.28}$$

Now, with $u = 0$ and $v = 0$ given for the lake-at-rest case, or at least $u << \sqrt{gh}$, this x-momentum balance law simplifies to:

$$\partial_x(\frac{1}{2}gh^2) = -gh\partial_x b = -gh\partial_x(s - h) \tag{3.29}$$

The surface elevation $s(x, y, t_n)$ is constant along the space domain for the lake-at-rest case. Consequently, its partial derivative in the x-direction is zero: $\partial_x s(x, y, t_n) = 0$. This further transforms the balance law to [3]:

$$\partial_x(\frac{1}{2}gh^2) = gh\partial_x(h) \tag{3.30}$$

It is necessary that a numerical method satisfies the equalities (3.28) and (3.30) for well-balancedness. As mentioned in Audusse et al. [1], (3.28) is the relevant balance that induces the specific modified source term in the local hydrostatic reconstruction approach, that will be presented in Subsection 3.5.2. Besides this lake-at-rest scenario, numerous other steady states exist with $u > 0$ and/or $v > 0$. These are more difficult to numerically handle.

3.5.2 Local Hydrostatic Reconstruction

In the previous section, some important numerical properties were introduced. Most approximate solvers can not fulfill these important properties simultaneously by default, as Audusse et al. mention [1]. Therefore, one needs supplementary numerical methods to ensure adherence to these properties. Thus, this work summarizes such an approach, which will be referred to as Local Hydrostatic Reconstruction. This approach, based on hydrostatic reconstructions at control volume interfaces, helps the respective Riemann solver work with irregular bathymetries while also preserving positive material heights and maintaining the lake-at-rest steady state. It was developed by Audusse et al. in 2004 and introduced by the paper [1]. It is quite a powerful technique as one can couple it with any homogeneous Riemann solver [1]. More useful literature describing it is found in [19], as well as in [3]. In [3], it is shown how this approach can be incorporated into a 2D scheme, where it is coupled with an HLL solver. For this approach, the respective individual Riemann solver must recompute the heights based on a simple scheme. This scheme ensures that these reconstructed heights contain information regarding bathymetry. The following section will describe the basic height reconstruction mechanism for the interface normal to the x-axis, located at $x_{i+\frac{1}{2}}, y_j$. Thus, the control volume $CV_{i,j}$ is the left cell, whilst $CV_{i+1,j}$ is the right cell. In this section, variables associated with reconstruction will be labeled as Re or $HyRe$ respectively.

At first, one has to determine the maximum of the two bathymetry values at the interface. In this work, these values are always constant per control volume, like the conserved variables. Equation (3.31) shows this maximum bathymetry determination for the interface at $x_{i+\frac{1}{2}}, y_j$.

$$b_{i+\frac{1}{2},j} = b_{Max} = max(b_{i,j}, b_{i+1,j}) = max(b_{Left}, b_{Right}) \tag{3.31}$$

Based on this maximum bathymetry, equations (3.32) and (3.33) show how the left and right reconstructed heights can be computed, containing information about the bathymetry.

$$h_{LeftRe} = h_{i+\frac{1}{2}-,j} = max(0.0, h_{Left} - b_{Max} + b_{Left}) \tag{3.32}$$

$$h_{RightRe} = h_{i+\frac{1}{2}+,j} = max(0.0, h_{Right} - b_{Max} + b_{Right}) \tag{3.33}$$

The respective momenta also need to be adjusted appropriately. Equations (3.34) and (3.35) display this for the momenta of the left control volume.

$$hu_{LeftRe} = hu_{i+\frac{1}{2}-,j} = h_{LeftRe} * u_{Left} \tag{3.34}$$

$$hv_{LeftRe} = hv_{i+\frac{1}{2}-,j} = h_{LeftRe} * v_{Left} \tag{3.35}$$

The Riemann solvers can now execute their algorithm with these new height and momentum values. This process ensures the properties mentioned earlier. Figure 3.9 illustrates such an exemplary height reconstruction process. Finally, because of this height modification and the prospect of well-balancedness, one must modify the source term sufficiently. As this source term computation for one control volume needs information about the neighbouring control volumes, relocating it into the Riemann solver is the appropriate method. Equations (3.36)-(3.39) constitute this extended numerical flux computation for the respective interfaces [1, 3].

$$\mathbf{F_{i+\frac{1}{2},j,HyRe}} = \mathbf{F_{i+\frac{1}{2},j}} + \frac{1}{2}g \begin{pmatrix} 0 \\ h_{i,j}^2 - h_{i+\frac{1}{2}-,j}^2 \\ 0 \end{pmatrix} \tag{3.36}$$

$$\mathbf{F_{i-\frac{1}{2},j,HyRe}} = \mathbf{F_{i-\frac{1}{2},j}} + \frac{1}{2}g \begin{pmatrix} 0 \\ h_{i,j}^2 - h_{i-\frac{1}{2}+,j}^2 \\ 0 \end{pmatrix} \tag{3.37}$$

$$\mathbf{F_{i,j+\frac{1}{2},HyRe}} = \mathbf{F_{i,j+\frac{1}{2}}} + \frac{1}{2}g \begin{pmatrix} 0 \\ 0 \\ h_{i,j}^2 - h_{i,j-\frac{1}{2}+}^2 \end{pmatrix} \tag{3.38}$$

$$\mathbf{F_{i,j+\frac{1}{2},HyRe}} = \mathbf{F_{i,j+\frac{1}{2}}} + \frac{1}{2}g \begin{pmatrix} 0 \\ 0 \\ h_{i,j}^2 - h_{i,j+\frac{1}{2}-}^2 \end{pmatrix} \tag{3.39}$$

This source term inclusion is preferred over a usual centered source term approach decoupled from the numerical fluxes. Equation (3.40) displays such a centered approach, which has been proven to not always satisfy the well-balancedness condition [3].

$$-gh\partial_x b = -gh_{i,j} * \frac{b_{i+1,j} - b_{i-1,j}}{2 * \Delta x} \tag{3.40}$$

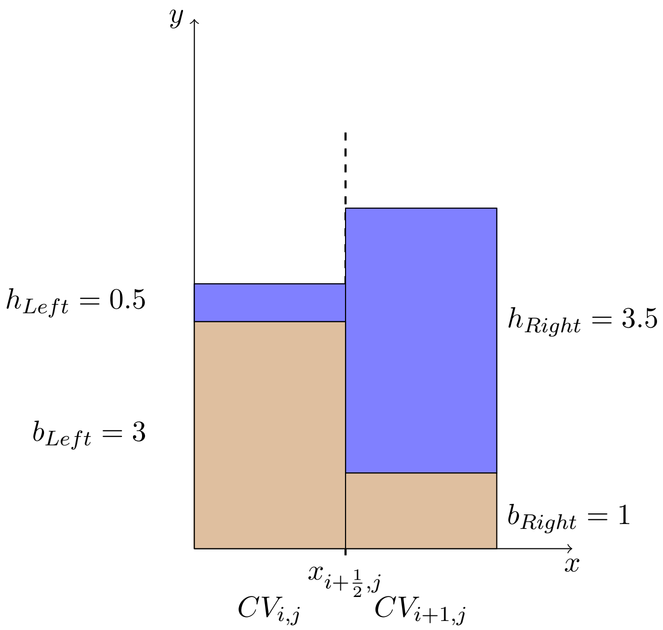


Figure 3.9: Examplary scenario for the height reconstruction at a cell interface (inspired by [3]).

This exemplary set up leads to the following reconstructed heights:

$$b_{Max} = max(b_{Left}, b_{Right}) = 3$$
$$h_{LeftHyRe} = 0.5 + 3 - 3 = 0.5$$
$$h_{RightHyRe} = 3.5 + 1 - 3 = 1.5$$

3.5.3 Kurganov Desingularization

This is a small numerical modification that can help with computing the horizontal velocities for very small values for $h(x, y, t_n)$. These velocities, u and v, necessitate the computations of $\frac{hu}{h}$ and $\frac{hv}{h}$, that can result in singularities for very small water heights [19]. An interesting approach for this problem is described in [9]. The formula for the height desingularization is displayed in Equation 3.41.

$$u = \frac{\sqrt{2}h(hu)}{\sqrt{h^4 + max(h^4, \epsilon)}} \tag{3.41}$$

3.5.4 The Rusanov Solver

The basic Rusanov solver is the least complex solver used in this work. The solver is described in [23]. This solver presumes a one-wave model. This is, of course, not sufficient for the two-dimensional shallow water equations, a three-wave model. Consequently, its approximate Riemann solutions are not accurate. Still, its simple structure enables its very robust nature, as it runs all validation cases without suffering from strong instabilities. Thus, it doen't need as many stability fixes as the more complex solvers. The solvers with a more complicated structure often need many modifications to also work with complex data. Thus, the robust Rusanov solver is still a popular numerical method.

3.5.5 The HLL Solver Family

As stated in the previous section, a Rusanov solver is not well-suited for numerical simulations based on the two-dimensional shallow water equations. Therefore, using solvers that incorporate more than just one wave seems reasonable. This section will shortly introduce the more sophisticated HLL and HLLC solvers, with more attention paid to the better working HLLC approach.

HLL Solver

This approximate Riemann solver comprises a two-wave model. Background on its derivation through control volumes and integral relations can be found in [22]. This solver can then directly approximate the interface fluxes between two control volumes. It necessitates pre-determined speed estimations s_{Left} and s_{Right} for the fastest waves of the solution to the Riemann problem. Generally, those estimates are better chosen to be greater or equal to the actual wave speeds; otherwise, instabilities can occur [5]. Still, this solver takes only two waves into account, which is insufficient for the two-dimensional shallow water equations. This will lead to inaccurate simulations, especially coupled with a first-order scheme. Thus, the next section will thoroughly discuss the better HLLC solver and different wave speed estimations.

HLLC Solver

This solver is related to the basic HLL solver. It also needs approximate wave speed estimations for its algorithm to work. The main difference is that the HLLC solver constitutes a three-wave model [22]. Therefore, the solver can approximate all the waves of the two-dimensional shallow-water equations 3.4, including the shear wave. Therefore, because of its relatively simple structure and fitting model, it has been implemented in ExaHyPE 2. This section will summarize the explicit formulas and assumptions this implementation utilizes. The ExaHyPE 2 implementation considers two different mechanisms to estimate the outer wave speeds s_{Right} and s_{Left}. These have distinct advantages and disadvantages, especially concerning robustness, complexity, and accuracy.

Einfeldt Speeds

The well-known Einfeldt speeds give the first implemented wave-speed estimation. Their computation encompasses just the characteristic and Roe speeds, followed by min and max operations. These wave speeds lead to depth-positive flux updates and robust updating steps.

$$\hat{h} = \frac{h_{Right} + h_{Left}}{2} \qquad \text{Roe averaged height}$$

$$\hat{u} = \frac{\sqrt{h_{Right}} * u_{Right} + \sqrt{h_{Left}} * u_{Left}}{\sqrt{h_{Right}} + \sqrt{h_{Left}}} \qquad \text{Roe averaged horizontal velocity}$$

$$\hat{c} = \sqrt{g * \hat{h}} \qquad \text{Roe celerity}$$

$$\lambda^{-}(\mathbf{Q_{Left}}) = u_{Left} - \sqrt{g * h_{Left}} \qquad \text{left characteristic speed}$$

$$\lambda^{+}(\mathbf{Q_{Right}}) = u_{Right} + \sqrt{g * h_{Right}} \qquad \text{right characteristic speed}$$

$$s_{Left} = min(\lambda_{Left}, \hat{u} - \hat{c}) \qquad \text{first Einfeldt speed}$$

$$s_{Right} = max(\lambda_{Right}, \hat{u} + \hat{c}) \qquad \text{second Einfeldt speed}$$

Middle height Based Speeds

The second wave speed estimates were proposed in [22] and [23]. They also consider the presumed wave family of each characteristic field. In this case, for the wave speed estimation process, one needs an approximation for the constant height of the middle state in the region spanned by the outer wave speeds. In the ExaHyPE 2 implementation, two such estimates have been considered. The first assumes that both waves are rarefactions. Equations (3.42) and (3.43) display this middle height estimation, which has been recommended in [22]. The estimated celerity for two arising rarefactions is:

$$a_{TR} = 0.5 * (a_{Left} + a_{Right}) - 0.25 * (u_{Right} - u_{Left}) \tag{3.42}$$

The subsequent estimated height is:

$$h^{*}_{estimate} = \frac{a_{TR}^{2}}{g} \tag{3.43}$$

This rarefaction-based wave speed estimation process is inaccurate for other wave families, especially shocks. During simulations executed for this work, this estimation process has been unstable for certain shock configurations when utilized in ExaHyPE 2. The aforementioned second middle height estimation mechanism is based on the deduced actual Riemann structure of the problem. Therefore, it can be more stable and accurate. Details can be found in [7].

