BACHELOR

Finite volume integration of the shallow water equations

On the numerical integration of the shallow water equations in a two-dimensional case of bottom deformation

Herps, Tristan

Award date:
2021

Technische Universiteit
Eindhoven
University of Technology

Department of Mathematics

Postbus 513, 5600 MB Eindhoven
The Netherlands
www.tue.nl

Author
Tristan Herps (1228916)

Responsible Lecturer
prof. dr. ir. Barry Koren

Date
January 28, 2021

Finite volume integration of the shallow water equations

On the numerical integration of the shallow water equations in a two-dimensional case of bottom deformation

Bachelor final project

Tristan Herps (1228916)
t.herps@student.tue.nl

Technische Universiteit
Eindhoven
University of Technology

Table of contents

Title
Finite volume integration of the shallow water equations

This report contains a study on the shallow water equations. In particular, this study implements the one-dimensional shallow water equations on a body of water on which a deformation in the bottom is incurred, instantaneous or over time. The implementation is done in MATLAB. The numerical integration method used for the shallow water equations is the finite volume integration method. This method consists of 3 main elements. Firstly, it includes cell face approximation schemes: second order central, first order upwind, unlimited second order upwind, limited second order upwind. Limited second order upwind is implemented with 3 different limiter functions: Minmod, Superbee, BK. Secondly, it includes flux approximation methods: direct flux evaluation and flux difference splitting. Lastly, it includes time integration: the RK1 and RK3b schemes. The proposed methods are tested and compared using the test case. This is a 100 metres wide tank filled with 50 metres of water, for which it is assumed its bottom can deform in any form at any time. The best results for the water surface in this problem are obtained by using second order upwind with flux difference splitting and RK3b, either limited or unlimited.

1 Introduction

Fluid mechanics is a field of study in physics concerning the mechanical behavior of gases, liquids and plasmas, collectively called fluids. In fluid mechanics, it is of great interest to describe this behavior mathematically, as this gives rise to the possibility to model and simulate this particular behavior. As a result, fluid mechanics plays an essential role in various branches in science and engineering. As fluids are ubiquitous, one can already sense that a quantitatively large report can be written on just how big a footprint fluid mechanics sets on various science and engineering disciplines. As one (of many) examples, think of meteorology, the field of study concerning the weather and, ultimately a more relevant field of interest greatly affecting our day-to-day lives, weather predictions. Namely, with help of the laws of fluid mechanics, meteorologists can predict the state of our atmosphere in the near future in terms of pressure, humidity, temperature etc. As a consequence, we now have many online tools helping us in deciding whether to bring an umbrella when travelling outdoors.

One subdiscipline of fluid mechanics is fluid dynamics. In fluid dynamics, the behavior of fluid flow is studied, i.e. liquids and gases in motion, as opposed to the other subdiscipline of fluid mechanics, fluid statics. As the name might already suggest, fluid statics concerns the behavior of fluids at rest. In this paper, we will mostly be considering the discipline of fluid dynamics. In fact, this paper will concern specific situations with fluid dynamics. The situations we will deal with are bodies of water with the important characteristic that the horizontal length scales are significantly larger than the vertical length scale. A simple example would be a river, which is much longer and wider than it is deep. What this paper will derive in upcoming chapters, is that this basically means that the horizontal velocity of the water in these situations such as the river is independent of the depth of the water. With this result, the main subject of this paper is born: the shallow water equations (SWE).

The shallow water equations are a set of equations derived from the famous Navier-Stokes equations by the well known mathematician Saint-Venant in 1871 [1]. Famous, as in finding a general analytical solution for the Navier-Stokes equations is one of the Millennium Problems [2] which still has not been solved yet. An analytical solution to the Navier-Stokes equations is only known for a few specific cases. The same situation applies for the shallow water equations.

This paper explores the possibility to use the shallow water equations on a body of water for which it is assumed that the bottom incurs a deformation which occurs either instantaneous or over time. We start with a derivation of both the one-dimensional and two-dimensional form of the shallow water equations. This is followed by presenting and motivating the choice of numerical integration method; finite volume integration. Various numerical schemes are presented for finite volume integration. These include cell face solution approximation methods, together with finding the corresponding flux across each cell face. Above that, two suitable time integration methods are presented. These methods are subsequently applied on the one-dimensional shallow water equations and implemented in MATLAB to simulate the water surface on a tank of water from the side undergoing bottom deformation. Using this test problem, the methods are compared.

2 Theoretical background

In this chapter the relevant mathematical theory concerning the SWE will be discussed. The SWE are a set of partial differential equations which originates from two fundamental laws in physics: the mass conservation law and the momentum conservation law. The aforementioned laws will be discussed firstly, followed by a section presenting the derivation of the SWE from the laws of fluid dynamics.

