A Non-Hydrostatic Shallow Water Model on Triangular Meshes in Sam(oa)^2

Guided Research Project

Philipp Samfaß
Fakulät für Informatik
Technische Universität München
Supervisor: Univ.-Prof. Dr. Michael Bader
Advisors: Dipl.-Inf. Oliver Meister, M.Sc. Kaveh Rahnema

April 16, 2015

For the simulation of tsunamis, the hydrostatic shallow water equations have established as a sound mathematical basis. However, due to the hydrostatic assumption, not all relevant physical effects—especially in coastal areas—can be modelled accurately. In this paper, we therefore show how to extend the PDE-framemwork sam(oa)^2 towards modified non-hydrostatic shallow water equations. We use the finite volume method based on Riemann solvers to solve the hydrostatic shallow water equations and a fractional step method for the inclusion of non-hydrostatic effects. After having explained the general numerical approach, we develop a dual grid discretization of the non-hydrostatic pressure equation on dynamically adaptive triangular grids. This will allow for a matrix-free solution of the corresponding system of linear equations. Finally, we show the validity of the new model based on four common test cases for non-hydrostatic shallow water models.

1. Introduction

Water waves are certainly among the most researched phenomena in natural sciences and engineering. Of particular interest to geoscientists and oceanographers is the accurate modeling of water wave propagation in tsunami scenarios. For this task, the non-linear hydrostatic 2D-shallow water equations have established as an appropriate mathematical basis. In their derivation (depth-integration of the Navier-Stokes equations), the vertical momentum equation is neglected and kinematic boundary conditions for the vertical velocity on the free surface and the bottom are used such that the vertical velocity

component drops out. Additionally, it is assumed that the pressure is given only by the hydrostatic pressure (being the result of gravity).

In general, this hydrostatic assumption applies only to waves fulfilling the shallow water characteristic $h/\lambda << 1$, where h is the height of the water column and λ the wave length [9]. Typically, tsunami waves propagate as such shallow water waves but change to deeper water waves when approaching the coast. While in the propagation phase, the influence of the vertical velocity component—a neglected quantity in the hydrostatic equations—is negligible compared to the horizontal one, its impact increases near the coast. Studies in [7] show that dispersive effects (dependency of the wave speeds on the frequency) have a significant influence on the run-up of tsunamis in coastal regions. Such effects are not part of the solutions produced by hydrostatic models [7].

Therefore, a different method based on modified non-hydrostatic shallow water equations has been successfully applied in various tsunami simulation frameworks, e.g. by [4],[5],[12] and [16]. In the non-hydrostatic shallow water equations, the vertical momentum equation is kept and the non-hydrostatic part of the pressure is now considered. At the same time, the depth-averaged nature of the hydrostatic shallow water equations is retained in order to preserve a 2D-model.

In [11], this approach has been adapted to and integrated into the finite volume solver SWE for the shallow water equations on a uniform cartesian grid. The goal of this research is to similarly extend the simulation package sam(oa)^2 (Space-filling curves and Adaptive Meshes for Oceanic And Other Applications)[1] which uses the same finite volume method but is based on adaptive triangular meshes. In this work, we present the numerical model for solving the non-hydrostatic shallow water equations and its realization on triangular meshes in sam(oa)^2 .

The remainder of this paper is organized as follows: first, a short review of the approach used for solving the non-hydrostatic shallow water equations is given. Then, the discretization of the non-hydrostatic pressure equation on triangular meshes in sam(oa)^2 is explained in detail. Finally, the results obtained with the new model are shown and a short summary and discussion of open questions for future research concludes this paper.

2. Numerical approach

2.1. Introduction: the non-hydrostatic correction

In the following, the non-hydrostatic correction shall be shortly reviewed. For a more self-contained and detailed discussion, please refer to [4],[5] and [11].

The basis for the method presented in this paper are the non-hydrostatic pressure

[1] https://github.com/meistero/Samoa

approximated shallow water equations in conservation law form (see [11] for a derivation):

$$\boxed{\begin{bmatrix} h \\ hU \\ hV \\ hW \end{bmatrix}_t + \begin{bmatrix} hU \\ hU^2 + \frac{1}{2}gh^2 \\ hUV \\ hUW \end{bmatrix}_x + \begin{bmatrix} hV \\ hUV \\ hV^2 + \frac{1}{2}gh^2 \\ hVW \end{bmatrix}_y = \begin{bmatrix} 0 \\ -ghb_x - \left([\frac{h\hat{q}}{2}]_x + \hat{q}b_x\right) \\ -ghb_y - \left([\frac{h\hat{q}}{2}]_y + \hat{q}b_y\right) \\ \hat{q} \end{bmatrix}.} \tag{1}$$

Here, $h(x,y) = \eta(x,y) - b(x,y)$ denotes the height of the water column, $\eta(x,y)$ the free surface elevation above the mean sea level, $b(x,y)$ the bathymetry, (U,V,W) the depth-averaged velocity vector, g the gravity of Earth and a subscript the partial derivative with respect to the given coordinate. Further, $q(x,y,z)$ represents the non-hydrostatic portion of the pressure and $\hat{q} = [q]_{z=b}$ the non-hydrostatic pressure at the sea bottom. Note that q will not be the real non-hydrostatic pressure but only a numerical correction parameter. This term stems from the pressure decomposition

$$p = p_H + q = g(\eta - z) + q \tag{2}$$

into a hydrostatic portion p_H and a non-hydrostatic part q which is used in the depth-integration of the Navier Stokes equations [2]. In the hydrostatic shallow water equations, $p = p_H$ is assumed instead. Stated differently, the hydrostatic shallow water equations are received as a special case of (1) by setting $q(x,y,z,t) = 0$ and neglecting the vertical momentum equation.

The non-hydrostatic shallow water equations (1) involve certain assumptions which are reviewed subsequently. We follow [16] and assume a linear vertical distribution of the non-hydrostatic pressure. Thus, for the depth-averaged non-hydrostatic pressure occuring during the depth-integration,

$$\frac{1}{h}\int_b^\eta q\,dz = \frac{1}{2}[q]_{z=b} \tag{3}$$

holds since the non-hydrostatic pressure vanishes at the surface. In addition, the vertical velocity component w is approximated linearly in z-direction [16]. In our model, the bathymetry is represented as a staircase. Hence, we can assume a zero vertical velocity at the bottom (cf. [4]) and W simplifies to:

$$W = \frac{1}{h}\int_b^\eta w\,dz = \frac{1}{2}[w]_{z=\eta}. \tag{4}$$

Thus, only the vertical velocity at the water surface will be computed. We therefore drop the subscript $z = \eta$ for simplicity in the remainder of this paper.