One can compute the wave speed estimates once one determines the approximate middle height. First, one computes the q_{Left} and q_{Right} parameters, which pose a case distinction based on the presumed wave structure depending on the previously investigated middle state height estimation. Equation (3.44) illustrates this for $K = Left/Right$. Note the similarity to the true shock and rarefaction speeds in Subsection 3.3.2.

$$q_K = \begin{cases} \sqrt{0.5 \frac{(h^*_{estimate} + h_K) * h^*_{estimate}}{h_K^2}} & if\ h^*_{estimate} > h_K \\ 1 & if\ h^*_{estimate} \leq h_K \end{cases} \tag{3.44}$$

Afterwards, equations (3.45) and (3.46) show the calculation of s_{Left} and s_{Right}.

$$s_{Left} = u_{Left} - q_{Left} * a_{Left} \tag{3.45}$$
$$s_{Right} = u_{Right} + q_{Right} * a_{Right} \tag{3.46}$$

Now, we have two different wave speed estimation procedures. For both of them, the following steps are consistent with HLLC-type solvers. As a first small addition, both implementations consider the case of a dry-state Riemann problem explicitly by incorporating the wave speed estimates $s_{Left} = u_{Right} - 2a_{Right}$, if the left cell is dry, and $s_{Right} = u_{Left} + 2a_{Left}$, if the right cell is dry, as they align with the true front speed of the single rarefaction fan for these scenarios [23]. As mentioned before, HLLC comprises a three-wave model. Therefore, one needs a third wave speed, which approximates the shear wave, as well as the approximated left and right state in the star region. Consistent with the scientific literature, the middle state-associated parameters will be marked with a star. Equations (3.47) display this shear wave speed approximation, equations (3.48) and (3.49) the approximate star region states of the conserved variables. Note that the star regions of the respective sides vanish if the corresponding initial water height is zero, so $\mathbf{Q^*_{Left}}$ vanishes if $h_{Left} = 0$ [5]. For this concrete scenario, $s_{Left} = s^*$ holds when using (3.47), therefore $\mathbf{Q^*_{Left}}$ does not have to be considered in case distinction (3.50). The analogous statement holds for a dry cell right of the interface.

$$s^* = \frac{s_{Left} * h_{Right} * (u_{Right} - s_{Right}) - s_{Right} * h_{Left} * (u_{Left} - s_{Left})}{h_{Right} * (u_{Right} - s_{Right}) - h_{Left} * (u_{Left} - s_{Left})} \tag{3.47}$$

$$\mathbf{Q^*_{Left}} = h_{Left} * (\frac{s_{Left} - u_{Left}}{s_{Left} - s^*}) * \begin{pmatrix} 1 \\ s^* \\ v_{Left} \end{pmatrix} \tag{3.48}$$

$$\mathbf{Q^*_{Right}} = h_{Right} * (\frac{s_{Right} - u_{Right}}{s_{Right} - s^*}) * \begin{pmatrix} 1 \\ S^* \\ v_{Right} \end{pmatrix} \tag{3.49}$$

Finally, one needs the intercell flux, which will be evaluated for every Riemann problem by the case distinction in (3.50).

$$\mathbf{F^{HLLC}_{i+\frac{1}{2},j}} = \begin{cases} \mathbf{F_{Left}} & if\ 0 \leq s_{Left} \\ \mathbf{F_{Left}} + s_{Left} * (\mathbf{Q^*_{Left}} - \mathbf{Q_{Left}}) & if\ s_{Left} \leq 0 \leq s^* \\ \mathbf{F_{Right}} + s_{Right} * (\mathbf{Q^*_{Right}} - \mathbf{Q_{Right}}) & if\ s^* \leq 0 \leq s^{Right} \\ \mathbf{F_{Right}} & if\ 0 \geq s_{right} \end{cases} \tag{3.50}$$

Until now, this solver has not incorporated bathymetry effects; therefore, it is feasible only for homogeneous shallow water simulations. To also account for varying bathymetry, this solver will be coupled with the local hydrostatic reconstruction described in Subsection 3.5.2. There are also very interesting approaches that incorporate source terms through the so-called HLLS and HLLCS solvers, discussed in [18]. These incorporate the source term via a steady state wave, similar to the augmented Riemann solver discussed in Subsection 3.5.6. The HLLS and HLLCs still have the advantage, that they can also incorporate friction terms, which makes them very suitable for granular material flows. The investigation of the feasibility of their implementation in ExaHyPE 2 remains future work.

3.5.6 The Augmented Riemann Solver

This is the most complex solver that this work will evaluate. David L. George has developed it, describing it in depth in [7], as well as [6]. For this work, this solver has been implemented in ExaHyPE 2. The publicly available Clawpack implementation[1] is the basis of this ExaHyPE 2 implementation. Still, modifications were incorporated into the implementation of this work to make it compatible with the individual Riemann solver interface in ExaHyPE 2. Furthermore, some details have been implemented with close regard to the original paper descriptions. This section will summarize the most important features of this solver. One can find further details in the respective scientific literature. This work will call this solver the augmented Riemann solver, as an augmented state decomposition constitutes its basic structure.

As before, the basic scenario is two neighbouring control volumes with piecewise constant data on opposite sites of an interface. This section considers an interface located at $x_{i+\frac{1}{2}}, y_j$. Thus, $CV_{i,j}$ is the left, $CV_{i+1,j}$, the right control volume. This setup leads to the Riemann problem, and an approximate solution is to be found. For this purpose, the augmented Riemann solver takes a different approach than the aforementioned Rusanov

[1] `https://github.com/clawpack/riemann/blob/master/src/rpn2_sw_aug.f90`

and HLLC solvers. Instead of directly approximating the interface flux, an approximate wave decomposition poses the basis of its solution approach. This means that the solver decomposes an augmented variable vector into several waves (discontinuities), making the augmented Riemann solver part of the simple solver category. Equation (3.52) displays the explicit wave decomposition for the two-dimensional shallow water equations for a Riemann problem in the x-direction, where $\varphi(Q_{i,j})$ represents the momentum flux in the x-direction (3.53). This decomposition and the calculation of the respective wave speeds pose the whole procedure to approximate the Riemann solution. This approach requires a wave propagation algorithm, displayed in (3.51). This algorithm is quite similar to the one that works with intercell fluxes, displayed in (3.12). The only difference is that in (3.51), one has fluctuations $A^+\Delta Q$ and $A^-\Delta Q$ at the interfaces. Therefore, the signs are different.

$$
\begin{aligned}
Q_{i,j}^{n+1} = Q_{i,j}^{n} &- \frac{\Delta t}{\Delta x}(A^+\Delta Q_{i-\frac{1}{2},j} + A^-\Delta Q_{i+\frac{1}{2},j}) \\
&- \frac{\Delta t}{\Delta y}(B^+\Delta Q_{i,j-\frac{1}{2}} + B^-\Delta Q_{i,j+\frac{1}{2}}) \\
&- \text{correction terms}
\end{aligned}
\tag{3.51}
$$

Computing these fluctuations is now the task of the augmented Riemann solver. The waves constituting the fluctuations are similar in structure to flux waves. Equation (3.54) shows how the solver calculates them based on the approximated eigenvectors $w^p_{i+\frac{1}{2},j}$ of the flux Jacobian and the respective coefficients $\alpha^p_{i+\frac{1}{2},j}$. Depending on the sign of the wave speeds $s^p_{i+\frac{1}{2},j}$, the flux wave will be added to the leftward ($A^-\Delta Q_{i+\frac{1}{2},j}$) or rightward fluctuation ($A^+\Delta Q_{i+\frac{1}{2},j}$) .

$$
\begin{pmatrix} h_{Right} - h_{Left} \\ hu_{Right} - hu_{Left} \\ \varphi(Q_{Right}) - \varphi(Q_{Left}) \\ b_{Right} - b_{Left} \end{pmatrix} = \sum_{p=0}^{3} \alpha^p_{i+\frac{1}{2},j} w^p_{i+\frac{1}{2},j}
\tag{3.52}
$$

$$
\varphi(Q_{i,j}) = hu^2 + \frac{1}{2}gh^2
\tag{3.53}
$$

$$
Z^p_{i+\frac{1}{2},j} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \alpha^p_{i+\frac{1}{2},j} w^p_{i+\frac{1}{2},j}
\tag{3.54}
$$

The next sections will shortly introduce the calculation for the four waves.

The first and third Wave

The first and third waves will be determined using the Einfeldt speeds, incorporating an additional wave speed compared to the standard form. This additional considered wave speed helps the solver to accurately deal with edge cases like dry-state Riemann problems or transonic rarefactions. This new speed is represented by the speeds shown in (3.55) and (3.56). In the ExaHyPE 2 and Clawpack implementations, this speed is substituted with a computed middle state speed of the true Riemann structure, which helps with robustness and accuracy. Still, the concept remains the same.

$$\lambda^{-*}_{i+\frac{1}{2},j}(h^*_{i+\frac{1}{2},j}) = u_{Left} + 2\sqrt{gh_{Left}} - 3\sqrt{gh^*_{i+\frac{1}{2},j}} \tag{3.55}$$

$$\lambda^{+*}_{i+\frac{1}{2},j}(h^*_{i+\frac{1}{2},j}) = u_{Right} - 2\sqrt{gh_{Right}} + 3\sqrt{gh^*_{i+\frac{1}{2},j}} \tag{3.56}$$

Equations (3.57) and (3.58) depict how the augmented Riemann solver then incorporates these additional speeds in the Einfeldt speed computation. The variables $\hat{u}$, $\hat{c}$, $\lambda^-(\mathbf{Q_{Left}})$ and $\lambda^+(\mathbf{Q_{Right}})$ are defined as in subsubsection 3.5.5.

$$\check{s}^-_{i+\frac{1}{2}} = min(\lambda^-(\mathbf{Q_{Left}}), \hat{u} - \hat{c}, \lambda^{+*}_{i+\frac{1}{2},j}(h^*_{i+\frac{1}{2},j})) \tag{3.57}$$

$$\check{s}^+_{i+\frac{1}{2}} = min(\lambda^+(\mathbf{Q_{Right}}), \hat{u} + \hat{c}, \lambda^{-*}_{i+\frac{1}{2},j}(h^*_{i+\frac{1}{2},j})) \tag{3.58}$$

The Corrector Wave

This is a surplus wave that the augmented Riemann solver uses to improve the accuracy of rarefaction representations as well as prevent some physically wrong Riemann solutions. There are two forms described in the [7]. The first form is quite simple to ensure linear independence of the eigenvectors (3.59), whereas the second form is quite useful to improve the accuracy of rarefaction representations (3.60) and (3.61). The solver's choice of the two versions should depend on some conditions that the developer can choose. These conditions should concern an assumption whether the concrete Riemann solution will contain such a large rarefaction that the more complex corrector wave is necessary.

$$w^2_{i+\frac{1}{2},j} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad s^2_{i+\frac{1}{2},j} = \frac{1}{2}(\check{s}^-_{i+\frac{1}{2}} + \check{s}^+_{i+\frac{1}{2}}), \quad \text{simple corrector wave} \tag{3.59}$$

Corrector wave with a large rarefaction in the left characteristic field:

$$w^2_{i+\frac{1}{2},j} = \begin{pmatrix} 1 \\ \lambda^{-*}_{i+\frac{1}{2},j}(h^*_{i+\frac{1}{2},j}) \\ (\lambda^{-*}_{i+\frac{1}{2},j}(h^*_{i+\frac{1}{2},j}))^2 \\ 0 \end{pmatrix}, \quad s^2_{i+\frac{1}{2},j} = \lambda^{-*}_{i+\frac{1}{2},j}(h^*_{i+\frac{1}{2},j}) \tag{3.60}$$

Corrector wave with a large rarefaction in the right characteristic field:

$$w^2_{i+\frac{1}{2},j} = \begin{pmatrix} 1 \\ \lambda^{+*}_{i+\frac{1}{2},j}(h^*_{i+\frac{1}{2},j}) \\ (\lambda^{+*}_{i+\frac{1}{2},j}(h^*_{i+\frac{1}{2},j}))^2 \\ 0 \end{pmatrix}, \quad s^2_{i+\frac{1}{2},j} = \lambda^{+*}_{i+\frac{1}{2},j}(h^*_{i+\frac{1}{2},j}) \tag{3.61}$$

The Steady State Wave

This wave is responsible for integrating the geometric source into the Riemann solver, as was also done through the local hydrostatic reconstruction approach in the previous sections. Through this source term incorporation, well-balancedness of the scheme can be assured. This wave then represents a stationary discontinuity at the interface of the Riemann problem ($s^0_{i+\frac{1}{2},j} = 0$), representing the source term induced by the discontinuous bathymetry at this interface [7]. (3.62) shows the respective vector and wave speed of this wave. The wave speed is always set to zero, whereas the coefficient of the decomposition $\alpha^0_{i+\frac{1}{2},j}$ is always the difference of the discontinuous bathymetry. Therefore the steady state wave is zero for continuous bathymetry, ($b_{Left} = b_{Right}$). The functions $\bar{h}(Q_{Left}, Q_{Right})$, $\tilde{h}(Q_{Left}, Q_{Right})$ and $\tilde{h}(Q_{Left}, Q_{Right})$ are all certain averages of the corresponding quantities, their definition and choice is described in [7]. The ExaHyPE 2 and Clawpack implementation impose several bounds and some small modifications for better numerical stability, but this steady state wave poses the basis of their respective implementation.

$$w^0_{i+\frac{1}{2},j} = \begin{pmatrix} \dfrac{g\bar{h}(Q_{Left}, Q_{Right})}{\overline{\lambda^+\lambda^-}(Q_{Left}, Q_{Right})} \\ 0 \\ -g\tilde{h}(Q_{Left}, Q_{Right}) \\ 1 \end{pmatrix}, \quad s^0_{i+\frac{1}{2},j} = 0, \quad \alpha^0_{i+\frac{1}{2},j} = (b_{Right} - b_{Left}) \tag{3.62}$$

4 Validation Cases

This chapter will evaluate several implemented scenarios in order to test the Riemann solvers introduced above. The following sections will investigate four different classes of scenarios. In each section, the utilized Riemann solvers will be specified and compared. Three different categories of solvers will be utilized. At first, the Rusanov solver. The next solver used for every validation case is the augmented Riemann solver, introduced in Subsection 3.5.6. The last class of tested solvers is represented by the HLLC solvers with different wave speed estimates, as investigated in Subsection 3.5.5.

4.1 Thacker's Tests

The first test belongs to the category of *Thacker's tests*. William Carlisle Thacker introduced them in [21]. These tests are for the two-dimensional shallow-water equations with a geometric source term. The topographies of the respective scenarios are paraboloids, in which the water typically moves periodically. Thus, the effects of the varying topography on the momentum are incorporated. Whilst the water moves within this paraboloid, it always moves in and out of dry-state areas. Thus, these scenarios pay attention to varying topography and dry-state occurrences and are very suitable for validating the previously introduced numerical methods [4]. Furthermore, Thacker's tests are analytically solvable; thus, one can compare the numerical results with this analytical solution.