2.1 The laws of conservation of mass and momentum

For the remainder of this report, we assume that we are dealing with fluids for which we can neglect the effects of viscosity and compressibility. As a consequence of this neglecting of shear stresses on the fluid, each fluid particle experiences pressure forces only. We denote the pressure by p. We can use Newton's law of conservation of momentum to describe the motion of each fluid parcel in our fluid. To properly understand the use of Newton's law, consider the following simplified visualization of an element of the fluid:

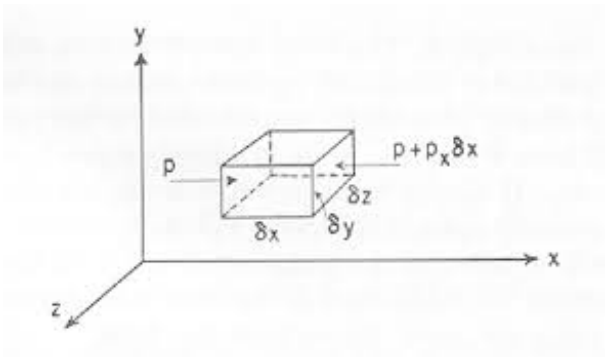

Figure 2.1: Pressure components on a rectangular fluid element as described by Stoker [3], where p represents the pressure

Newton's second law of motion defines an equation between the net force applied on a body and the change in momentum. In the case of our rectangular fluid element in figure 2.1, the equation for the x-direction becomes:

$$\mathrm{F}_{(x)} = m\mathbf{a}_{(x)}, \tag{2.1}$$

$$[-(p + p_x\delta x) + p]\delta y\delta z + X\rho\delta x\delta y\delta z = \rho\delta x\delta y\delta z\mathbf{a}_{(x)}. \tag{2.2}$$

Here, X represents the x directed component of the gravitational acceleration, ρ represents the density of the fluid, which is constant as, by assumption, the fluid is incompressible. And lastly, $\mathbf{a}_{(x)}$ is the x directed component of the acceleration of the fluid element. Important to note is that here, p is a field variable, i.e. $p = p(x, y, z, t)$. Hence, in figure 2.1, the pressure on the faces of the fluid element which are perpendicular to the x axis will generally not be the same. To account for this, the pressure working in the negative x direction is represented as a first order Taylor polynomial. In this report, we will use letter subscripts to represent differentiation with respect to the letter used in the subscript, unless explicitly stated elsewhere. Dividing both sides of equation (2.2) by $\delta x\delta y\delta z$ and subsequently taking the limit where $\delta x, \delta y$ and δz all approach 0 yields the equation of motion for the x-direction:

$$-\frac{1}{\rho}p_x + X = \mathbf{a}_{(x)}. \tag{2.3}$$

The equations of motion for the y and z direction can be found analogously. This yields a set of equations of motion which can be represented in vector form:

$$-\frac{1}{\rho}\text{grad } p + \mathbf{F} = \mathbf{a}, \tag{2.4}$$

where $\mathbf{a} = (\mathbf{a}_{(x)}, \mathbf{a}_{(y)}, \mathbf{a}_{(z)})$ and the body force $\mathbf{F} = (X, Y, Z) = (0, -g, 0)$, where g is the acceleration of gravity, as we set the positive y-axis to be vertically upward.

The form of the differential equations for motion as given in (2.4) is the so-called Lagrangian form. The Lagrangian form focuses on the motion of each individual fluid parcel in the fluid volume over time. For the sake of deriving the SWE, it is more useful to consider a different form of the equations of motion, which is referred to as the Eulerian form. The Eulerian form is focused on determining the velocity distribution over time at fixed points in a spatial domain which is occupied by the fluid.

In the system as depicted in figure 2.1 we introduce the velocity field, with components u, v and w, in terms of the space variables and time. To obtain the three equations of motion when expressing (2.4) componentwise in the Eulerian variables u, v and w, we have to calculate the time derivatives of functions describing the motion of a given fluid particle. In order to determine the vector components of $\mathbf{a}$ in (2.4), we start by introducing an arbitrary function $G(x, y, z, t)$ for a certain particle in the fluid which follows the path given by

$$\mathbf{x} = (x(t), y(t), z(t)). \tag{2.5}$$

Using (2.5), we can obtain the velocity vector $\mathbf{v}$:

$$\mathbf{v} = (\dot{x}(t), \dot{y}(t), \dot{z}(t)) = (u, v, w). \tag{2.6}$$

Then the particle/material derivative becomes:

$$\frac{\text{D}G}{\text{D}t} = G_t + \mathbf{v}\text{grad } G, \tag{2.7}$$

$$\frac{\text{D}G}{\text{D}t} = G_t + uG_x + vG_y + wG_z. \tag{2.8}$$

Note that $G_t = \frac{\partial G}{\partial t} \neq \frac{\text{D}G}{\text{D}t}$. The acceleration vector $\mathbf{a}$ as seen in (2.4) consequently becomes $\mathbf{a} = (\frac{\text{D}u}{\text{D}t}, \frac{\text{D}v}{\text{D}t}, \frac{\text{D}w}{\text{D}t})$. Starting with the x-direction and following an analogous computation as seen above, we know for $\mathbf{a}_{(x)} = \frac{\text{D}u}{\text{D}t}$:

$$\mathbf{a}_{(x)} = u_t + uu_x + vu_y + wu_z. \tag{2.9}$$

Performing analogous computations for y and z, the equations of motion (2.4) in the desired Eulerian form become:

$$u_t + uu_x + vu_y + wu_z = -\frac{1}{\rho}p_x, \tag{2.10}$$

$$v_t + uv_x + vv_y + wv_z = -\frac{1}{\rho}p_y - g, \tag{2.11}$$

$$w_t + uw_x + vw_y + ww_z = -\frac{1}{\rho}p_z. \tag{2.12}$$

As the fluid is incompressible by assumption, ρ is a given constant. Hence, the set of the three partial differential equations above are expressed in four unknown quantities. Using the incompressibility of the fluid, the necessary fourth equation for the system, which implements the law of conservation of mass, can be added relatively easily. In order to derive the mass conservation law, consider an arbitrary

region R in $\mathbb{R}^3$ in which no liquid is destroyed or created and is enclosed by a fixed closed surface S. The derivation starts using the relation

$$\iint_S \rho v_{(n)} dS = 0. \tag{2.13}$$

Here, $v_{(n)}$ represents the positive velocity component directed in the direction of the outward normal vector of S. The above relation indicates that the mass flux travelling outward through our surface S is equal to 0. Applying the divergence theorem of Gauss, we can express our surface integral as

$$\iint_S \rho v_{(n)}\, dS = \iiint_R \text{div } (\rho \mathbf{v}) d\tau. \tag{2.14}$$

This leads to

$$\iiint_R \text{div } (\rho \mathbf{v}) d\tau = 0, \tag{2.15}$$

where ρ is assumed to be constant. This implies that

$$\text{div } \mathbf{v} = u_x + v_y + w_z = 0. \tag{2.16}$$

Equation (2.16) is also known as the continuity equation. As soon as appropriate initial and boundary conditions are introduced, equations (2.10)-(2.12) and (2.16) form a system in which the velocity components u, v, w and the pressure p can be determined. The introduction of the initial and boundary conditions will be presented below.

2.2 Initial and boundary conditions

Consider the following visual representation of flow in a channel:

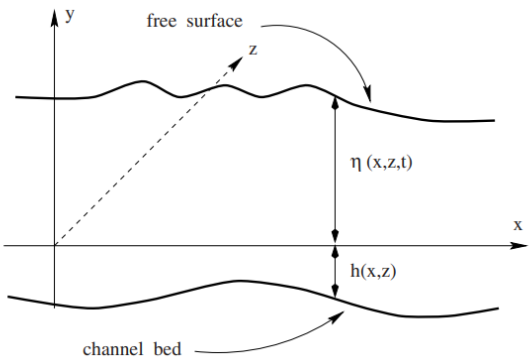

Figure 2.2: Flow in a channel with a free surface under gravity as described by Toro [4]. Note that the bottom in this scenario is considered to be fixed in time, i.e. h does not depend on t.

In figure 2.2, there are two important boundaries to be considered in the domain. Firstly, we consider the free surface elevation, which is denoted by $y = \eta(x, z, t)$. Secondly, the bottom of the channel is to be considered, denoted by $y = -h(x, z, t)$. It is assumed that the bottom is not necessarily fixed in time. Ergo, h is a function of t too. For a good reason to consider a bottom which is time dependent, one could think of seaquakes. It is also possible to work with a time-independent bottom, which is merely a simplified scenario of considering a time-dependent bottom. Having introduced these notions, the following boundary conditions are imposed:

$$\eta_t + u\eta_x - v + w\eta_z = 0 \text{ at } y = \eta, \tag{2.17}$$

$$p = 0 \text{ at } y = \eta, \tag{2.18}$$

$$h_t + uh_x + v + wh_z = 0 \text{ at } y = -h. \tag{2.19}$$

For a justification of the aforementioned boundary conditions, see section 1.4 in the book on water waves by Stoker [3]. Next to these boundary conditions, $h(x, z, t)$ and $\eta(x, z, 0)$ are given. All together, the system formed by equations (2.10)-(2.12) and (2.16) is now consequently a well-defined problem, referred to as the free-surface gravity problem, which should be solvable. However, difficulties have shown to arise trying to solve this problem numerically. Above that, solving this problem analytically is merely possible in specific simple cases of the free surface gravity problem [4].