In order to solve the non-hydrostatic shallow water equations (1) with the introduced assumptions, we employ the commonly applied semi-implicit fractional step scheme: in each time step, the hydrostatic shallow water equations are solved with the finite volume method first. After this solution has been computed, the discharges hU and hV and the vertical velocity w at the surface will be corrected with the non-hydrostatic pressure

$[q]_{z=b} := \hat{q}$ at the bottom. This numerical parameter will be obtained by a pressure projection as introduced in [3]: it is computed as the solution to a system of linear equations in order to render each numerical control volume divergence-free. Naturally, the basis for the derivation of the system of linear equations will be the law of mass-conservation of the Navier-Stokes equations:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0. \tag{5}$$

In order to derive the system of linear equations for the non-hydrostatic pressure, correction formulas will be plugged into an integral form of (5) and discretized in space, see Sec. 2.2. These formulas express how the discharges/velocities have to be corrected with the non-hydrostatic pressure and follow from an Euler time-discretization of the governing equations (1):

$$(hU)^{n+1} = (\widetilde{hU})^{n+1} - \Delta t \left(\left[\frac{1}{2} h^n \hat{q}^{n+1} \right]_x + \hat{q}^{n+1} b_x \right) \tag{6}$$

$$(hV)^{n+1} = (\widetilde{hV})^{n+1} - \Delta t \left(\left[\frac{1}{2} h^n \hat{q}^{n+1} \right]_y + \hat{q}^{n+1} b_y \right) \tag{7}$$

$$w^{n+1} = \widetilde{w}^{n+1} + 2\Delta t \frac{\hat{q}^{n+1}}{h^{n+1}}. \tag{8}$$

Here, the tilde is used to represent the intermediate quantities computed by the hydrostatic solver in each time-step. Note that for the vertical velocity correction formula, the vertical momentum equation in (1) has been divided by h and the non-linear terms have been neglected (cf. [4]). Because of the latter approximation, a hydrostatic time step (corresponding to considering the equations for $q = 0$) does not have an impact on the vertical velocity (i.e. $\widetilde{w}^{n+1} = w^n$).

2.2. Discretization on Triangular Grids in sam(oa)^2

Introduction to sam(oa)^2

Before the spatial discretization of the non-hydrostatic pressure equation can be explained, some preliminary remarks on the simulation package sam(oa)^2 and its numerical model for solving the hydrostatic shallow water equations have to be made.

sam(oa)^2 is a parallel framework for solving partial differential equations on dynamically adaptive triangular meshes. It uses the Sierpinski space-filling curve in order to provide memory and cache-efficient algorithms for grid generation, refinement and traversal and supports finite element, finite volume and discontinuous-galerkin discretizations [10].

In this paper, we focus only on the SWE subpackage in sam(oa)^2 which implements the finite volume method for solving the 2D-hydrostatic shallow water equations based on the theory in [8]. For each finite volume cell or element i, a cell average or state $Q_i^n = (h_i^n, [hU]_i^n, [hV]_i^n)^T$ is determined. In order to compute Q_i^{n+1} during a timestep,

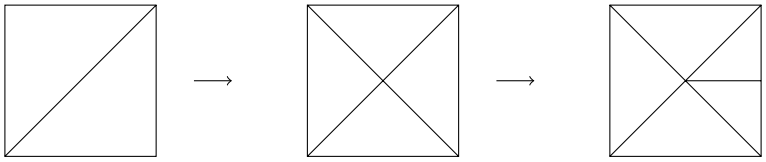

Figure 1: Example refinement: in the first step, both elements are refined while in the second, only the rightmost element is refined.

each Q_i^n is updated with the numerical fluxes which are obtained as the solutions to Riemann problems on the cell edges. The timestep Δt is restricted by the CFL-condition and thus varies depending on the scenario and the grid resolution.

Regarding the grid generation, sam(oa)^2 starts with a square that is divided by its diagonal into two isosceles right triangles (cells) as its computation domain. Initially, these two elements are then refined recursively into smaller isosceles right triangles through bisection until a sufficient grid resolution is obtained (see Fig. 1). Cells can also be refined during the simulation in the same way. The state of new cells generated in this process is determined using a simple interpolation according to the (constant) basis functions. On the other hand, the reverse process (i.e. coarsening) might happen as well: two right triangles of the same refinement depth are merged into a single one by computing an average of the two cell states.

A key concept used in sam(oa)^2 is the grid traversal through which access to the grid variables is granted. During a traversal, different operators are called such as the element operator ELEMENT_OP for each visited element. The NODE_FIRST_TOUCH and NODE_LAST_TOUCH operators allow for a different treatment of a node depending on whether it is visited for the first or for the last time in a traversal.

Dual grid discretization

The non-hydrostatic extension as reviewed previously introduces two additional degrees of freedom: the vertical velocity w at the water surface and the non-hydrostatic pressure correction parameter $\hat{q}$ at the bottom. In [11], these quantities were co-located with the other variables h, hU, hV in the centers of the rectangular cells. The non-hydrostatic correction formulas were discretized using finite differences to obtain the system of linear equations for $\hat{q}$. However, the realization of this approach is not directly applicable in sam(oa)^2. The adaptivity of the grid would be problematic for determining the differences since it is not clear how the mesh-width shall be chosen at a discontinuity between two non-uniform triangles. Yet, this would be necessary for the computation of the non-hydrostatic pressure gradient. Second, accessing neighbouring cell data would be required for the assembly of the global matrix for the non-hydrostatic pressure. However, the underlying traversal concept of sam(oa)^2 is strictly local: data access during an operator call is generally restricted to the data of the current element (element operator), node (node operators) or edge (edge operator).

In order to circumvent these difficulties, we propose to use a dual grid for the control volumes for the non-hydrostatic pressure. Instead of placing $w, \hat{q}$ in the cell center, we

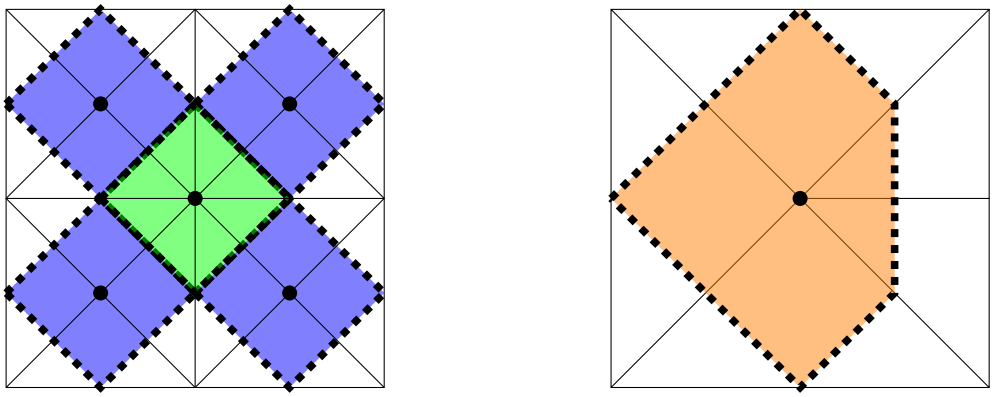

Figure 2: Control volumes/dual grid cells (dashed) for the non-hydrostatic pressure: the left picture shows the two kinds (green and blue) of dual grid cells that might occur in a grid of uniform refinement depth. On the right, a control volume for the non-uniform case is depicted. Note that an element will contribute to three different control volumes. The points denote the nodes on which the unknowns $\hat{q}_d$ and the vertical velocity w_d are located. For each $\hat{q}_d$, there is exactly one corresponding control volume/dual cell d.

position these variables on the element nodes. For each $\hat{q}$ on a given vertex, the control volume being rendered divergence-free with the continuity equation will then involve all the elements that are adjacent to this node (cf. Fig. 2). This will finally allow for a matrix-free solution of the system of linear equations.