For this purpose, a Thacker's test and the corresponding analytical solution were implemented in ExaHyPE 2. This implementation is mainly based on [4], while [21] first introduced it as *oscillations for which the surface remains planar*. For the comparison with the analytical solution, the implementation uses the ExaHyPE 2 built-in error measurement feature. This feature calculates different errors based on the analytical solution over the whole domain for every unknown. Equations (4.1)-(4.4) displays how the respective errors are calculated for finite volume methods. There, e represents the error, while CV represents the volume of the respective cell. So, for this scenario, the errors concern the differences between the conserved variables h, hu, and hv, calculated by the numerical method on the one hand and the actual analytical value on the other hand. The error measurement tool of ExaHyPE 2 takes the sum of these differences for every conserved variable for each cell.

$$e = |numericalSolution - analyticalSolution| \tag{4.1}$$
$$L1_{error} = CV * e \tag{4.2}$$
$$L2_{error} = CV * e^2 \tag{4.3}$$
$$L\infty_{error} = max(L\infty_{error}, e) \tag{4.4}$$

Figure 4.1 displays the initial condition of this scenario. In this figure, the paraboloid, representing the basin [21], is marked brown. The logarithmic scaling is then applied to the

min-depth	patch-size	end-time	width	offset	amr-levels	dry-tol
6	32	10.0	[4.0, 4.0]	[-2.0, -2.0]	0	1e-4

Table 4.1: Simulation parameters for the Thacker's test.

material height. Equations (4.5)-(4.9) display the corresponding mathematical relations, taken from [4]. These formulas represent the initial condition when inserting $t = 0$ and subsequently the analytical solution when $t > 0$. Equations (4.10)-(4.14) show the selected parameters, similar to the case described in [4], where more information about them can be found. The main difference to the scenario in [4] is that the ExaHyPE 2 implementation is using the spatial domain $[-2, -2] \times [2, 2]$. It should be mentioned that by this analytical definition, the height will be set to below zero in some areas of the domain, while here, we set these height values to zero for this simulation. Thus, they are dry-state. The concrete relevant simulation parameters are displayed in Table 4.1.

$$r(x, y) = \sqrt{x^2 + y^2} \tag{4.5}$$

$$b(r) = -h_{Zero} * (1 - \frac{r(x, y)^2}{a^2}) \tag{4.6}$$

$$h(x, y, t) = \frac{\eta * h_{Zero}}{a^2} * (2.0 * x * cos(\omega t) + 2.0 * y * sin(\omega t) - \eta) - b(x, y) \tag{4.7}$$

$$hu(x, y, t) = -\eta\omega sin(\omega t) * h(x, y, t) \tag{4.8}$$

$$hv(x, y, t) = \eta\omega cos(\omega t) * h(x, y, t) \tag{4.9}$$

$$a = 1 \tag{4.10}$$

$$h_{Zero} = 0.1 \tag{4.11}$$

$$\eta = 0.5 \tag{4.12}$$

$$\omega = \frac{\sqrt{2gh_{Zero}}}{a} \tag{4.13}$$

$$T = 3\frac{2\pi}{\omega} \tag{4.14}$$

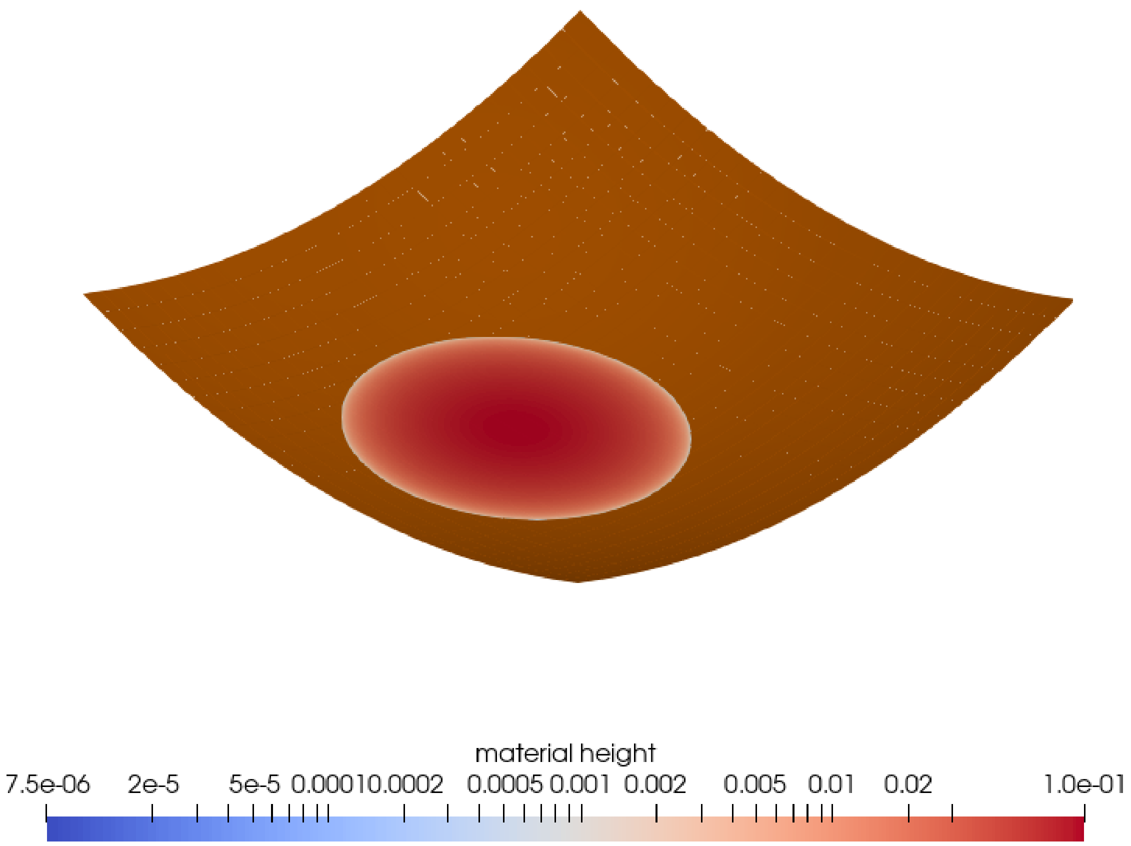
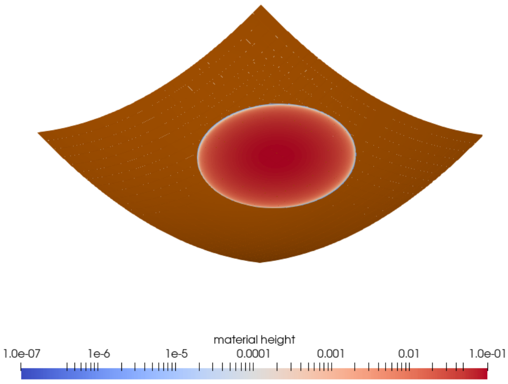
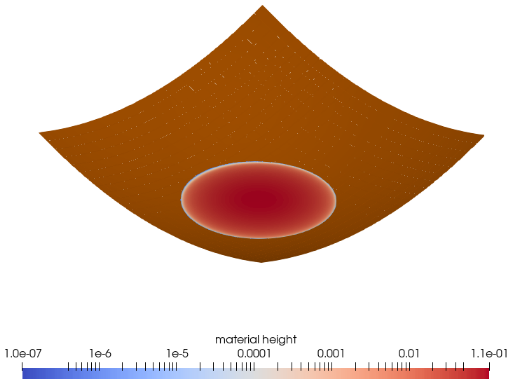


Figure 4.1: Thacker's test at time 0.0 with logarithmic scaling.

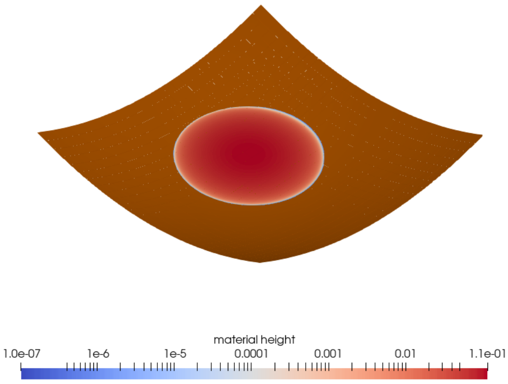
Three different solvers were chosen to simulate this scenario in ExaHyPE 2. The first employed solver is given by the Rusanov solver. The time evolution of its simulation is displayed in Figure 4.2. Between these four time steps, the granular material circles around the center of the basin while keeping its shape. This is the expected behaviour. The error compared to the analytical solution is displayed in Figure 4.3. The three respective errors display oscillatory behaviour, which seems to coincide with the periodic motion of the material. The L_∞ error reaches maximum values of ≈ 0.038, while both the L_1- and the L_2-error reach a maximum value of around 0.02. These values are quite high, which is usual for the simple one-wave Rusanov solver.



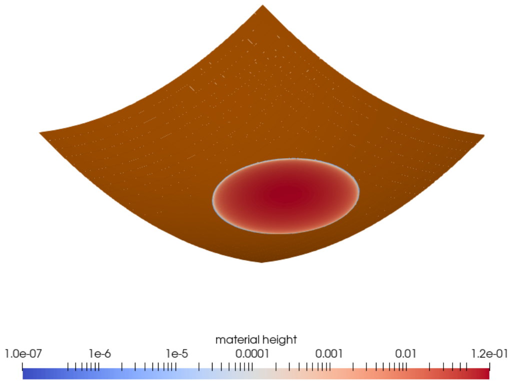
(a) t = 2.5s



(b) t = 5s



(c) t = 7.5s



(d) t = 10s

Figure 4.2: Time evolution of the granular material height $h(x, y, t)$ for Thacker's test scenario simulated with the generic Rusanov solver.

Secondly, the simulation was executed with the augmented Riemann solver. Its results are displayed in Figure 4.4 and Figure 4.5. Similarly to the Rusanov solver, the granular material displays a periodic motion, which is physically feasible. Furthermore, the material leaves behind a material trail corresponding to the value of the dry tolerance at $1e-4$. The errors for the augmented Riemann are increasing over time. The material trail is one possible explanation for this. Still, the L_1-error reaches a maximum value of ≈ 0.008. The L_2 and L_∞ errors do not surpass 0.004. Thus, all errors are much smaller than for the Rusanov solver, which underlines the accuracy of the augmented Riemann solver.

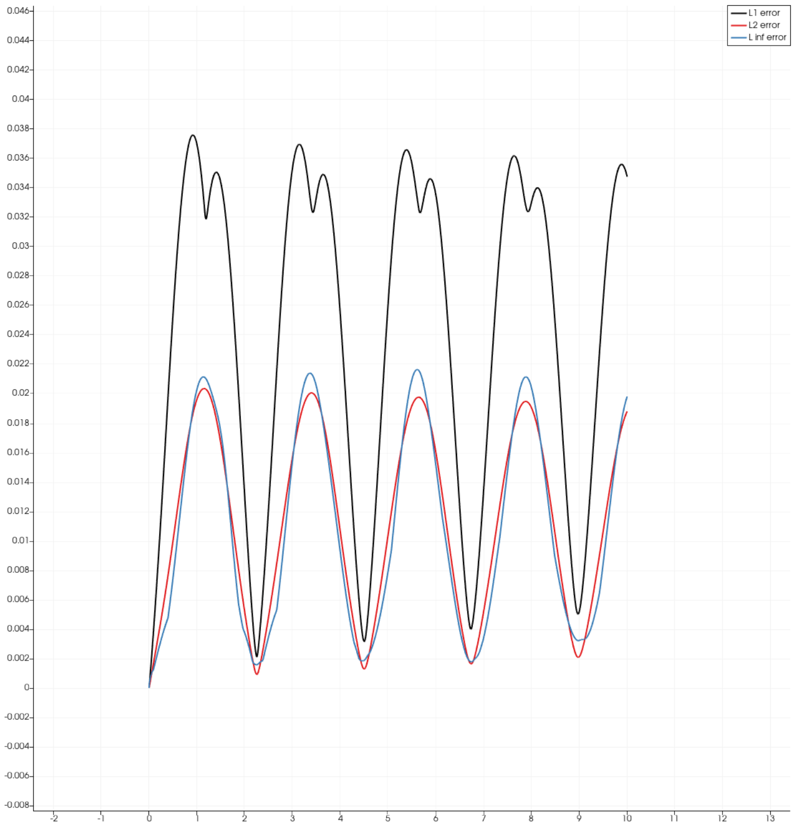
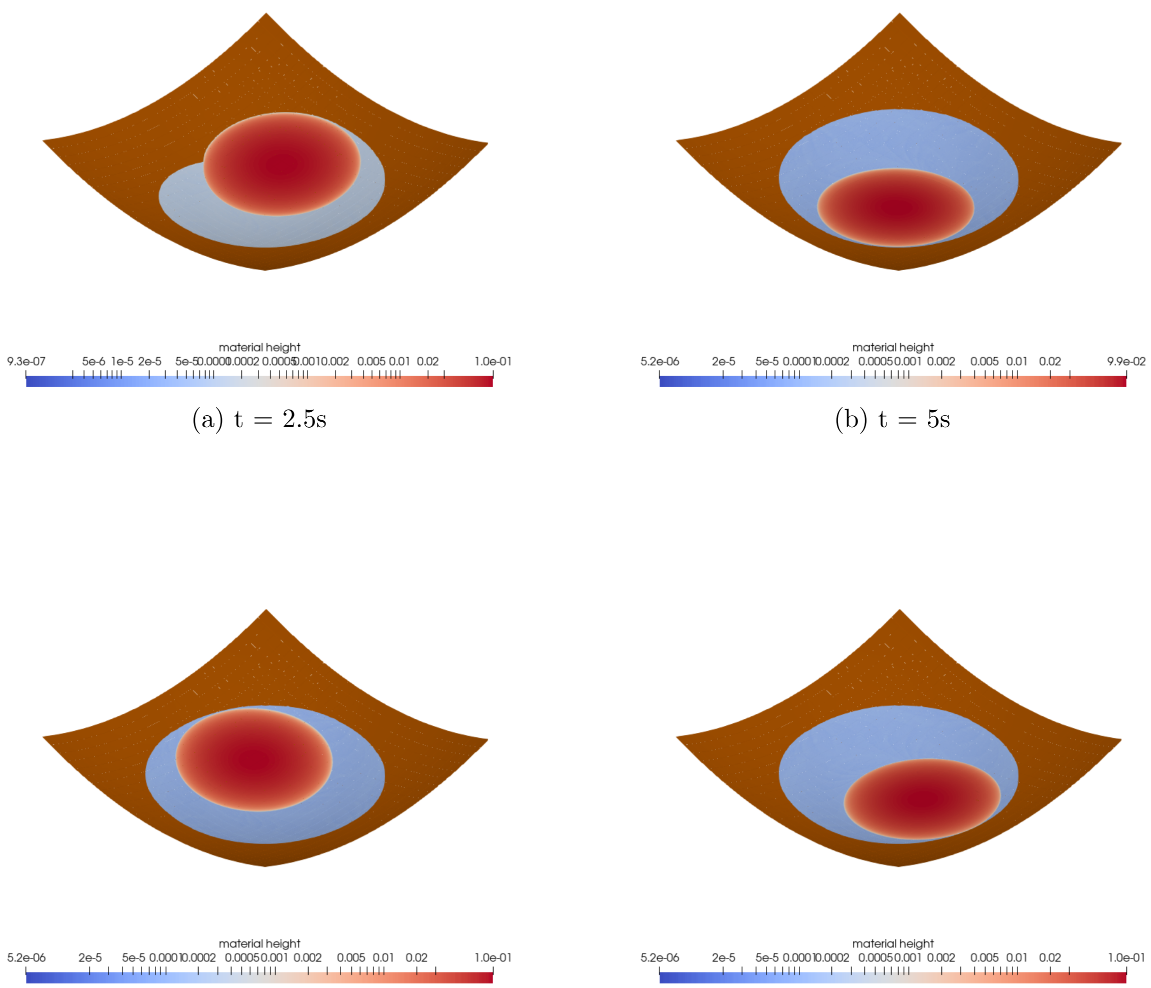