2.3 The two-dimensional shallow water equations for a time-dependent bottom

Consider again the flow of water in a channel with a free surface, i.e. the free-surface gravity problem. The shallow water equations are actually an approximation to this problem, and arise when making the assumption that the vertical component of the acceleration is negligible. Water is considered shallow when the water height is much smaller than the wavelength. Mathematically, we can set $\frac{\mathrm{D}v}{\mathrm{D}t} = 0$ now. Note that in the case of a time-dependent bottom, the validity of this equation holds if the bottom amplitude and the rate of change in vertical velocity are not too big. Substituting this into equations (2.11) yields

$$-\frac{1}{\rho}p_y - g = 0, \tag{2.20}$$

$$p = -\rho g y + C, \tag{2.21}$$

where C is the constant of integration. By our boundary condition (2.18), we know that at $y = \eta$:

$$p = \rho g(\eta - y). \tag{2.22}$$

Equation (2.22) is commonly referred to as the hydrostatic pressure relation. If (2.22) is differentiated with respect to x and z, we obtain

$$p_x = \rho g \eta_x, \tag{2.23}$$

$$p_z = \rho g \eta_z. \tag{2.24}$$

Important to note is that, next to u, both p_x and p_z are independent of y and therefore equations (2.10) and (2.12) are independent of y. For the sake of convenience, the resulting equations of the free-surface gravity problem are written below:

$$u_t + uu_x + wu_z = -g\eta_x, \tag{2.25}$$

$$0 = -\frac{1}{\rho}p_y - g, \tag{2.26}$$

$$w_t + uw_x + ww_z = -g\eta_z, \tag{2.27}$$

$$u_x + v_y + w_z = 0. \tag{2.28}$$

Next, the continuity equation (2.28) will be integrated with respect to y, which yields

$$\int_{-h}^{\eta} u_x dy + \int_{-h}^{\eta} w_z dy + v|_{-h}^{\eta} = 0. \tag{2.29}$$

Then, using boundary conditions (2.17) and (2.19) by summing them together with (2.29), we obtain after simplifications:

$$(\eta + h)_t + \frac{\partial}{\partial x}\int_{-h}^{\eta} u dy + \frac{\partial}{\partial z}\int_{-h}^{\eta} w dy = 0. \tag{2.30}$$

Notice that u and w are independent of y, which means that (2.30) can be simplified to

$$(\eta + h)_t + (u(\eta + h))_x + (w(\eta + h))_z = 0. \tag{2.31}$$

Next, to bring the equations more in divergence form, multiply (2.25) by $(\eta + h)$ and (2.31) by u and add the two results, which yields

$$(u(\eta + h))_t + (u^2(\eta + h))_x + (uw(\eta + h))_z = -g(\eta + h)\eta_x. \tag{2.32}$$

Analogous to above but using (2.27) and (2.31) respectively, we obtain

$$(w(\eta + h))_t + (uw(\eta + h))_x + (w^2(\eta + h))_z = -g(\eta + h)\eta_z. \tag{2.33}$$

The right hand side of equation (2.32) can be expressed as

$$-g(\eta + h)\eta_x = g(\eta + h)h_x - \frac{1}{2}g((\eta + h)^2)_x. \tag{2.34}$$

For notation's sake, denote with $H = \eta(x, z, t) + h(x, z, t)$, i.e. the total water height. Then, by using (2.34) and our notation, (2.32) can be rewritten as

$$(Hu)_t + (Hu^2 + \frac{1}{2}gH^2)_x + (Huw)_z = gHh_x. \tag{2.35}$$

Analogous to the computation for (2.35), we arrive at the following for (2.33):

$$(Hw)_t + (Huw)_x + (Hw^2 + \frac{1}{2}gH^2)_z = gHh_z. \tag{2.36}$$

The final step is to properly write down the shallow water equations in the two-dimensional case. In order to express equations (2.31), (2.35) and (2.36), we introduce the following vectors:

$$\mathbf{u} = \begin{pmatrix} H \\ Hu \\ Hw \end{pmatrix}, \tag{2.37}$$

$$\mathbf{f} = \begin{pmatrix} Hu \\ Hu^2 + \frac{1}{2}gH^2 \\ Huw \end{pmatrix}, \tag{2.38}$$