We start deriving the system of linear equations by integrating (5) over a control volume/dual cell. By the Gaussian divergence theorem and the Leibniz rule, we get

$$\int_{\Gamma_d} n_x \, hU \, d\Gamma_d + \int_{\Gamma_d} n_y \, hV \, d\Gamma_d + \int_{\Omega_d} w \, d\Omega_d = 0 \tag{9}$$

where Γ_d denotes the boundary, $\mathbf{n} = (n_x, n_y)$ its outward pointing normal unit vector and Ω_d the region of the dual cell d. In (9), a surface integral had to be assumed as negligible, please see the short derivation in App. A for details.

Let N_d denote the set of primary elements that contribute to the dual cell d, $\Gamma_{e,d}$ the part of the boundary of d that lies in the element e and $\Omega_{e,d}$ the region of the dual cell that lies within e. Then, (9) can be assembled as:

$$\sum_{e \in N_d} \left(\int_{\Gamma_{e,d}} n_x \, hU \, d\Gamma_{e,d} + \int_{\Gamma_{e,d}} n_y \, hV \, d\Gamma_{e,d} + \int_{\Omega_{e,d}} w \, d\Omega_{e,d} \right) = 0. \tag{10}$$

The correction formulas will be plugged into the summand in (10). The thereby resulting equation corresponds to the contribution of an element e to the dual cell equation. One advantage of the dual arrangement is that the derivatives of h and b in normal direction over the dual cell boundary Γ_d are zero due to the constant basis functions used

in our finite volume model. Thus, we avoid having to evaluate derivatives at discontinuities and the correction formulas for the discharges hU, hV are reduced, since $\hat{q}\nabla b = 0$ and $\nabla h\hat{q} = h\nabla\hat{q}$ can be assumed for the boundary $\Gamma_{e,d}$. The cell equation contribution becomes then ($\cdot$ denotes the standard dot product):

$$\int_{\Gamma_{e,d}} \mathbf{n} \cdot \left[\begin{pmatrix} \widetilde{hU}^{(n+1)} \\ \widetilde{hV}^{(n+1)} \end{pmatrix} - \Delta t \frac{h^{(n)}}{2} \nabla \hat{q} \right] d\Gamma_{e,d} + \int_{\Omega_{e,d}} \widetilde{w}^{(n+1)} + 2\Delta t \frac{\hat{q}}{h^{(n+1)}} \, d\Omega_{e,d} = 0. \quad (11)$$

We choose constant basis functions for $\hat{q}$ and w which makes evaluating the integral (11) straightforward. Note that linear basis functions have been applied successfully as well which yielded qualitatively comparable results. However, a linear interpolation introduces additional dependencies which increase the stencil for "type green" dual cells from five to eight points in the uniform case. Additionally, the hydrostatic solver uses constant basis functions. It therefore seems reasonable to employ the same discretization for the non-hydrostatic correction.

Some attention shall be paid to the normal vectors and the non-hydrostatic pressure gradient. The standard reference element in Fig. 3 shows the partitioning of a primary element into the three dual cell contributions. Here, the normal vectors can easily be

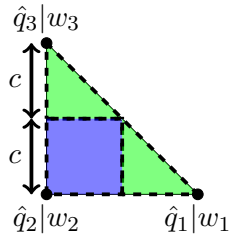

Figure 3: Standard reference element.

obtained and the non-hydrostatic pressure gradient is approximated by:

$$\nabla \hat{q} = \begin{pmatrix} \frac{\hat{q}_1 - \hat{q}_2}{2c} \\ \frac{\hat{q}_3 - \hat{q}_2}{2c} \end{pmatrix}. \quad (12)$$

In the general case, however, an element does not necessarily have the orientation of the standard reference element. Therefore, a respective rotation matrix is applied to the normal vectors and the non-hydrostatic pressure gradient. Altogether, this discretization procedure yields a 3×3 element matrix A with the dual cell equation contributions and a vector b for the right hand side of the system of linear equations. For further details and a short derivation involving the rotation of the normal vectors and the non-hydrostatic pressure gradient, please see App. B.

For first test-runs, we use a matrix-free Jacobi solver for the system of linear equations for $\hat{q}$. As soon as all elements contributing to a dual cell equation have been traversed, the residual of this equation is known and the respective unknown $\hat{q}_d$ can be updated

according to a Jacobi step. This happens exactly when the NODE_LAST_TOUCH operator is called during a Jacobi traversal.

After this system has been solved, the discharges $\widetilde{hU}, \widetilde{hV}$ are corrected according to the (reduced) correction formulas. However, special care has to be given to the correction of the vertical velocity. In order to be consistent with the system of linear equations, we compute a corrected average $\bar{w}_d$ of w over the dual cell d in each time step as follows:

$$\bar{w}_d^{(n+1)} := \frac{1}{|\Omega_d|}\int_{\Omega_d} w^{(n+1)} d\Omega_d = \underbrace{\frac{1}{|\Omega_d|}\int_{\Omega_d} \widetilde{w}^{(n+1)} d\Omega_d}_{=\bar{w}_d^{(n)}} + 2\Delta t \frac{1}{|\Omega_d|}\hat{q}_d \int_{\Omega_d} \frac{1}{h}\, d\Omega_d. \tag{13}$$

The global matrix has been investigated, although general statements about its properties cannot be made since the matrix weights depend on the water heights. However, the examined matrices of the validation experiments (see below) were diagonally-dominant and had a five-point stencil in the non-adaptive case. Additionally, by setting the water level h to an arbitrary constant and by neglecting the matrix weights that stem from the correction of w, a (scaled) system of linear equations in the form of the discretized Poisson equation is obtained.

Boundary conditions

Appropriate boundary conditions for $\hat{q}$ have to be set at closed boundaries, open boundaries and dry cells.

We follow [5] and impose the Neumann boundary condition $\frac{\partial \hat{q}}{\partial n} = 0$ at impermeable boundaries. For completely dry dual cells, the Dirichlet condition $\hat{q} = 0$ follows naturally. A difficulty arises for dual cells that consist both of dry and wet primary elements. Such dual cells therefore contain an open boundary. In this case, we set the Dirichlet condition $\hat{q} = 0$, too. Additionally, the quantities in primary elements that contribute to such a dual cell are not corrected. This corresponds to neglecting the influence of the non-hydrostatic pressure for these cells.