Figure 4.3: Time evolution of the error measurements for the Rusanov solver



(a) t = 2.5s (b) t = 5s

(c) t = 7.5s (d) t = 10s

Figure 4.4: Time evolution of the granular material height $h(x, y, t)$ for Thacker's test scenario simulated with the augmented Riemann solver.

The last utilized solver is the HLLC solver with hydrostatic reconstruction and middle-height-based wave speed estimates. As Figure 4.6 shows, it displays similar behaviour as the augmented Riemann solver concerning the material trail and the movement of the material. Figure 4.7 shows that its errors are also increasing over time. The respective error values are higher than for the augmented Riemann solver, but lower compared to the Rusanov solver. This was expected and shows the functionality of the HLLC solver and the hydrostatic reconstruction approach.

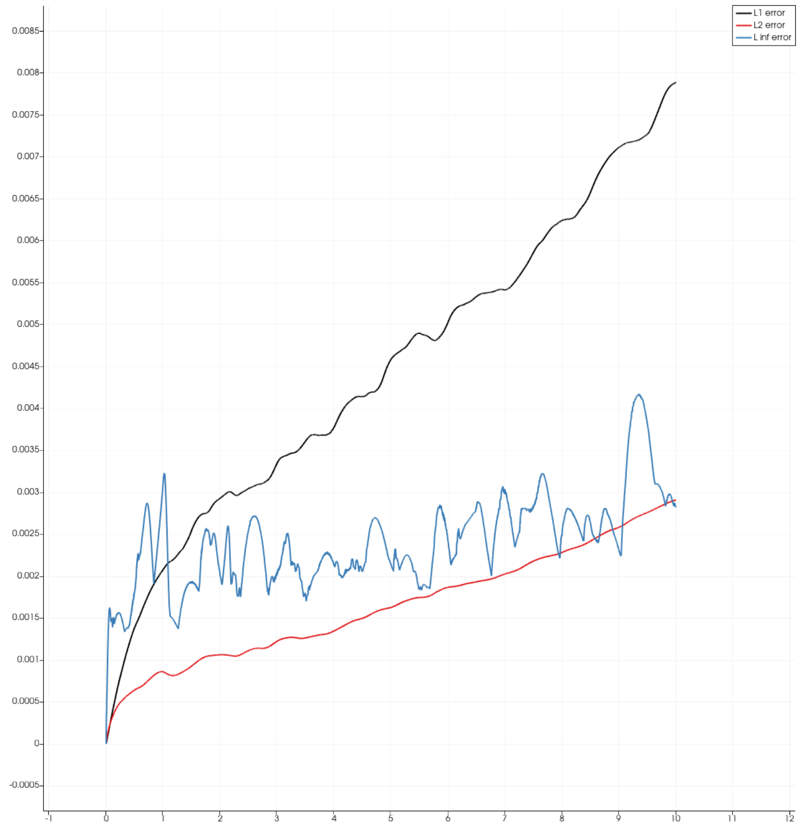
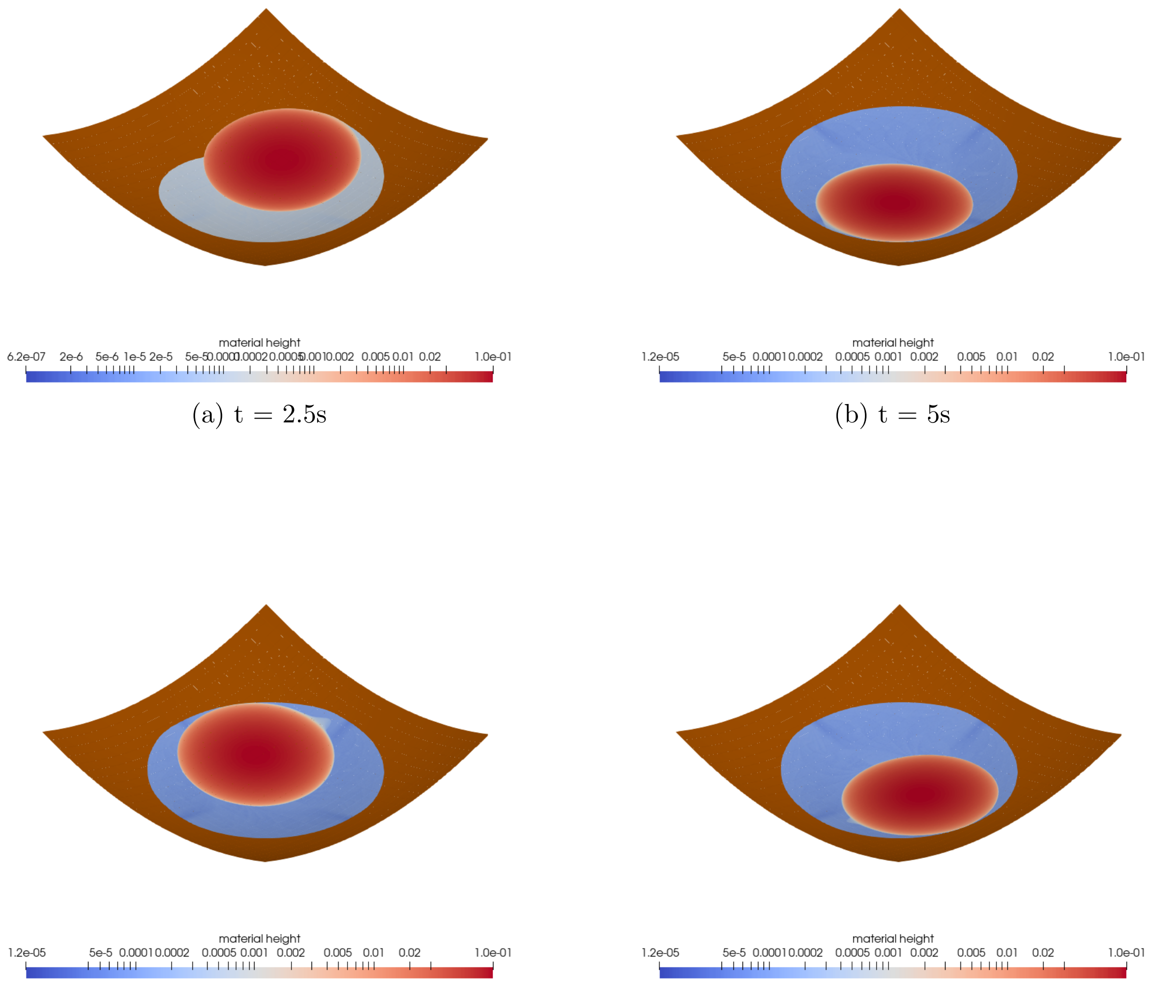


Figure 4.5: Time evolution of the error measurements for the augmented Riemann solver



(a) t = 2.5s

(b) t = 5s

(c) t = 7.5s

(d) t = 10

Figure 4.6: Time evolution of the granular material height $h(x, y, t)$ for Thacker's test scenario simulated with the HLLC solver with hydrostatic reconstruction and middle height-based wave speed estimates.

To conclude, as described in [21], the material is supposed to circle the center of the paraboloid. Every of the utilized solvers was able to reproduce this motion over a variable topography. The main disadvantage of the HLLC and augmented Riemann solver is the necessity of a dry tolerance. Still, they obtain better results than the Rusanov solver. Furthermore, the omitting of the dry tolerance and the robustness makes the Rusanov solver still an attractive choice.

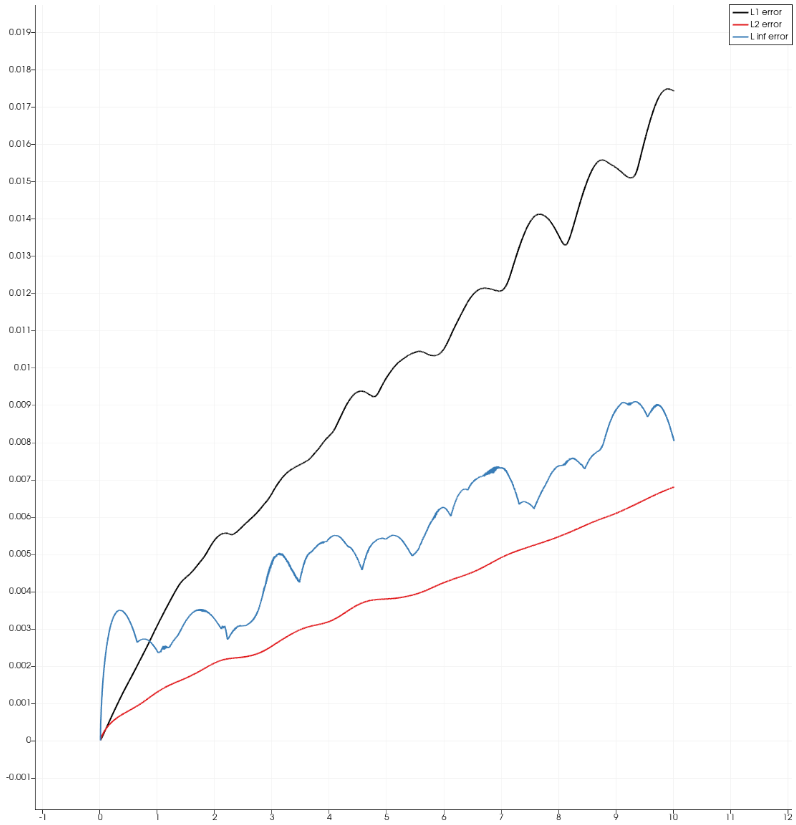


Figure 4.7: Time evolution of the error measurements for the HLLC solver

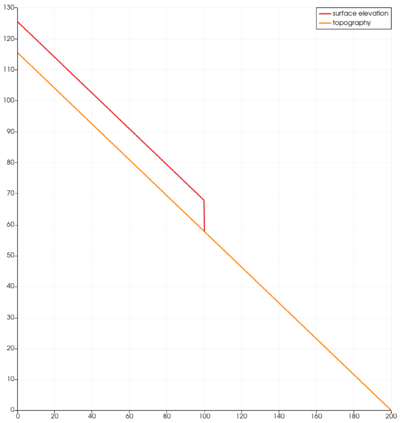
4.2 A Simplified Landslide Scenario

In this scenario, the topography is given by a perfect slope. On this slope, granular material with constant height is at rest for the initial condition until an idealized instant dam-break sets in. The inspiration for this scenario was given in [13], where several benchmarking scenarios were investigated, and in [17], where an analytical solution for a related scenario was developed. The main difference between the ExaHyPE 2 implementation and the scenarios of the aforementioned papers is that, in ExaHyPE 2, the material height is initialized normal to the global x-axis, whereas, in these papers, the material was initialized normal to the slope. Furthermore, these papers implemented topography-linked governing equations, whereas the ExaHyPE 2 implementation works with a global coordinate system. Just as in [13], the material height is set to 10 meters, and the slope angle is given by $\theta = 30$. Similar to [17], the incorporated friction force is given by the Coulomb friction law. Thus, this scenario is suitable for developing granular flows. Furthermore, it also poses a good test for the implementation of the wetting and drying mechanisms and the incorporation of the source term. Table 4.2 displays the concrete simulation parameters. The respective initial-dam positions are located at the center of the x-component of the spatial domain. Reflective boundary conditions were imposed.