$$\mathbf{g} = \begin{pmatrix} Hw \\ Huw \\ Hw^2 + \frac{1}{2}gH^2 \end{pmatrix}, \tag{2.39}$$

$$\mathbf{s} = \begin{pmatrix} 0 \\ gHh_x \\ gHh_z \end{pmatrix}. \tag{2.40}$$

Then the conservative form of the two-dimensional shallow water equations is

$$\mathbf{u}_t + \mathbf{f}_x + \mathbf{g}_z = \mathbf{s}. \tag{2.41}$$

2.4 The two dimensional shallow water equations for a stationary bottom

In this section we consider again the free-surface gravity problem as described in section 2.2, but with one alteration. We now assume that the bottom is time-independent, i.e. the bottom of the channel is merely given by $y = -h(x, z)$. The time-independency of the bottom results in a subtle simplification

in the boundary condition concerning the bottom of the channel due to the fact that now $h_t = 0$. The full slip boundary condition concerning the bottom of the channel is equation (2.19), which now becomes:

$$uh_x + v + wh_z = 0 \text{ at } y = -h. \tag{2.42}$$

Following an mostly analogous path as for the derivation of the two-dimensional SWE for a time-dependent bottom, one will find the same system of equations as seen in (2.41) with the same vectors (2.37)-(2.40). The sole difference lies in deriving equation (2.31) from (2.30). In the time-independent bottom case, the first term of (2.30) would be η_t. By observing that in this case $h_t = 0$, we can again obtain (2.31) by simply adding h_t to (2.30). It is important to note that in the derivative of quantity H with respect to time, H_t, in the case of a time-dependent bottom, the term h_t is not necessarily equal to 0, as opposed to the case of a time-independent bottom.

2.5 One dimensional shallow water equations for a time-dependent bottom

Before continuing to presenting several numerical methods for the SWE, it is essential to present the one-dimensional SWE beforehand, as the numerical methods to be discussed concern the one-dimensional SWE. Consider now, instead of the three dimensional situation of the free surface gravity problem, a two-situation. Here, the z coordinate is omitted, which means only the x and y coordinates are to be considered. Following analogous computations as for the derivation of the two-dimensional SWE, we arrive at the one-dimensional version in conservative form:

$$H_t + (Hu)_x = 0, \tag{2.43}$$

$$(Hu)_t + (Hu^2 + \frac{1}{2}gH^2)_x = gHh_x, \tag{2.44}$$

where $H = H(x,t) = h(x,t) + \eta(x,t)$, $u = u(x,t)$ and $h = h(x,t)$. Setting

$$\mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \begin{pmatrix} gH \\ gHu \end{pmatrix}, \tag{2.45}$$

$$\mathbf{f(q)} = \begin{pmatrix} q_2 \\ \frac{q_2^2}{q_1} + \frac{1}{2}q_1^2 \end{pmatrix}, \tag{2.46}$$

$$\mathbf{s} = \begin{pmatrix} 0 \\ gq_1h_x \end{pmatrix}. \tag{2.47}$$

yields the vector equation for equations (2.43) and (2.44):

$$\mathbf{q}_t + (\mathbf{f(q)})_x = \mathbf{s}. \tag{2.48}$$

Important to note is that the subscripts in q_1, q_2 in this case do not represent partial derivatives, but are the components in vector $\mathbf{q}$.

3 Numerical integration for one-dimensional SWE

This chapter is devoted to presenting an approach to numerically integrate the one-dimensional shallow water equations, which is called finite volume integration (FVI). Finite volume integration is a method to represent and evaluate partial differential equations, and hence applicable to the SWE. The motivation for implementing FVI on the SWE will become clear later on in this chapter. The necessary steps to be taken in preparation of using FVI on the one-dimensional SWE are discussed here.

3.1 Dealing with discontinuities

Finite difference methods for solving differential equations rely on the approximation of derivatives with finite differences. In the previous section, we derived the one-dimensional shallow water equations in conservative form ((2.48)). Equation (2.48) can be rewritten in quasilinear form as follows:

$$\mathbf{q}_t + A\mathbf{q}_x = \begin{pmatrix} 0 \\ gq_1h_x \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 1 \\ -\frac{q_2^2}{q_1^2} + q_1 & \frac{2q_2}{q_1} \end{pmatrix}. \tag{3.1}$$