3. Results

Four numerical experiments that are commonly used to validate non-hydrostatic models have been conducted. In all experimental runs, the augmented Riemann solver from geoclaw [6] was used for the computation of the numerical fluxes. As a stopping criterion for the Jacobi-solver, we used a maximum absolute local residual threshold of $\epsilon = 0.1$.

Standing wave

The first experiment is a standing wave in a closed basin. It has also been conducted in related work such as [4], [5] and [12].

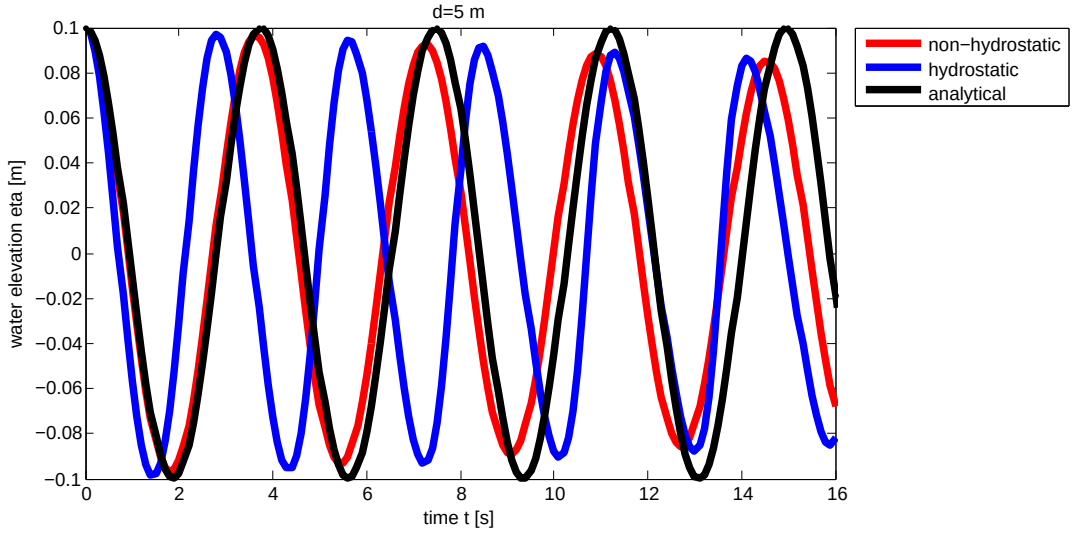

Figure 4: Standing wave: plot of the water elevation at $(x, y) = (10m, 5m)$ over time. 32768 cells were used for the uniform grid.

The basin has quadratic shape and a length of $10m$. For this scenario, the initial surface elevation is computed by [5]

$$\eta^0(x, y) = -a \cos\left(\frac{2\pi x}{\lambda}\right), \tag{14}$$

where the amplitude a is set to $1cm$ and the wave length λ to $20m$. The depth of the basin is $d = 5m$ and all four walls are impermeable. In this test case, the hydrostatic assumption no longer applies since d/λ is relatively high and the vertical acceleration has a non-negligible impact. The propagation phase speed c of the wave is therefore given by the linear dispersion relation from linear wave theory [12]

$$c = \sqrt{\frac{g\lambda}{2\pi} \tanh\left(\frac{2\pi d}{\lambda}\right)} \tag{15}$$

and with wave period $T = \lambda/c$, we can compute the analytical solution as

$$\eta(x, y, t) = -a \cos\left(\frac{2\pi x}{\lambda} + \frac{2\pi t}{T}\right). \tag{16}$$

In Fig. 4, the water elevation at the right border of the basin is plotted over time. Note that the hydrostatic solution produces an incorrect wave period while the non-hydrostatic one is in good agreement with the analytical solution. The damping of the amplitude decreases with higher grid resolution and therefore seems to be resulting from the discretization error.

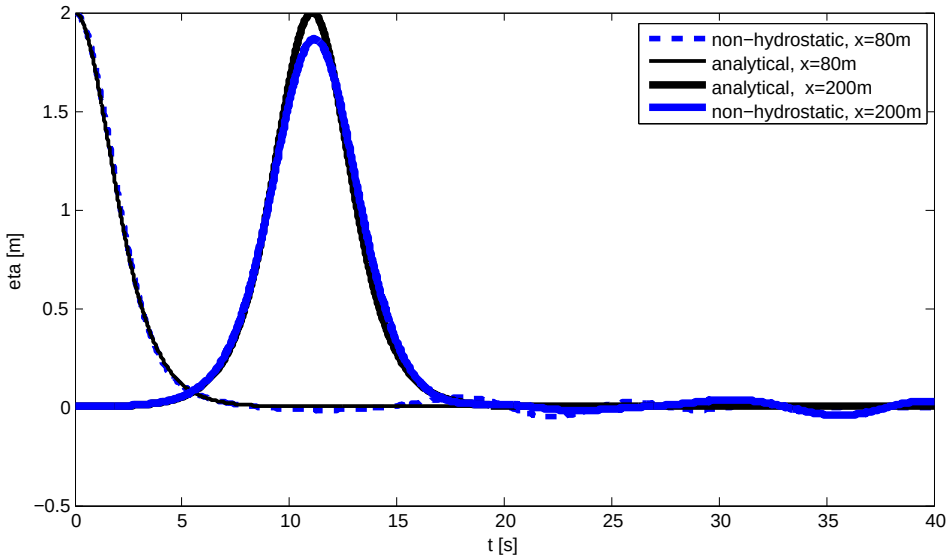

Figure 5: Solitary wave (#cells = 38400): water elevation over time at the points $(x_1, y_1) = (80, 0.25)$ and $(x_2, y_2) = (200, 0.25)$.

Solitary wave

As a second test case, we examined a solitary wave in a closed channel of length $600m$, width $2m$ and depth $d = 10m$. The solitary wave is a single elevation of the water surface that propagates from right to left. Since friction and viscosity are absent in the model, the wave should not deform while propagating. This experiment has become a standard test for non-hydrostatic models as the solitary wave is not a solution of the hydrostatic shallow water equations (see e.g. [4], [12], [16] and [17]).

With wave amplitude $a = 2m$, the wave speed is given by $c = \sqrt{g(a+d)}$. According to [12] and [17], the water elevation, horizontal (depth averaged) and vertical velocity can be computed analytically as

$$\eta(x,t) = a\,\mathrm{sech}^2\left(\sqrt{\frac{3a}{4d^3}}(x - x_s - ct)\right) \qquad U = c\frac{\eta}{h} \qquad w = -\eta\frac{\partial U}{\partial x} \tag{17}$$

where $x_s = 80m$ denotes the initial wave crest position. For $t = 0$, the equations in (17) specify the initial condition.

As shown in Fig. 5, the non-hydrostatic model gives a good approximation to the analytical solution. Fig. 6 shows that the hydrostatic model produces an incorrect sawtooth wave, while the non-hydrostatic model conserves the correct profile over time. A small reduction of the wave amplitude and trailing waves occur. However, these perturbations have also been obtained by other non-hydrostatic models, see e.g. [12].