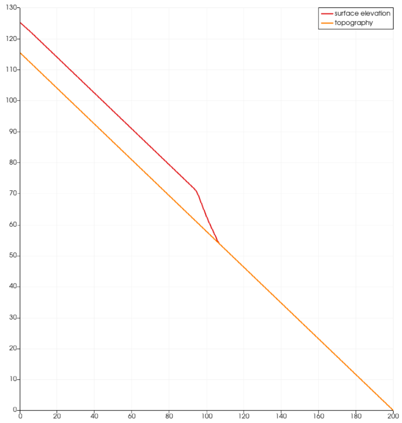
min-depth	patch-size	end-time	width	offset	amr-levels	dry-tol
6	32	10.0	[200.0, 200.0]	[0.0, 0.0]	0	1e-3

Table 4.2: Simulation parameters for the landslide scenario

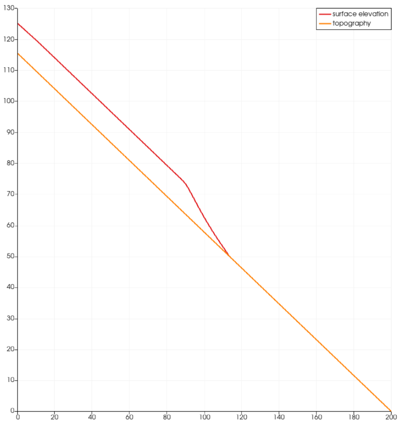
The generic Rusanov solver was employed first. The respective surface elevation time evolution is displayed in Figure 4.8, and the time evolution of the material height is illustrated in Figure 4.9. In these figures, one can observe the single rarefaction that develops as a result of the initial material discontinuity. This shape of this rarefaction is too step, which can be explained as the Rusanov solver does not incorporate information concerning the wet or dry state of the respective cells. Still, because of the simple geometry of the scenario, the Rusanov solver delivers satisfactory results.



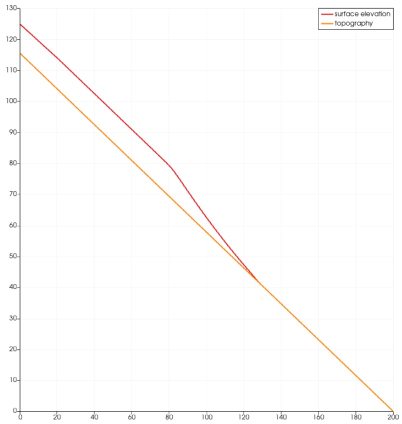
(a) t = 0.0s



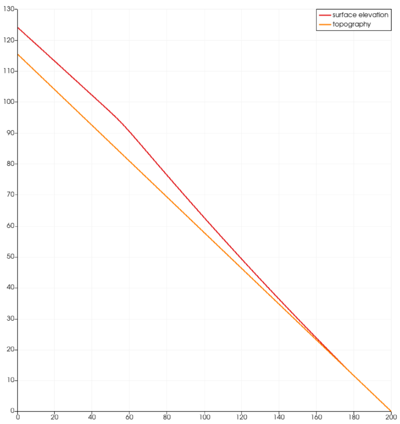
(b) t = 0.5s



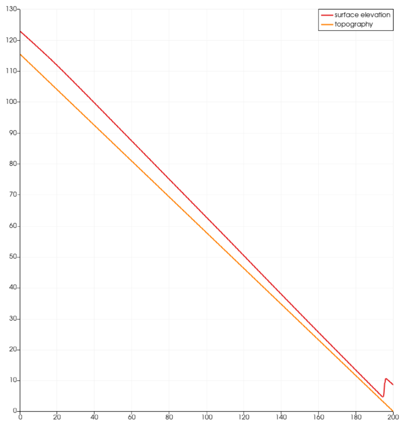
(c) t = 1.0s



(d) t = 2.0s

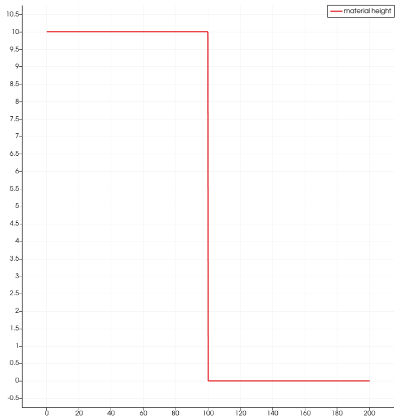


(e) t = 5.0s

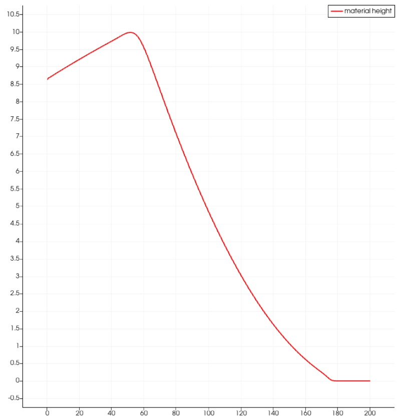


(f) t = 10.0s

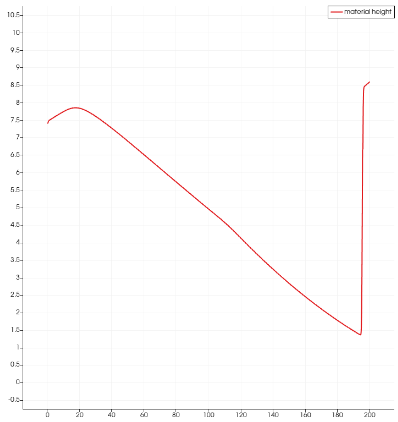
Figure 4.8: The time evolution of the surface elevation $s(x, y, t)$ of the idealized landslide scenario simulated with a generic Rusanov solver.



(a) t = 0.0s



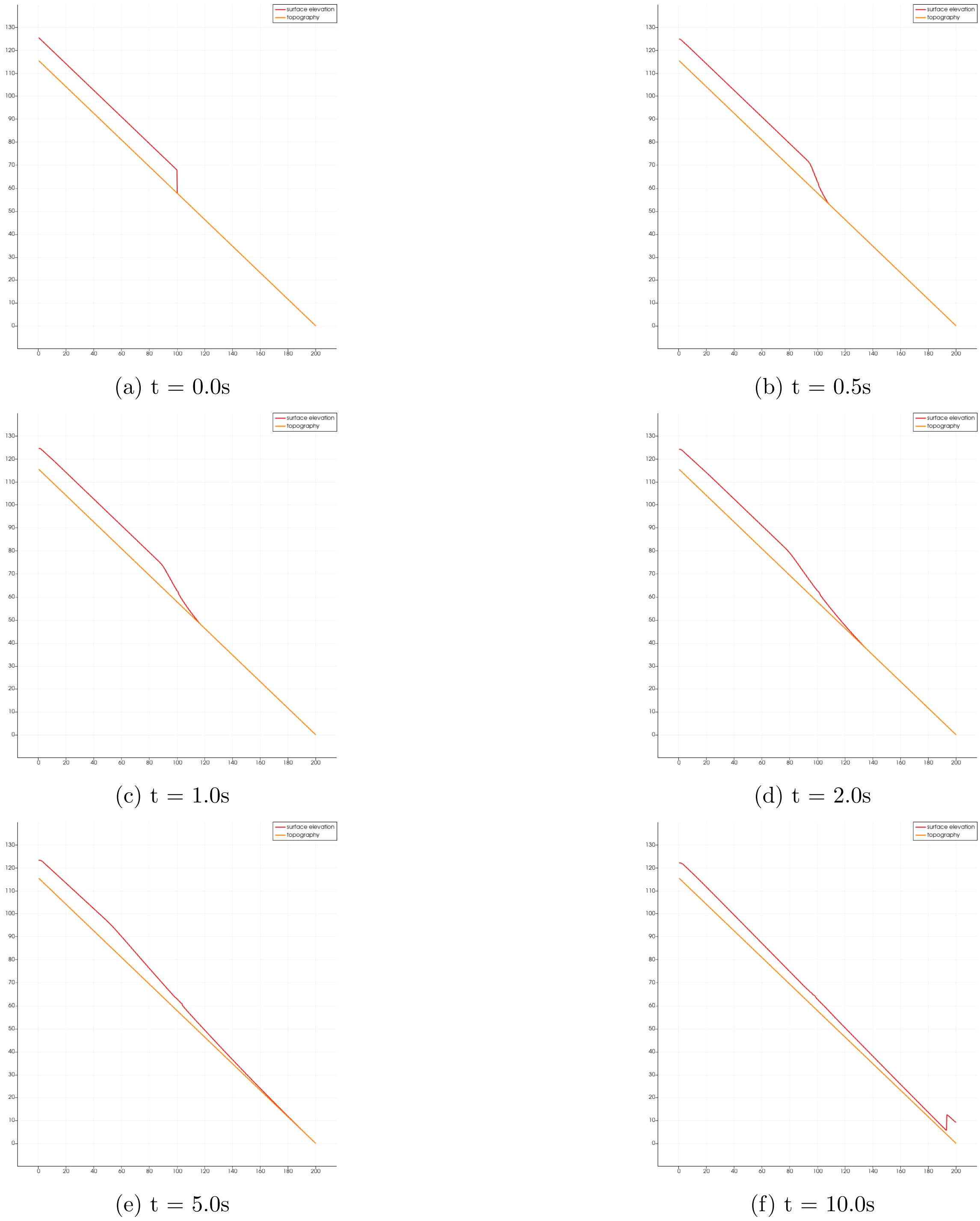
(b) t = 5.0s



(c) t = 10.0s

Figure 4.9: The time evolution of the material height $h(x, y, t)$ of the idealized landslide scenario simulated with a generic Rusanov solver.

The second employed solver for this scenario is the augmented Riemann solver. Again, the surface elevation time evolution is displayed in Figure 4.10, and the time evolution of the material height is illustrated in Figure 4.11. The displayed surface elevations and material heights are quite close to the ones from the Rusanov solver, even though the augmented Riemann solver is able to compute the speed of wet-dry fronts in an exact manner. Still, the augmented Riemann solver delivers the respected results concerning the development of a rarefaction connecting the area containing granular material with the dry area. Still, there seems to be an area with oscillating material heights with a small amplitude at $x \approx 95$. This numerical instability needs to be investigated further in the future.



(a) t = 0.0s

(b) t = 0.5s

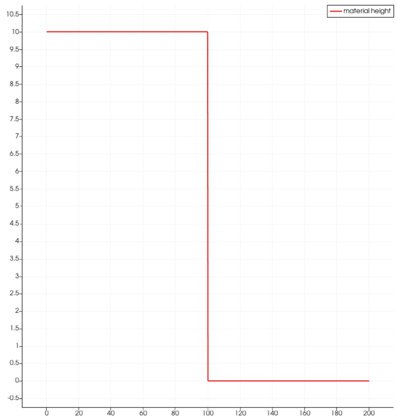
(c) t = 1.0s

(d) t = 2.0s

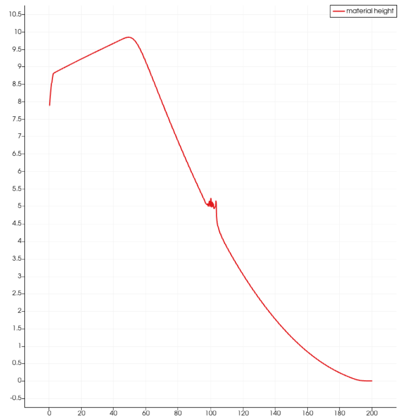
(e) t = 5.0s

(f) t = 10.0s

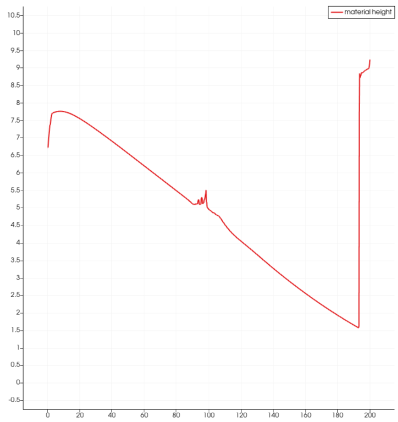
Figure 4.10: The time evolution of the surface elevation $s(x, y, t)$ of the idealized landslide scenario with the augmented Riemann solver.



(a) t = 0.0s



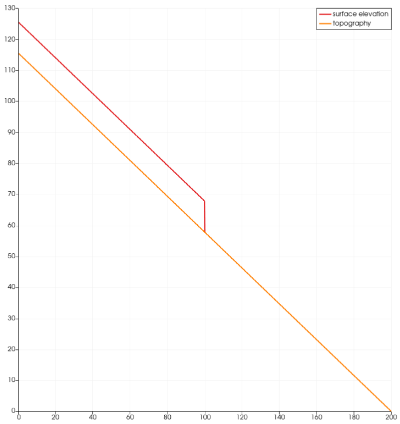
(b) t = 5.0s



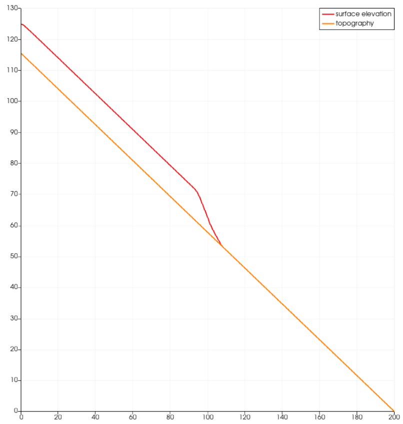
(c) t = 10.0s

Figure 4.11: The time evolution of the material height $h(x, y, t)$ of the idealized landslide scenario simulated with the augmented Riemann solver.