Note that A is the Jacobian matrix of $\mathbf{F}$ in (2.48). This matrix can be diagonalized as $A = R\Lambda R^{-1}$, with

$$\Lambda = \begin{pmatrix} u - \sqrt{gH} & 0 \\ 0 & u + \sqrt{gH} \end{pmatrix}, R = \begin{pmatrix} \frac{1}{u-\sqrt{gH}} & \frac{1}{u+\sqrt{gH}} \\ 1 & 1 \end{pmatrix}, \; R^{-1} = \begin{pmatrix} \frac{-gH+u^2}{2\sqrt{gH}} & \frac{1}{2} - \frac{u}{2\sqrt{gH}} \\ \frac{gH-u^2}{2\sqrt{gH}} & \frac{1}{2} + \frac{u}{2\sqrt{gH}} \end{pmatrix}. \tag{3.2}$$

The matrix Λ contains the eigenvalues of matrix A on its diagonal entries. R is the matrix with the eigenvectors of A as columns, R^{-1} represents the inverse of R. The eigenvalues of A represent the wave speed. If no background flow is present, the wave speed equals $\sqrt{gH}$. This means that for the shallow water case, the wave speed is dependent on the total height of the wave, resulting in a higher propagation speed in the top of a wave compared to its front. Therefore, the top catches up with its front, which results in a scenario as below:

Figure 3.1: The top of a wave catches up with its bottom in shallow water, creating a steeper wave

The process of becoming a steeper wave eventually tends to a wave with a breaking front. Since the finite difference methods rely on the local derivatives in the problem, they are not the optimal candidates for numerically solving our problem in question.

3.2 Finite volume integration

Instead, we turn to the finite volume integration method. This section will shed some light on the reason why the FVI method handles discontinuities in the problem better than finite difference methods.

3.2.1 Finite volume equation

Firstly, we need the one-dimensional integral form of the one-dimensional SWE, which can be derived easily from its conservative form (2.48). Integrating (2.48) over an arbitrary, finite interval $[a, b]$ yields

$$\int_a^b (\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x - \mathbf{s})dx = \mathbf{0}, \tag{3.3}$$

which can be simplified to

$$\int_a^b \mathbf{q}_t dx + [\mathbf{f}(\mathbf{q})]_a^b - \int_a^b \mathbf{s}dx = \mathbf{0}. \tag{3.4}$$

The computational domain is partitioned into N cells of equal distances with N+1 cell faces. We denote the i-th cell by Ω_i. Above that, we assume that the solution $\mathbf{q}$ is uniformly constant per cell in both its components q_1, q_2. We will denote the value of $\mathbf{q}$ in cell Ω_i by $\mathbf{q}_i$, and its corresponding component values by $q_{1,i}, q_{2,i}$. Furthermore, the value of $\mathbf{q}$ at the right face of cell Ω_i will be denoted by $\mathbf{q}_{i+\frac{1}{2}}$. The fluxes $\mathbf{F}$ are consequently given by

$$\mathbf{f}(\mathbf{q}_{i+\frac{1}{2}}) = \begin{pmatrix} q_{2,i+\frac{1}{2}} \\ \frac{q_{2,i+\frac{1}{2}}^2}{q_{1,i+\frac{1}{2}}} + \frac{1}{2}q_{1,i+\frac{1}{2}}^2 \end{pmatrix}. \tag{3.5}$$

Although the subscripts contain the letter i, it is important to note that the subscript in this case is an index, and therefore does not represent partial differentiation. Using the introduced notation together with equations (3.4), we obtain for cell Ω_i

$$\int_{\Omega_i} \mathbf{q}_t dx + (\mathbf{f}(\mathbf{q}_{i+\frac{1}{2}}) - \mathbf{f}(\mathbf{q}_{i-\frac{1}{2}})) - \int_{\Omega_i} \mathbf{s}dx = 0. \tag{3.6}$$

3.2.2 Approximation of the solution at the cell faces

The next step concerns the determination of the values of $\mathbf{q}_{i+\frac{1}{2}}$, as determining them results in the possibility to compute the fluxes across the cell faces. The methods that will be implemented for the research in this report will be listed and explained below.

Second order central scheme

The simplest way to compute the value $\mathbf{q}_{i+\frac{1}{2}}$ is by taking the average of the values of $\mathbf{q}$ in the two neighboring cells of cell face $i + \frac{1}{2}$. I.e.:

$$\mathbf{q}_{i+\frac{1}{2}} = \frac{\mathbf{q}_i + \mathbf{q}_{i+1}}{2}. \tag{3.7}$$

Left and right cell face states

We now take a more general approach to determine the flux across the cell faces. For every cell face in the computational domain, we introduce two quantities for the solution $\mathbf{q}$; a left state and a right state. We denote the left state of cell face $i + \frac{1}{2}$'s solution by $\mathbf{q}_{i+\frac{1}{2}}^l$ and its corresponding right state by $\mathbf{q}_{i+\frac{1}{2}}^r$. Both quantities are to be determined by inter- or extrapolation from the neighboring cell solutions, for which we consider three different methods, distinguishable in their order of accuracy.