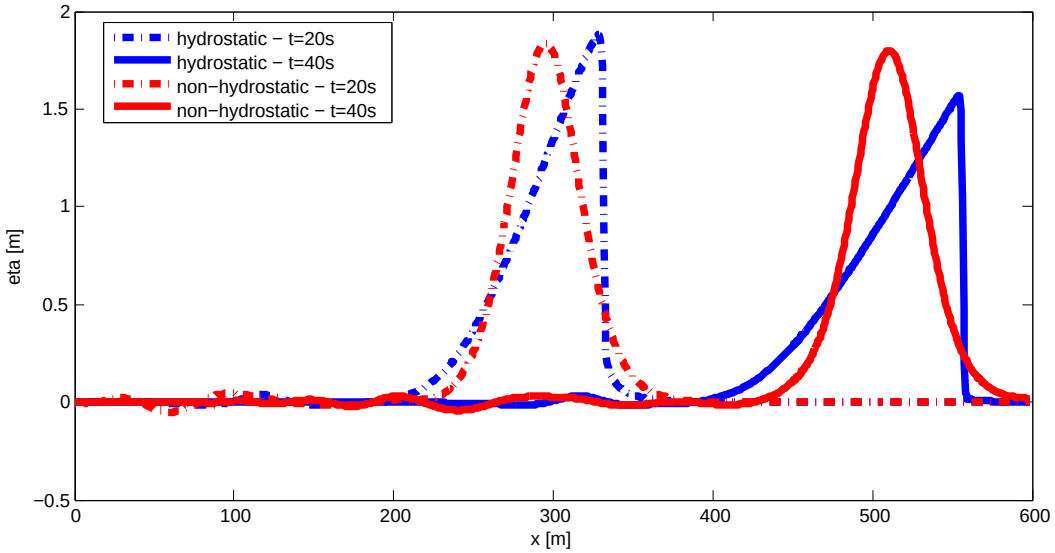

Figure 6: Solitary wave (#cells = 38400): water level at two different points in time.

Wave run-up on a beach

The next experiment is a solitary wave that runs-up on a beach of constant slope. It was conducted as a laboratory experiment at the California Institute of Technology in Pasadena ([13],[14])[2]. Numerical tests of this scenario in a non-hydrostatic model can also be found in [5]. According to [15], the initial elevation of the water surface is set to

$$\eta^0(x,y) = a_s \operatorname{sech}^2\left(\sqrt{\frac{3a_s}{4}}(x - x_s)\right) \tag{18}$$

with initial amplitude $a_s = 0.3m$ and x_s given by:

$$x_s = \frac{d}{\tan(\alpha)} + \sqrt{\frac{4}{3a_s}}\operatorname{arccosh}(\sqrt{20}). \tag{19}$$

Here, $d = 1m$ is the mean water level and $\alpha = tan^{-1}(1/19.85)$ the gradient angle of the beach. As the wave shall approach the coast, the depth-averaged horizontal velocity vector is initialized with

$$\vec{u}^0(x,y) = \begin{pmatrix} -\eta^0\sqrt{g/d} \\ 0 \end{pmatrix}. \tag{20}$$

The initial vertical velocity w at the water surface is set to zero for simplicity. In order to be able to compare the numerical data with the laboratory data, all spatial quantities are rendered dimensionless through a scaling with $1/d$. The time is scaled as well which gives the dimensionless time $\tau = t\sqrt{g/d}$. [15]

[2]The laboratory data is available at: `http://nctr.pmel.noaa.gov/benchmark/Laboratory/Laboratory_CanonicalBathymetry/index.html`.

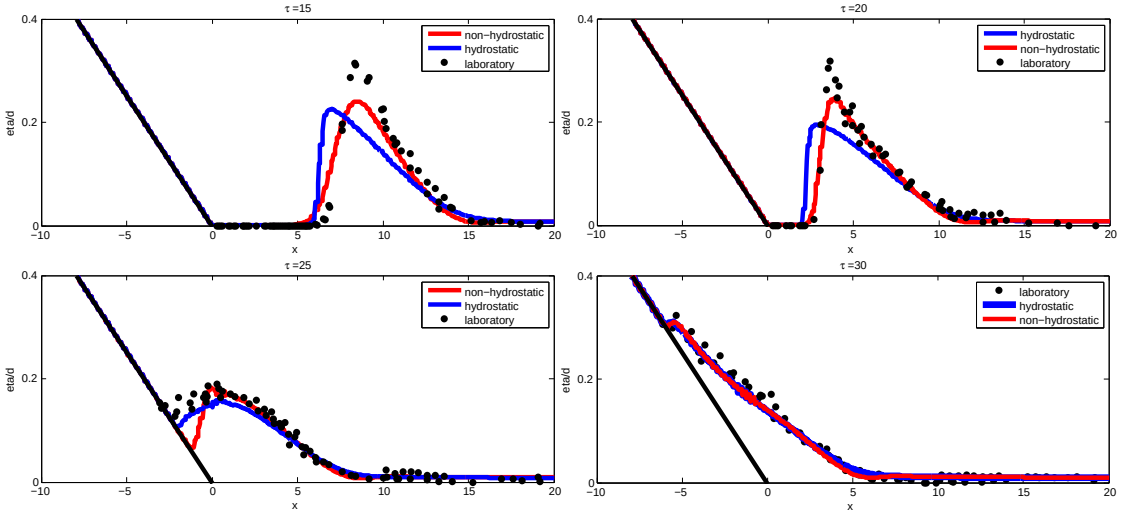

Figure 7: Wave run-up on a beach (#cells = 5120)

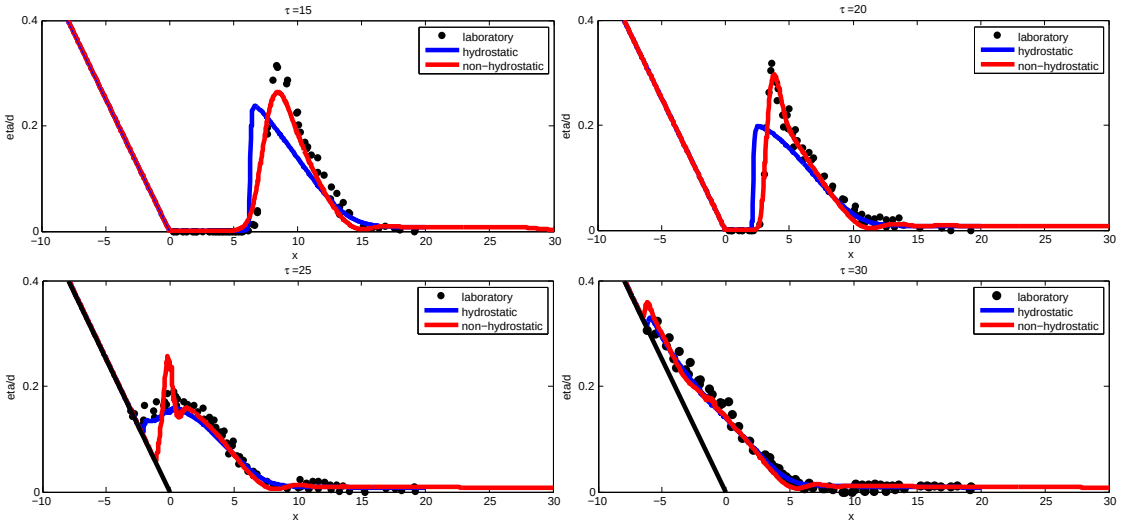

Figure 8: Wave run-up on a beach (#cells = 20480).