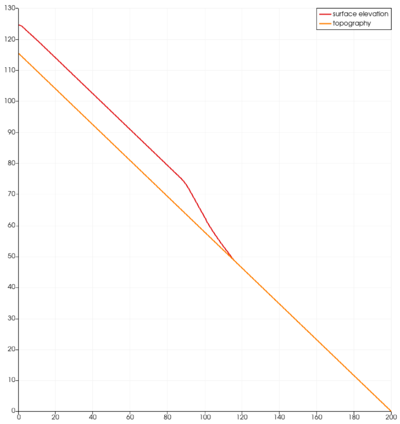
At last, the HLLC solver, coupled with hydrostatic reconstruction and middle height-based speed estimation, was deployed. For the middle height estimation, the aforementioned two rarefaction approximations were used. Figure 4.12 shows the time evolution of surface elevation, Figure 4.13 the time evolution of the granular material height. The results are similar to those from the two previously analyzed solvers. Thus, this solver also demonstrates its ability to handle wet-dry fronts and the incorporation of source terms.



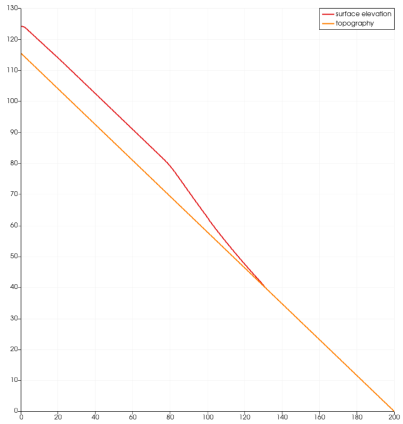
(a) t = 0.0s



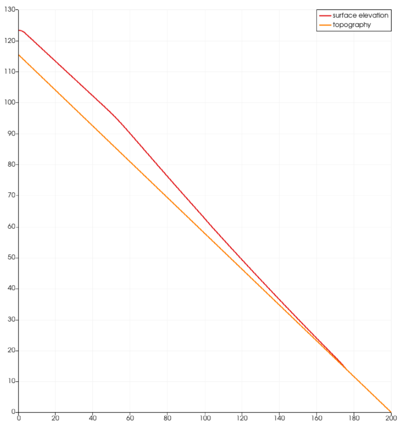
(b) t = 0.5s



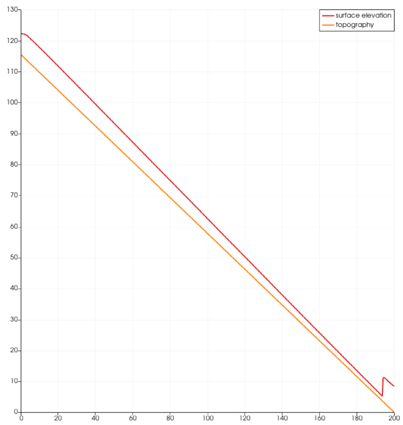
(c) t = 1.0s



(d) t = 2.0s

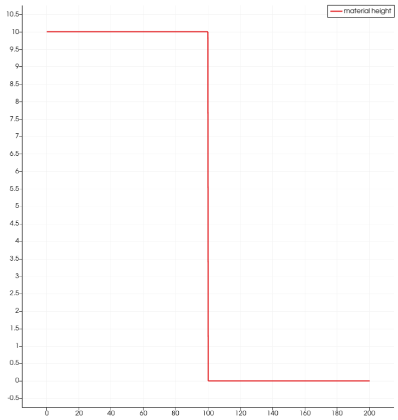


(e) t = 5.0s

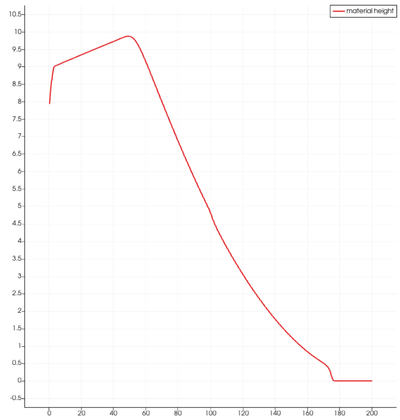


(f) t = 10.0s

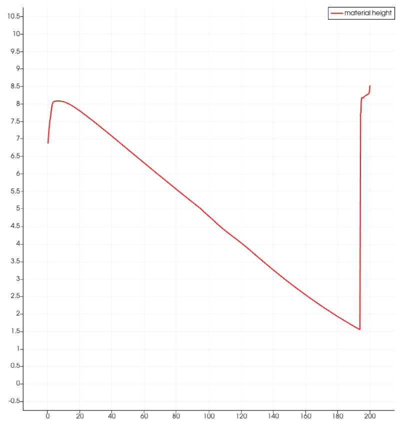
Figure 4.12: The time evolution of the surface elevation $s(x, y, t)$ of the idealized landslide scenario with the HLLC solver coupled with hydrostatic reconstruction and middle height-based wave speeds.



(a) t = 0.0s



(b) t = 5.0s



(c) t = 10.0s

Figure 4.13: The time evolution of the material height $h(x, y, t)$ of the idealized landslide scenario simulated with the HLLC solver coupled with hydrostatic reconstruction and middle height-based wave speeds.

One aspect that stands out between the simulations of the respective solvers is the smooth shape of the head of the rarefaction associated with the speed of the wet-dry front. The augmented Riemann solver captures its smooth shape the best at $x \approx 180$, as one can see in Figure 4.11 at time stamp 5.0s, compared to the respective time stamps of Figure 4.9 and Figure 4.13.

4.3 Landslide Scenario With a Topography Bump

This next scenario is related to the previously analyzed idealized landslide scenario. The only difference is, that in the interval $x = [120, 140]$, an additional bump, a jump in the topography, was added. This bump's elevation is constant, with the height being four meters above the elevation value at the initial bump beginning at $x = 120$. Figure 4.14 displays the topography for this scenario.

Table 4.3 defines the simulation parameters.

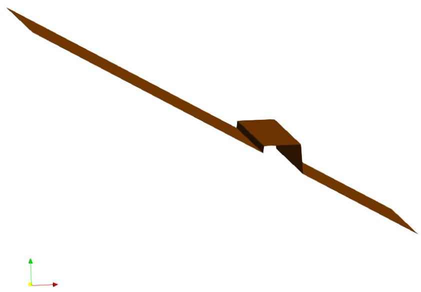
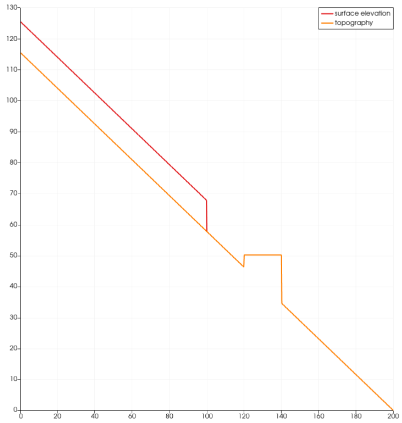


Figure 4.14: Topography of the Landslide with a bump

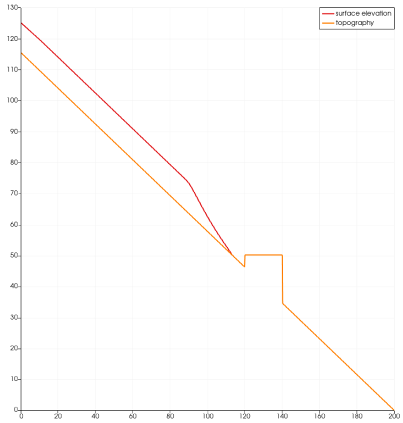
min-depth	patch-size	end-time	width	offset	amr-levels	dry-tol
6	32	15.0	[200.0, 200.0]	[0.0, 0.0]	0	1e-3

Table 4.3: Simulation parameters for the landslide scenario with an integrated jump in the topography

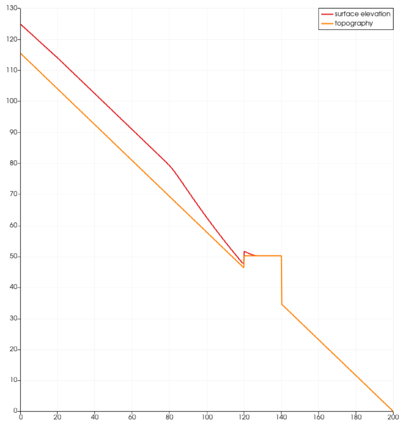
As in the scenario above, a plot displaying the time evolution of the surface elevation $s(x, y, t)$ is employed in order to investigate the respective solvers. Figure 4.15 displays the results obtained by executing the simulation with the generic Rusanov solver. In the first steps, the dry-wet front develops similarly as in the scenario with a constant inclined slope. At time stamp $t = 2.0$, one can observe how the granular material collides with the wall represented by the bump. In the following time steps, the material passes over the bump. At $x \approx 120$, an oscillation of the material height seems to develop.



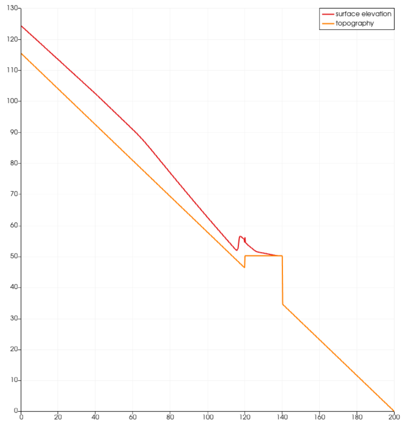
(a) t = 0.0s



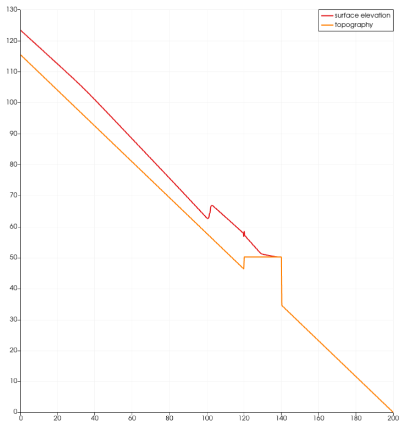
(b) t = 1.0s



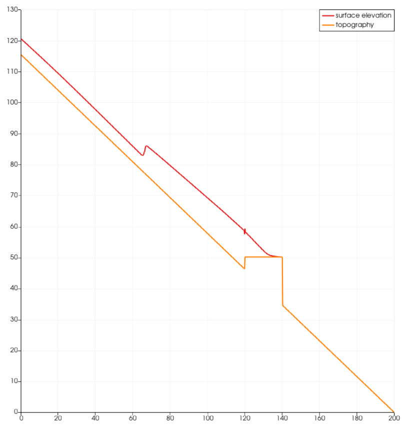
(c) t = 2.0s



(d) t = 4.0s



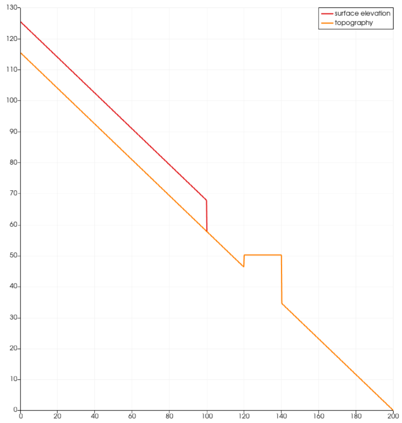
(e) t = 8.0s



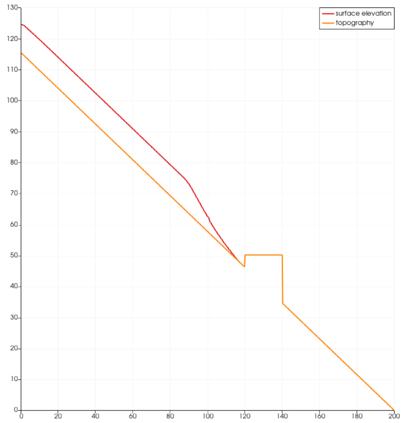
(f) t = 15.0s

Figure 4.15: The time evolution of the surface elevation $s(x, y, t)$ of the idealized landslide scenario with a bump in the topography simulated with a generic Rusanov solver.

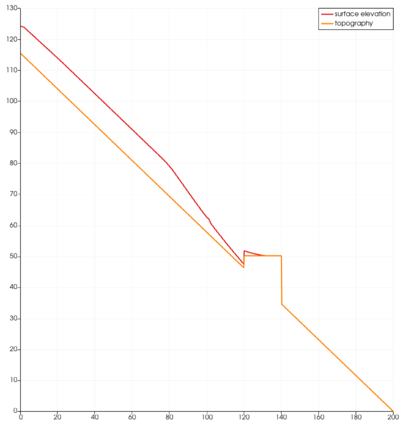
As before, the augmented Riemann solver delivers comparable results to the Rusanov solver, as displayed in Figure 4.16. The augmented Riemann solver incorporates a feature that determines whether the momentum of the material can overcome a wall. This feature works, as one can observe at the timestamp $t = 2.0$. As the Rusanov solver, the material inundates the area behind the topography jump step by step.