As Fig. 7 and Fig. 8 show, there is a significant qualitative improvement of the wave form for the non-hydrostatic model in the plots at $\tau = 15$ and $\tau = 20$. Also note, that increasing the grid resolution improves the approximation of the laboratory data for these points in time.

However, for $\tau = 25$, a disturbance at the wet-/dry-interface occurs and the hydrostatic solution approximates the laboratory data better. This error can also be observed in [5] and [11], although it is much weaker in the plots presented there. As shown in Fig. 7 and Fig. 8, the disturbance increases with higher grid resolution. Indeed, numerical experiments with the model of [11] (SWE-NH) show a similar unphysical peak for fine grids.

The source of this error is presently unknown. Several factors may account for this behaviour. First, it might be the case that one of the assumptions included in our model breaks down near the wet-/dry-interface. For instance, the magnitude of the non-linear terms in the vertical momentum equation of the non-hydrostatic shallow water equations might be in the order of the time-derivative such that a negligence is no longer justified. Second, the wetting and drying algorithm of the hydrostatic solver makes the ramp wet by setting the water level of the respective cells to the dry tolerance. In that way, small discharges develop since the artificially created water runs down the ramp. Coupled with the non-hydrostatic correction, this might lead to an unphysical behaviour. Finally, the negligence of the surface integral in the continuity equation (9) as explained in App. A might be problematic since ∇h might no longer vanish near the wet-/dry-interface.

Wave propagation over a submerged bar

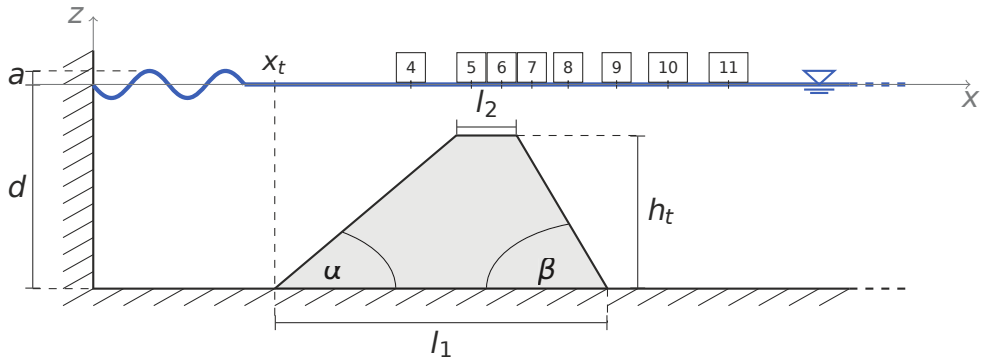

Figure 9: Setup of the submerged bar test case [5].

gauge	[i]	4	5	6	7	8	9	10	11
position	x_i [m]	10.5	12.5	13.5	14.5	15.7	17.3	19.0	21.0

Table 1: Gauges of the submerged bar scenario [5].

As a final validation test case, we conducted the Beji and Battjes experiment [1] which examines the propagation of sinusoidal waves over a submerged trapecoidal bar. At the

left boundary of the channel, periodic waves are generated. Over time, these waves propagate over the bar and a complex wave pattern develops through wave decomposition. This demands a sufficiently accurate dispersive numerical model [1]. Hence, this experiment has established as a standard test case for non-hydrostatic models, see e.g. [4],[5],[12] and [16].

In Fig. 9, the experimental setup is depicted. The sinusoidal waves generated at the left boundary in the channel of depth $d = 0.4m$, width $w = 0.09375m$ and length $l = 64m$ have amplitude $a = 1cm$ and wave period $T = 2.02s$. The obstacle starts at $x_t = 6m$ and has height $h_t = 0.3m$. For the bases, $l1 = 11m$ and $l2 = 2m$ holds. The gradient angle of the ascent (α) is $tan^{-1}(1/20)$ and the one of the descent (β) is given by $tan^{-1}(1/10)$.

We measured the surface elevation at eight different gauges S4-S11 whose positions are given in Tab. 1. The results are plotted in Fig. 10. Unfortunately, the laboratory data of this experiment is currently not available. We therefore compared our graphs to the graphs of the laboratory data as depicted in [5].

Our non-hydrostatic model simulates the decomposition of the wave into two smaller waves correctly. For gauges S4-S8, the correct wave pattern is obtained although the amplitudes are occasionally not high enough. Again, this seems to be a result of the discretization error and can be improved by using a higher-order method and/or a multi-layer model. In the plots of the gauges S9-S11, the same discrepancies (small phase shift) as in [5] can be observed. In total, however, the non-hydrostatic solution approximates the experimental data significantly better than the hydrostatic solution.

4. Conclusion and Outlook

Based on the results of the validation test cases, we may conclude that the non-hydrostatic extension of sam(oa)^2 presented here is a physically sound model for the simulation of weakly-dispersive water waves. Yet, high resolutions near a wet/dry interface seem to cause an unphysical disturbance that has to be investigated further. With the exception of the neglected integral in the continuity equation, the presented dual grid discretization of the non-hydrostatic pressure equation avoids having to evaluate derivatives at discontinuities—a major difficulty in our model. In that way, adaptivity is fully supported. Also, data access during a grid traversal is local (on elements and nodes) which allows for a matrix-free solution of the system of linear equations.

Regarding computational cost, the most expensive step involved in the non-hydrostatic correction is solving the system of linear equations. Future work will therefore have to deal with faster (parallel) solvers and preconditioning.

In addition, further research might investigate how the accuracy of the non-hydrostatic extension can be improved. Besides a higher-order discretization both of the hydrostatic shallow water equations and the non-hydrostatic correction, one might also implement a multi-layer model in order to obtain a better approximation of the vertical velocity.

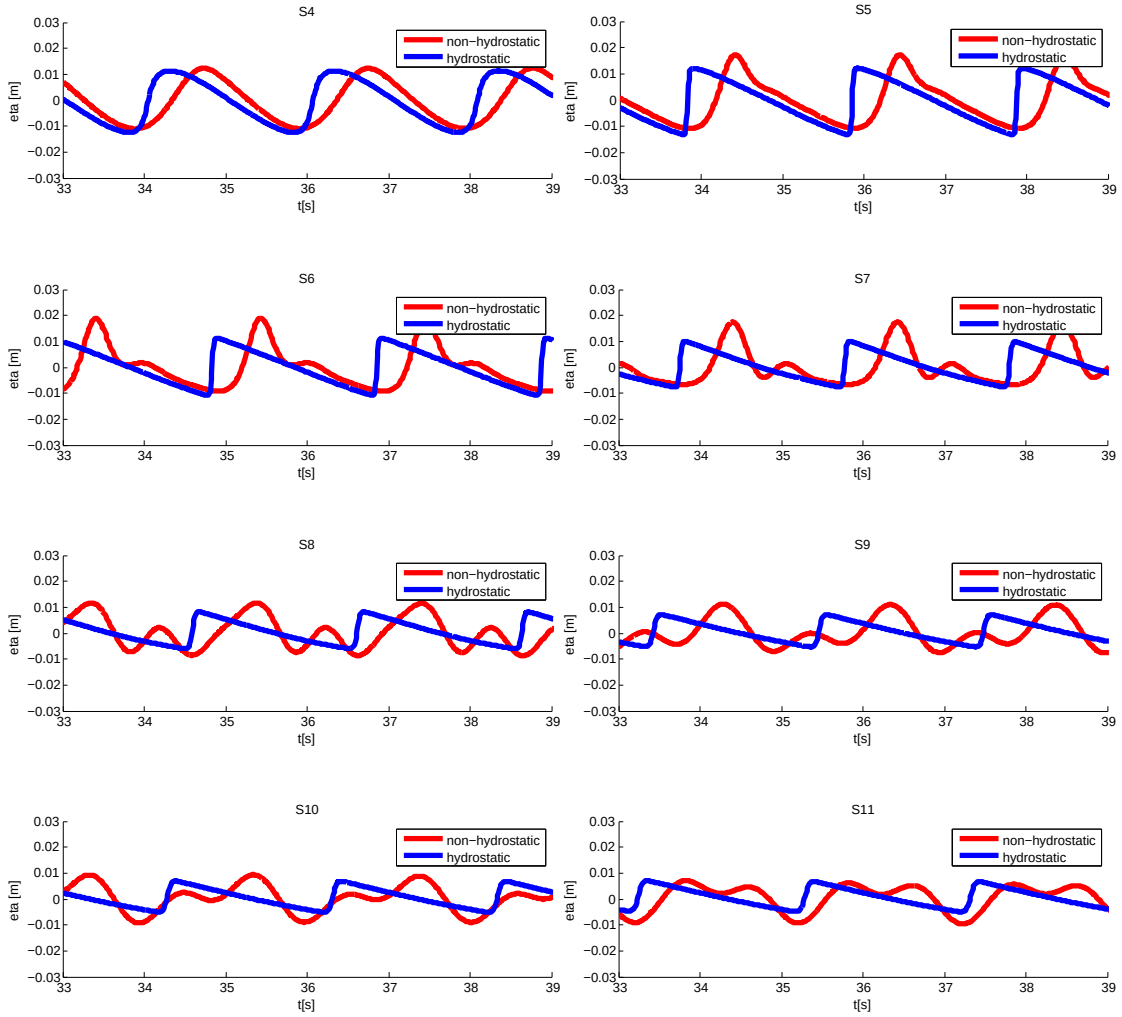

Figure 10: Wave propagation over a submerged bar (#cells = 196608): water elevation over time at the gauges S4–S11.

A. Derivation of the integrated continuity equation

We integrate the first term in (5) over the control volume of a dual grid cell:

$$\begin{aligned}\int_{V_d} \frac{\partial u}{\partial x} dV_d &= \int_{\Omega_d} \int_{b(x,y)}^{\eta(x,y)} \frac{\partial u}{\partial x} dz d\Omega_d \\ &= \int_{\Omega_d} \frac{\partial hU}{\partial x} d\Omega_d + \int_{\Omega_d} u_{z=b} \frac{\partial b}{\partial x} - u_{z=\eta} \frac{\partial \eta}{\partial x} d\Omega_d \\ &= \int_{\Gamma_d} n_x hU d\Gamma_d + \int_{\Omega_d} u_{z=b} \frac{\partial b}{\partial x} - u_{z=\eta} \frac{\partial \eta}{\partial x} d\Omega_d.\end{aligned}$$

Here, n_x denotes the x-component of the outward pointing normal unit vector. In the integration, we have used the Leibniz rule for parameter integrals and the Gaussian divergence theorem.

A problem occurs in the evaluation of the surface integral: since bathymetry and water elevation are constant in a primary cell, the integrand is only non-zero at the boundary between two primary cells. However, at this boundary, there is a discontinuity of b and η due to the constant basis functions. In the conducted validation test cases, there are no steep bathymetry gradients. We therefore neglected the surface integral for simplicity. For non-artificial and realistic tsunami scenarios where abrupt changes in the bathymetry will likely occur, one might have to approximate $u_{z=b}$ and $u_{z=\eta}$ with the depth-averaged velocity U, neglect the derivatives of η and precompute the derivatives of b on the cell edges instead.

While the same integration as above is carried out for $\int_{V_d} \frac{\partial v}{\partial y} dV_d$, no difficulties arise in the integration of the vertical velocity:

$$\begin{aligned}\int_{V_d} \frac{\partial w}{\partial z} dV_d &= \int_{\Omega_d} \int_{b(x,y)}^{\eta(x,y)} \frac{\partial w}{\partial z} dz d\Omega_d \\ &= \int_{\Omega_d} w_{z=\eta} d\Omega_d.\end{aligned}$$

B. Element matrix and right-hand side

We consider an arbitrary primary element e for which we want to compute the element matrix of dual cell equation contributions. In sam(oa)^2, we have access to a 2×2 rotation matrix $J = (j_{i,k})_{1<=i,k<=2}$ that transforms the standard reference element into the element e.

In the standard reference element, let n_h^+ denote the outward pointing normal vector on the horizontal boundary of the blue dual cell contribution and n_v^+ the one on the vertical boundary, respectively. After applying the rotation matrix J on these vectors, we get the rotated normal vectors that are subsequently represented with a tilde (e.g. $\tilde{n}_v^+$). Further, we denote with $\tilde{n}_v^-$ and $\tilde{n}_h^-$ the rotated outward pointing normal vectors for the green dual cell equation contributions and with an additional subscript the components of the normal vectors (e.g. $\tilde{n}_{v,x}^-$ for the x-component of $\tilde{n}_v^-$). If we use the numbering of the

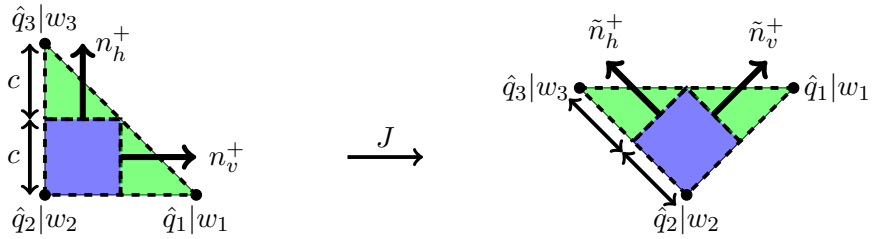

Figure 11: Rotation of the standard reference element.