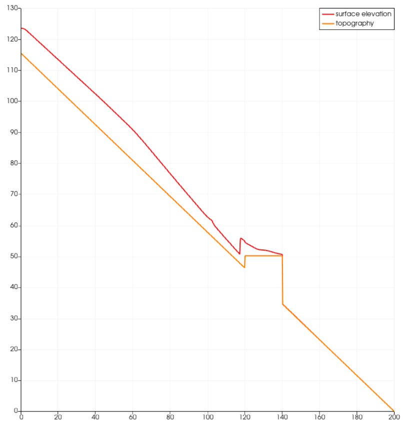
(a) t = 0.0s



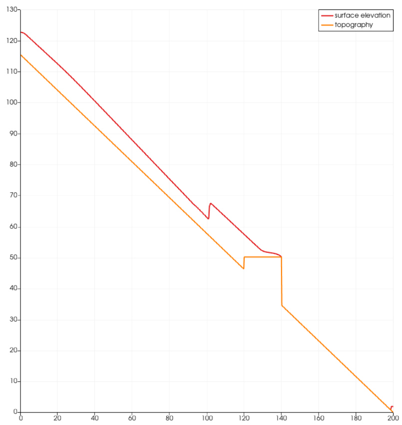
(b) t = 1.0s



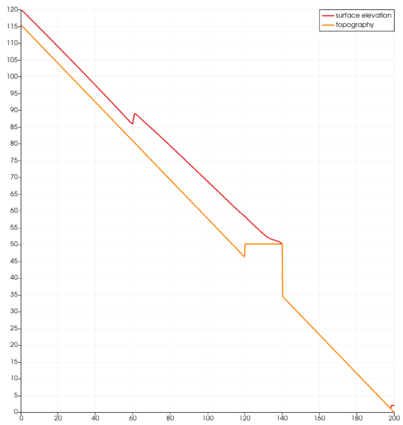
(c) t = 2.0s



(d) t = 4.0s



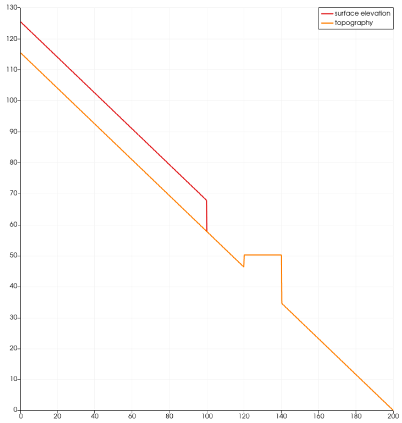
(e) t = 8.0s



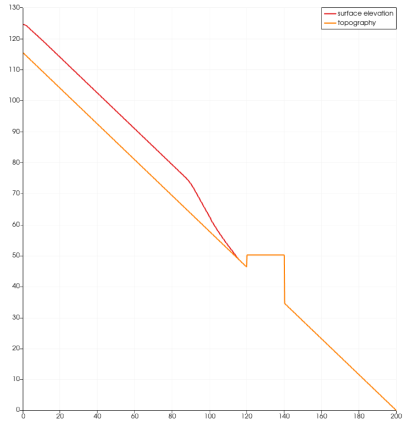
(f) t = 15.0s

Figure 4.16: The time evolution of the surface elevation $s(x, y, t)$ of the idealized landslide scenario with a bump in the topography simulated with the augmented Riemann solver.

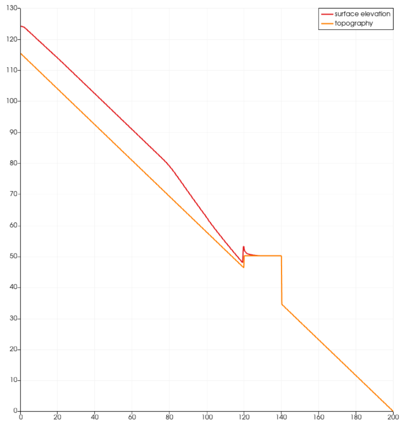
Again, the last utilized solver for this scenario is the HLLC solver with hydrostatic reconstruction and wave speed estimation based on the star region height caused by two outer rarefactions. Its results are displayed in Figure 4.17. It again delivers similar results; thus, the HLLC-coupled hydrostatic reconstruction approach is also able to deal with topography walls as long as the momentum of the material is large enough.



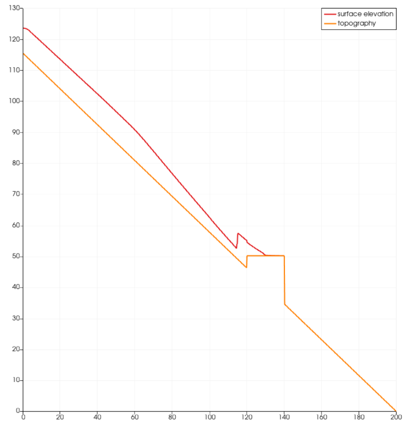
(a) t = 0.0s



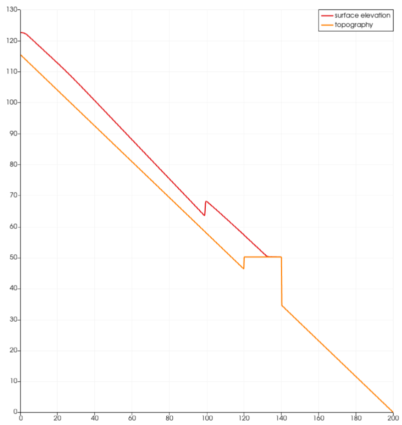
(b) t = 1.0s



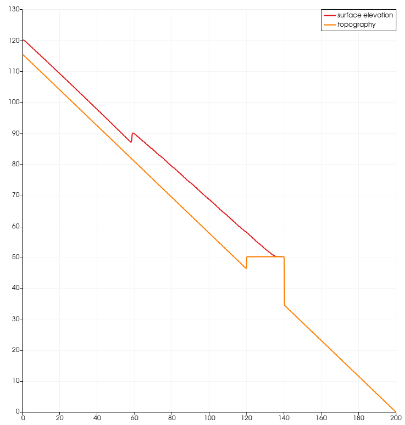
(c) t = 2.0s



(d) t = 4.0s



(e) t = 8.0s



(f) t = 15.0s

Figure 4.17: The time evolution of the surface elevation $s(x, y, t)$ of the idealized landslide scenario with a bump in the topography simulated with the HLLC solver coupled with hydrostatic reconstruction and middle height-based wave speeds.

Every solver delivered satisfactory results for this simulation scenario. For the Rusanov and HLLC solver, the material does not reach the area behind the end of the topography bump. This could be because the momentum flux and the Coulomb friction balance out and stop the material. For the augmented Riemann solver, one can see that the material reaches the reflective boundary, thus overcoming the end of the bump. This can be seen by the material gathering at the reflective boundary at $x \approx 200$ for the timestamps $t = 10$ and $t = 15$ in Figure 4.16.

4.4 The Tafjord Landslide Disaster

In addition to the testing with the previous artificial scenarios, it is necessary to utilize the implemented numerical methods in a realistic scenario. Thus, this section will investigate the accuracy of the numerical methods introduced above by simulating the landslide taken from the historical Tafjord landslide disaster. This disaster occurred in 1934 in Norway and consisted of a landslide and a tsunami caused by this landslide. It cost several human lives. In the present, the danger of this class of events, namely landslides flowing into Fjords, is still omnipresent in Norway [8]. Thus, employing numerical simulations to better understand these scenarios is paramount.

For this validation case, only the granular flow phase of said scenario will be investigated. The governing equations are given by the simplified landslide model, and again three different solvers will be tested. Table 4.4 specifies the concrete simulation parameters.

min-depth	patch-size	end-time	width	offset	amr-levels
5	16	50.0	[1900, 1950]	[414895.5, 6904495.5]	0

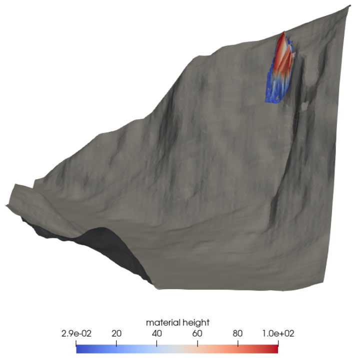
Table 4.4: Simulation parameters for the Tafjord scenario

This scenario constitutes a highly complex geometry with rapidly changing gradients of the bottom topography. Furthermore, the initial landslide volume will inundate a large previously dry area. Therefore, it is the most challenging simulation conducted in this thesis. Thus it also poses an important benchmark for the stability and accuracy of the solvers.

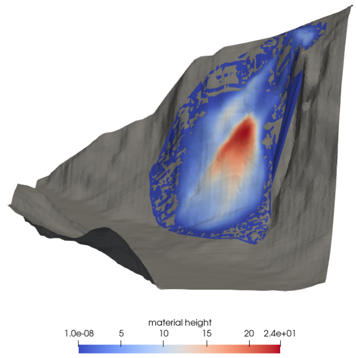
For this scenario, the generic Rusanov solver and the augmented Riemann solver will be used. The third employed solver is the HLLC solver coupled with hydrostatic reconstruction. The utilized wave speeds are given by the Einfeldt speed estimates (HLLC Einfeldt solver). During the simulations, it emerged that these more simple wave speed estimates provided better stability for this validation case.

For the augmented Riemann solver, the dry tolerance is set to $1e-2$; for the HLLC Einfeldt solver, it is set to $1e-3$.

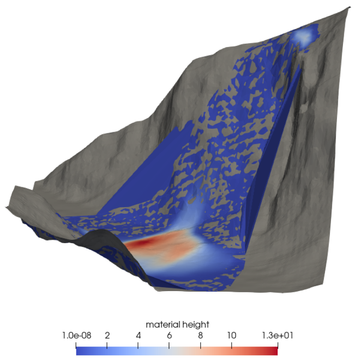
At first, this work investigates the Rusanov solver employed in this simulation scenario. Figure 4.18 shows the time evolution of the granular material atop the topography. One can observe that the material flows down the steep, irregular slope; after 30 seconds, only minimal further movement occurs, as the majority of the material has gathered at a low point of the topography.



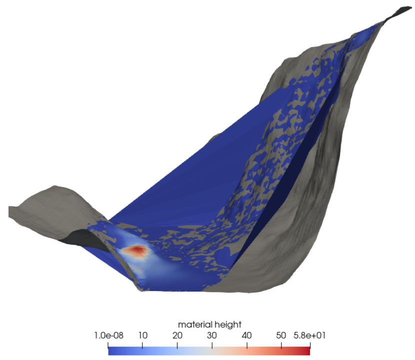
(a) t = 0.0s



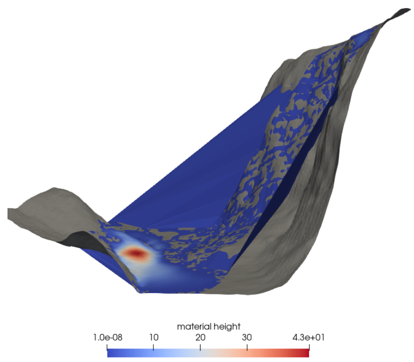
(b) t = 10.0s



(c) t = 20.0s



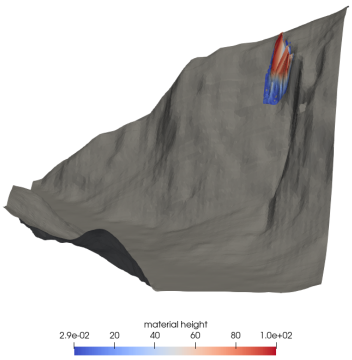
(d) t = 30.0s



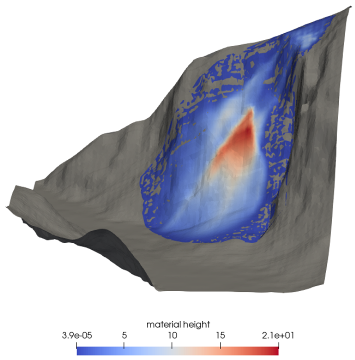
(e) t = 50.0s

Figure 4.18: The time evolution of the granular material height $h(x, y, t)$ of the Tafjord scenario simulated with the generic Rusanov solver. The perspective is changed for $t >= 30s$ to better display the runout.

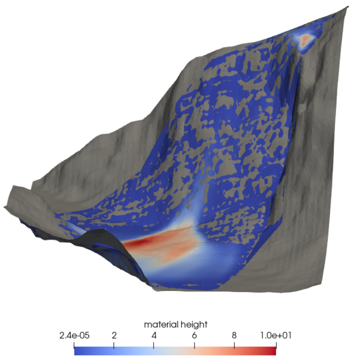
Figure 4.19 displays the same simulation, now equipped with the augmented Riemann solver. The results are very similar to those obtained with the generic Rusanov solver, showing that this solver can also handle complex cases.



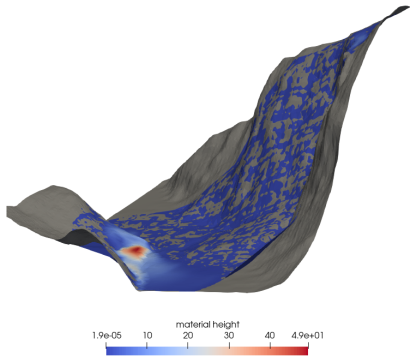
(a) t = 0.0s



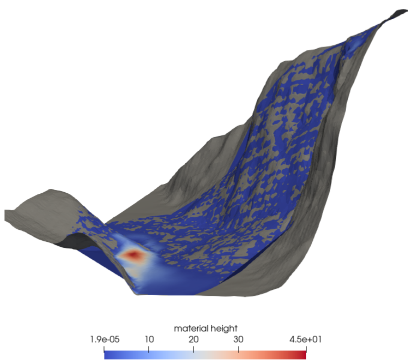
(b) t = 10.0s



(c) t = 20.0s



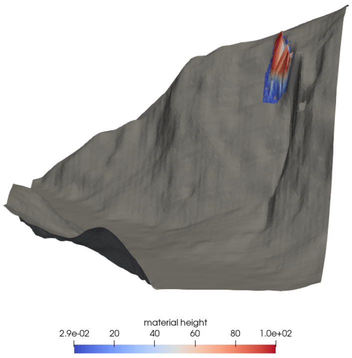
(d) t = 30.0s



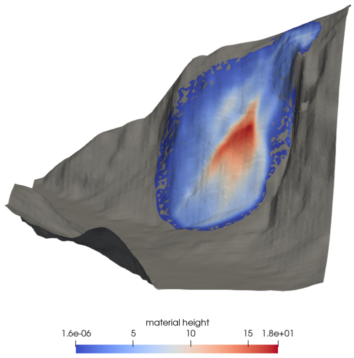
(e) t = 50.0s

Figure 4.19: The time evolution of the granular material height $h(x, y, t)$ of the Tafjord scenario simulated with the augmented Riemann solver. The perspective is changed for $t >= 30s$ to better display the runout.