standard reference element and evaluate equation (11) for each of the three contributions using the rotated normal vectors and the rotated non-hydrostatic pressure gradient, we get a system of linear equations $Ax = b$ for the element e. A is given by

$$
\begin{aligned}
a_{1,1} &= -\frac{h^{(n)}}{4}\left(\tilde{n}^-_{v,x} j_{1,1} + \tilde{n}^-_{v,y} j_{2,1}\right) + \frac{c^2}{h^{(n+1)}} \\
a_{1,2} &= \frac{h^{(n)}}{4}\left(\tilde{n}^-_{v,x}(j_{1,1} + j_{1,2}) + \tilde{n}^-_{v,y}(j_{2,1} + j_{2,2})\right) \\
a_{1,3} &= -\frac{h^{(n)}}{4}\left(\tilde{n}^-_{v,x} j_{1,2} + \tilde{n}^-_{v,y} j_{2,2}\right)
\end{aligned}
$$

$$
\begin{aligned}
a_{2,1} &= -\frac{h^{(n)}}{4}\left(\tilde{n}^+_{h,x} j_{1,1} + \tilde{n}^+_{h,y} j_{2,1} + \tilde{n}^+_{v,x} j_{1,1} + \tilde{n}^+_{v,y} j_{2,1}\right) \\
a_{2,2} &= \frac{h^{(n)}}{4}\left(\tilde{n}^+_{h,x}(j_{1,1} + j_{1,2}) + \tilde{n}^+_{h,y}(j_{2,1} + j_{2,2}) + \tilde{n}^+_{v,x}(j_{1,1} + j_{1,2}) + \tilde{n}^+_{v,y}(j_{2,1} + j_{2,2})\right) + \frac{2c^2}{h^{(n+1)}} \\
a_{2,3} &= -\frac{h^{(n)}}{4}\left(\tilde{n}^+_{h,x} j_{1,2} + \tilde{n}^+_{h,y} j_{2,2} + \tilde{n}^+_{v,x} j_{1,2} + \tilde{n}^+_{v,y} j_{2,2}\right)
\end{aligned}
$$

$$
\begin{aligned}
a_{3,1} &= -\frac{h^{(n)}}{4}\left(\tilde{n}^-_{h,x} j_{1,1} + \tilde{n}^-_{h,y} j_{2,1}\right) \\
a_{3,2} &= \frac{h^{(n)}}{4}\left(\tilde{n}^-_{h,x}(j_{1,1} + j_{1,2}) + \tilde{n}^-_{h,y}(j_{2,1} + j_{2,2})\right) \\
a_{3,3} &= -\frac{h^{(n)}}{4}\left(\tilde{n}^-_{h,x} j_{1,2} + \tilde{n}^-_{h,y} j_{2,2}\right) + \frac{c^2}{h^{(n+1)}}.
\end{aligned}
$$

and the vector b can be written as:

$$
b = \frac{1}{\Delta t}\begin{pmatrix}
-c^2\left(\frac{\widetilde{w_1}^{(n+1)}}{2}\right) - c\tilde{n}^-_{v,x}\widetilde{hU}^{(n+1)} - c\tilde{n}^-_{v,y}\widetilde{hV}^{(n+1)} \\
-c^2\widetilde{w_2}^{(n+1)} - c\widetilde{hU}^{(n+1)}(\tilde{n}^+_{h,x} + \tilde{n}^+_{v,x}) - c\widetilde{hV}^{(n+1)}(\tilde{n}^+_{h,y} + \tilde{n}^+_{v,y}) \\
-c^2\left(\frac{\widetilde{w_3}^{(n+1)}}{2}\right) - c\tilde{n}^-_{h,x}\widetilde{hU}^{(n+1)} - c\tilde{n}^-_{h,y}\widetilde{hV}^{(n+1)}
\end{pmatrix}.
$$

References

[1] S Beji and JA Battjes. Experimental investigation of wave propagation over a bar. *Coastal Engineering*, 19(1):151–162, 1993.

[2] V. Casulli and G. Stelling. Numerical simulation of 3d quasi-hydrostatic, free-surface flows. *Journal of Hydraulic Engineering*, 124(7):678–686, 1998.

[3] Alexandre Joel Chorin. Numerical solution of the navier-stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.

[4] Haiyang Cui. *A New Numerical Model for Simulating the Propagation of and Inundation by Tsunami Waves*. VSSD, 2013.

[5] Annika Fuchs. *Effiziente parallele Verfahren zur Lösung verteilter, dünnbesetzer Gleichungssysteme eines nichthydrostatischen Tsunamimodells*. PhD thesis, Staats- und Universitätsbibliothek Bremen, 2013.

[6] David L. George. Augmented riemann solvers for the shallow water equations over variable topography with steady states and inundation. *J. Comput. Phys.*, 227(6):3089–3113, March 2008.

[7] Juan Horrillo, Zygmunt Kowalik, and Yoshinori Shigihara. Wave dispersion study in the indian ocean-tsunami of december 26, 2004. *Marine Geodesy*, 29(3):149–166, 2006.

[8] R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.

[9] Boris Levin and Mikhail Nosov. *Physics of tsunamis*. Springer, 2008.

[10] Oliver Meister, Kaveh Rahnema, and Michael Bader. A software concept for cache-efficient simulation on dynamically adaptive structured triangular grids. In *PARCO*, pages 251–260, 2011.

[11] Philipp Johannes Samfaß. Extension of the finite volume solver swe towards the non-hydrostatic shallow water equations. Bachelor's thesis, Institut für Informatik, Technische Universität München, September 2014.

[12] Guus Stelling and Marcel Zijlema. An accurate and efficient finite-difference algorithm for non-hydrostatic free-surface flow with application to wave propagation. *International Journal for Numerical Methods in Fluids*, 43(1):1–23, 2003.

[13] Constantine Emmanuel Synolakis. *The runup of long waves*. PhD thesis, California Institute of Technology, 1986.

[14] Costas Emmanuel Synolakis. The runup of solitary waves. *Journal of Fluid Mechanics*, 185:523–545, 12 1987.

[15] Vasily Vladimirovitch Titov and Costas Emmanuel Synolakis. Modeling of breaking and nonbreaking long-wave evolution and runup using vtcs-2. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 121(6):308–316, 1995.

[16] Roy A. Walters. A semi-implicit finite element model for non-hydrostatic (dispersive) surface waves. *International Journal for Numerical Methods in Fluids*, 49(7):721–737, 2005.

[17] H Weilbeer and JA Jankowski. A three-dimensional non-hydrostatic model for free surface flows–development, verification and limitations. In *Proc. Of 6th International Conference Estuarine and Coastal Modeling*, pages 162–177, 2000.