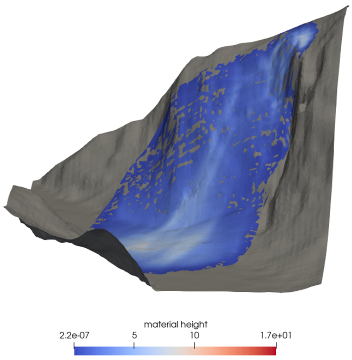
At last, Figure 4.20 displays the material height obtained by using the Einfeldt HLLC solver. Again, the solver manages to produce physically feasible results.



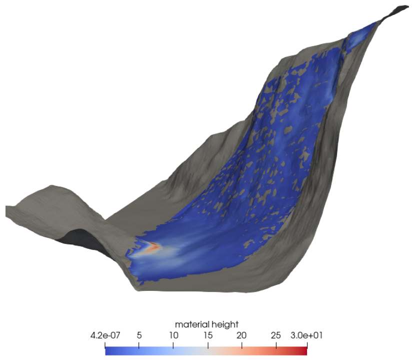
(a) t = 0.0s



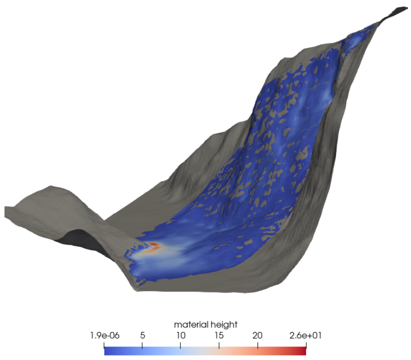
(b) t = 10.0s



(c) t = 20.0s



(d) t = 30.0s



(e) t = 50.0s

Figure 4.20: The time evolution of the granular material height $h(x, y, t)$ of the Tafjord scenario simulated with the Einfeldt HLLC solver. The perspective is changed for $t >= 30s$ to better display the runout.

To better compare the results of these different solvers, Figure 4.21 and Figure 4.22 compare the material height values at the timestamps $t = 10s$ and $t = 50s$ of the simulation.

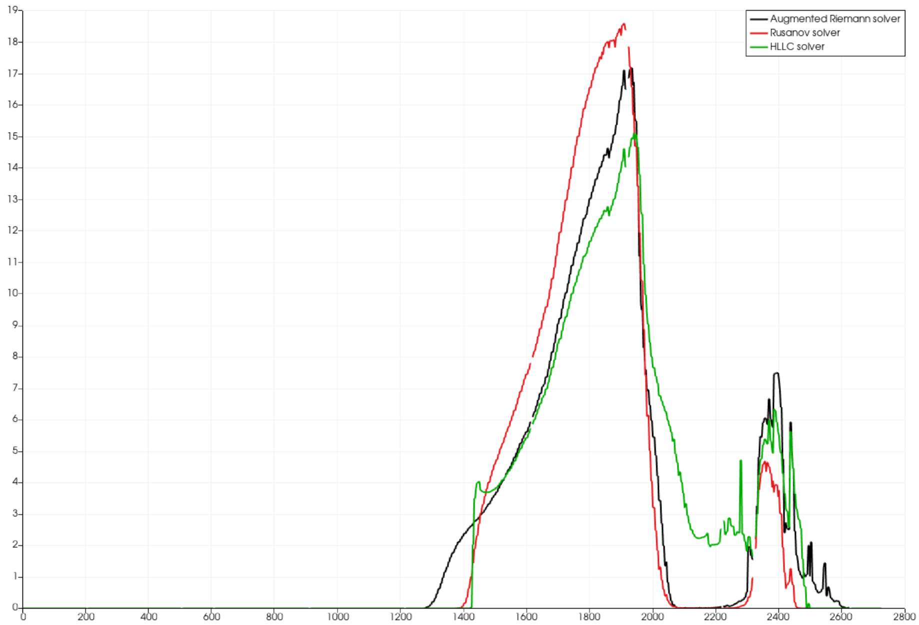


Figure 4.21: Line plot for the different material heights obtained by the respective solvers at time $t = 10.0$

The height values were obtained by a line plot spanning over the bottom left to the top right corner of the domain in order to capture the relevant physical processes. At time $t = 10s$, the solvers deliver similar results. One can recognize the material front developing as the material flows down the slope. For time $t = 50.0s$, the augmented Riemann solver and the Rusanov solver again show similar results. For the Einfeldt HLLC solver, the difference in the runout distance could be caused by the integrated local hydrostatic reconstruction approach. Furthermore, it displays a lower maximum material height.

Still, this validation case proved the ability of these solvers to handle complex scenarios. All of the solvers delivered physically feasible results, displaying granular material flowing down the slope of the Fjord and gathering at a low point of the topography.

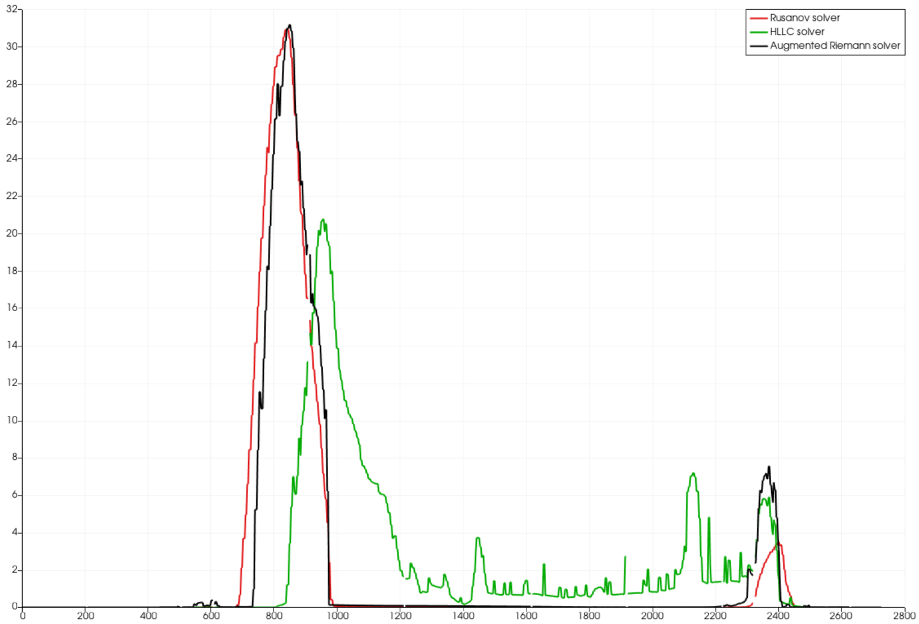


Figure 4.22: Line plot for the different material heights obtained by the respective solvers at time $t = 50.0$

5 Conclusion and Outlook

In this work, the basic principles of the two-dimensional shallow water equations were summarized. After introducing the respective governing models, the topic of Riemann problems and the respective solvers was a main focus. Riemann problems occur when one solves the two-dimensional shallow water equations on a Cartesian mesh when the data of the cells is piecewise constant. The respective solution to this problem can be calculated either exactly or approximately through approximate Riemann solvers, which are computationally less expensive. In this work, multiple concrete approximate Riemann solvers were investigated, and their respective numerical attributes were evaluated. Especially different HLLC solvers and an augmented Riemann solver were thematized. The more simple, but also very robust Rusanov Solver was also tested. The respective implementations are made available through ExaHyPE 2. After this, multiple validation cases were analyzed. These validation cases belong to either artificial scenarios, like the planar surface in a paraboloid case, or real-world examples, like the scenario of the Tafjord landslide disaster. With respect to these numerical simulations and results, one can conclude that all tested solvers, the Rusanov Solver, the HLLC solver, and the augmented Riemann Solver, were able to deliver satisfactory and physically feasible results. Still, even though they achieve better results than the Rusanov Solver, the main disadvantage of the more sophisticated HLLC Solver and augmented Riemann Solver is the necessity of incorporating a dry tolerance. Otherwise, numerical instabilities can occur, especially for complex topographies or incorporated source terms. It thus would be feasible to further improve the stability of these solvers. One way to increase the stability would be the implementation of transverse solvers like in Clawpack. For example, the very complex Tafjord case necessitated increasing the dry tolerance for a stable simulation with respect to the augmented Riemann solver. Another aspect for the future remains the implementation of second-order finite volume schemes, such as the MUSCL-Hancock scheme [3]. Even if one utilizes a very complex Riemann Solver, first-order schemes can often only provide limited accuracy. This can be seen, for example, by the still occurring error terms in the planar surface in a paraboloid case. The ADER-DG method, which is available as a high-order method in ExaHyPE 2, should be another focal point for future simulations. Still, the necessary flux limiting also requires an accurate finite volume method. With respect to the scenarios, more scenarios with analytical solutions, like different wave runup scenarios, will provide very good numerical tests for these solvers [12]. These tests are especially suitable for tsunami simulations. Furthermore, investigating more real world scenarios, concerning especially landslides and tsunamis, seems reasonable.

List of Figures

List of Tables

Bibliography

[1] Emmanuel Audusse, François Bouchut, Marie-Odile Bristeau, Rupert Klein, and Benoıt Perthame. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing*, 25(6):2050–2065, 2004.

[2] Hans-Joachim Bungartz, Stefan Zimmer, Martin Buchholz, and Dirk Pflüger. *Modeling and simulation: an application-oriented introduction.* Springer Science & Business Media, 2013.

[3] S. Clain, C. Reis, R. Costa, J. Figueiredo, M. A. Baptista, and J. M. Miranda. Second-order finite volume with hydrostatic reconstruction for tsunami simulation. *Journal of Advances in Modeling Earth Systems*, 8(4):1691–1713, 2016.

[4] Olivier Delestre, Carine Lucas, Pierre-Antoine Ksinant, Frédéric Darboux, Christian Laguerre, T-N-Tuoi Vo, Francois James, and Stéphane Cordier. Swashes: a compilation of shallow water analytic solutions for hydraulic and environmental studies. *International Journal for Numerical Methods in Fluids*, 72(3):269–300, 2013.

[5] Luigi Fraccarollo and Eleuterio F. Toro. Experimental and numerical assessment of the shallow water model for two-dimensional dam-break type problems. *Journal of Hydraulic Research*, 33(6):843–864, 1995.

[6] David L George. *Finite volume methods and adaptive refinement for tsunami propagation and inundation.* University of Washington, 2006.

[7] David L. George. Augmented riemann solvers for the shallow water equations over variable topography with steady states and inundation. *Journal of Computational Physics*, 227(6):3089–3113, 2008.

[8] CB Harbitz, S Glimsdal, F Løvholt, V Kveldsvik, GK Pedersen, and A Jensen. Rockslide tsunamis in complex fjords: From an unstable rock slope at åkerneset to tsunami risk in western norway. *Coastal engineering*, 88:101–122, 2014.

[9] Alexander Kurganov and Guergana Petrova. A second-order well-balanced positivity preserving central-upwind scheme for the saint-venant system. 2007.

[10] Randall J LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

[11] Ying Li and Fredric Raichlen. Solitary wave runup on plane slopes. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 127(1):33–44, 2001.

[12] Ying Li and Fredric Raichlen. Non-breaking and breaking solitary wave run-up. *Journal of Fluid Mechanics*, 456:295–318, 2002.

[13] Antoine Lucas, Anne Mangeney, François Bouchut, Marie-Odile Bristeau, and Daniel Mège. Benchmarking exercises for granular flows. *arXiv preprint arXiv:2106.13572*, 2021.

[14] J. Macías, C. Escalante, and M. J. Castro. Multilayer-hysea model validation for landslide-generated tsunamis – part 2: Granular slides. *Natural Hazards and Earth System Sciences*, 21(2):791–805, 2021.

[15] Per A Madsen and Hemming A Schaeffer. Analytical solutions for tsunami runup on a plane beach: single waves, n-waves and transient waves. *Journal of Fluid Mechanics*, 645:27–57, 2010.

[16] Kyle T Mandli, Aron J Ahmadia, Marsha Berger, Donna Calhoun, David L George, Yiannis Hadjimichael, David I Ketcheson, Grady I Lemoine, and Randall J LeVeque. Clawpack: building an open source ecosystem for solving hyperbolic pdes. *PeerJ Computer Science*, 2:e68, 2016.

[17] ANNE Mangeney, PHILIPPE Heinrich, and ROGER Roche. Analytical solution for testing debris avalanche numerical models. *Pure and Applied Geophysics*, 157:1081–1096, 2000.

[18] Javier Murillo and P García-Navarro. Augmented versions of the hll and hllc riemann solvers including source terms in one and two dimensions for shallow flow applications. *Journal of Computational Physics*, 231(20):6861–6906, 2012.

[19] Sigrun Ortleb, Jonathan Lambrechts, and Tuomas Kärnä. *Wetting and DryingWetting and drying Procedures for Shallow Water Simulations*, pages 287–314. Springer International Publishing, Cham, 2022.

[20] Anne Reinarz, Dominic E Charrier, Michael Bader, Luke Bovard, Michael Dumbser, Kenneth Duru, Francesco Fambri, Alice-Agnes Gabriel, Jean-Matthieu Gallard, Sven Köppel, et al. Exahype: An engine for parallel dynamically adaptive simulations of wave problems. *Computer Physics Communications*, 254:107251, 2020.

[21] William Carlisle Thacker. Some exact solutions to the nonlinear shallow-water wave equations. *Journal of Fluid Mechanics*, 107:499–508, 1981.

[22] E. F. Toro. The HLLC Riemann solver. *Shock Waves*, 29(8):1065–1082, November 2019.

[23] Eleuterio F Toro. *Computational Algorithms for Shallow Water Equations*. Springer Cham, 2024.