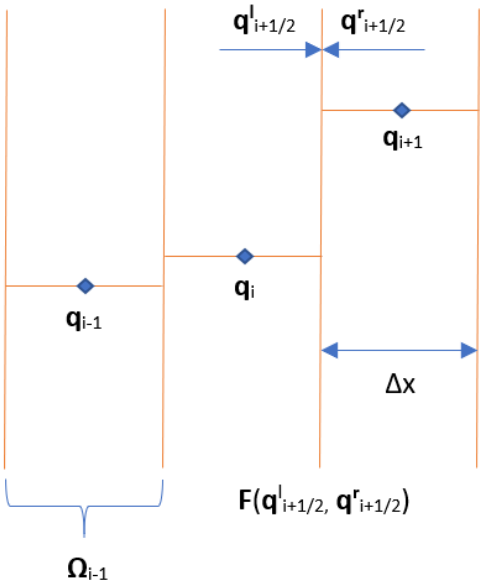

Figure 3.2: Computing the flux through a cell face by using its corresponding left and right state

a) First order upwind scheme: The easiest way to approximate $\mathbf{q}^l_{i+\frac{1}{2}}$ and $\mathbf{q}^r_{i+\frac{1}{2}}$ is by the first order upwind scheme, which sets the left and right cell face state equal to the direct left and right value of $\mathbf{q}$. In other words:

$$\mathbf{q}^l_{i+\frac{1}{2}} = \mathbf{q}_i, \quad \mathbf{q}^r_{i+\frac{1}{2}} = \mathbf{q}_{i+1}. \tag{3.8}$$

b) Second order upwind scheme unlimited For the second order upwind scheme, the left and right cell face states are computed in a more complex way compared to the first order upwind scheme by interpolating:

$$\mathbf{q}^l_{i+\frac{1}{2}} = \mathbf{q}_i + \frac{1+\kappa}{4}(\mathbf{q}_{i+1} - \mathbf{q}_i) + \frac{1-\kappa}{4}(\mathbf{q}_i - \mathbf{q}_{i-1}), \tag{3.9}$$

$$\mathbf{q}^r_{i+\frac{1}{2}} = \mathbf{q}_{i+1} + \frac{1+\kappa}{4}(\mathbf{q}_i - \mathbf{q}_{i+1}) + \frac{1-\kappa}{4}(\mathbf{q}_{i+1} - \mathbf{q}_{i+2}), \tag{3.10}$$

with $\kappa \in [-1, 1]$. For $\kappa = -1$, we have fully one-sided extrapolation. For $\kappa = 1$, we have second order central interpolation. Lastly, for $\kappa = \frac{1}{3}$ the approximation is third order accurate. Even though this method has a higher order of accuracy than scheme **a)**, equations (3.9) and (3.10) are non-positive discretizations due to their extrapolation term. The next method we propose deals with this issue.

c) Second order upwind scheme limited: To prevent non-positive discretization, we can implement a limiter function ϕ in scheme **b)** as follows:

$$\mathbf{q}^l_{i+\frac{1}{2}} = \mathbf{q}_i + \frac{1}{2}\phi(\mathbf{r}^l_{i+\frac{1}{2}})(\mathbf{q}_i - \mathbf{q}_{i-1}), \tag{3.11}$$

$$\mathbf{q}^r_{i+\frac{1}{2}} = \mathbf{q}_{i+1} + \frac{1}{2}\phi(\mathbf{r}^r_{i+\frac{1}{2}})(\mathbf{q}_{i+1} - \mathbf{q}_{i+2}). \tag{3.12}$$

The quantities $\mathbf{r}^l_{i+\frac{1}{2}}$ and $\mathbf{r}^r_{i+\frac{1}{2}}$ are dependent on the solution $\mathbf{q}$ in neighboring cell faces, which means that they are vector functions. Therefore, they are component wise given by:

$$\mathbf{r}^l_{i+\frac{1}{2}} = \frac{\mathbf{q}_{i+1} - \mathbf{q}_i + \epsilon}{\mathbf{q}_i - \mathbf{q}_{i-1} + \epsilon}, \tag{3.13}$$

$$\mathbf{r}^r_{i+\frac{1}{2}} = \frac{\mathbf{q}_i - \mathbf{q}_{i+1} + \epsilon}{\mathbf{q}_{i+1} - \mathbf{q}_{i+2} + \epsilon}, \tag{3.14}$$

with ϵ a parameter to be chosen small to prevent division by zero. In this report we take $\epsilon = 10^{-6}$. For the limiter function ϕ, we use the Minmod (M), Superbee (S) and BK limiter [5], defined as follows:

$$\phi_M(r) = \begin{cases} 0, & \text{if } r < 0 \\ r, & \text{if } 0 \leq r < 1 \\ 1, & \text{if } r \geq 1 \end{cases}, \tag{3.15}$$

$$\phi_S(r) = \begin{cases} 0, & \text{if } r < 0 \\ 2r, & \text{if } 0 \leq r < \frac{1}{2} \\ 1, & \text{if } \frac{1}{2} \leq r < 1 \\ r, & \text{if } 1 \leq r < 2 \\ 2, & \text{if } r \geq 2 \end{cases}, \tag{3.16}$$

$$\phi_{BK}(r) = \begin{cases} 0, & \text{if } r < 0 \\ 2r, & \text{if } 0 \leq r < \frac{1}{4} \\ \frac{1}{3} + \frac{2}{3}r, & \text{if } \frac{1}{4} \leq r < \frac{5}{2} \\ 2, & \text{if } r \geq \frac{5}{2} \end{cases}. \tag{3.17}$$

Again, important to note is that ϕ: $\mathbb{R} \to \mathbb{R}$ (all three versions). Since $\mathbf{q}$ is a vector function, the above expressions are evaluated component wise for q_1 and q_2. I.e., the limiter function is used twice; once for each component of $\mathbf{q}$.

3.2.3 Computing the flux across the cell face

Now that we have presented the methods to approximate the solution at the cell faces, we can compute the terms outside of the integrals in equation (3.6). The method to do so depends on the method used to compute $\mathbf{q}$ at the cell face.

Direct evaluation for second order central

For the second order central scheme, the obtained values for $\mathbf{q}_{i+\frac{1}{2}}$ will be directly substituted into equation (3.5) to obtain the flux through that cell face.

Flux difference splitting

The upwind schemes require a more complex method of evaluating the flux across the cell faces, as the solution in the cell face is extended to a left and right state instead of using just one quantity for the solution through the cell face as seen in the second order central scheme. The proposed method for the upwind schemes is the flux difference splitting method. Consider again the quasilinear form of the

conservative form of the one-dimensional SWE, as seen in equation (3.1). Recall that the diagonalized form of A is given by

$$A = R\Lambda R^{-1}, \text{ with } \Lambda = \begin{pmatrix} u - \sqrt{gH} & 0 \\ 0 & u + \sqrt{gH} \end{pmatrix}. \tag{3.18}$$

Now we define the matrices A^+ and A^- by

$$A^+ = R\Lambda^+ R^{-1}, \quad A^- = R\Lambda^- R^{-1}, \tag{3.19}$$

where Λ^+ and Λ^- are the positive and negative eigenvalue part of Λ, respectively. I.e.:

$$\Lambda^+ = \begin{pmatrix} \max(u - \sqrt{gH}, 0) & 0 \\ 0 & \max(u + \sqrt{gH}, 0) \end{pmatrix}, \tag{3.20}$$

$$\Lambda^- = \begin{pmatrix} \min(u - \sqrt{gH}, 0) & 0 \\ 0 & \min(u + \sqrt{gH}, 0) \end{pmatrix}. \tag{3.21}$$

To compute the flux at the cell face $i + \frac{1}{2}$of the i-th cell, we use the following formula:

$$\mathbf{F}(\mathbf{q}^l_{i+\frac{1}{2}}, \mathbf{q}^r_{i+\frac{1}{2}}) = \mathbf{f}(\mathbf{q}^l_{i+\frac{1}{2}}) + A^-(\mathbf{q}^r_{i+\frac{1}{2}})(\mathbf{q}^r_{i+\frac{1}{2}} - \mathbf{q}^l_{i+\frac{1}{2}}). \tag{3.22}$$

In (3.22), $\mathbf{f}$ is given by equation (3.5). Since A^- represents the propagation velocity of information from right to left, we can therefore set

$$A^-(\mathbf{q}^l_{i+\frac{1}{2}}, \mathbf{q}^r_{i+\frac{1}{2}}) = A^-(\mathbf{q}^r_{i+\frac{1}{2}}). \tag{3.23}$$

3.2.4 Time integration

The spatial domain has been discretized, and the solution $\mathbf{q}$ at the cell faces can be computed together with their corresponding fluxes. The next step is to decide on a proper time integration method. We require that the solutions remain positive during the time integration. This means for the time integration method $\mathbf{q}_i^{n+1} = w(\mathbf{q}_i^n)$ for a small enough time step:

$$\mathbf{q}_i^n \geq 0 \implies \mathbf{q}_i^{n+1} \geq 0. \tag{3.24}$$

This is to prevent unrealistic phenomena to occur in $\mathbf{q}$. As an example, $q_1 = gH$ should never become negative in the simulation, as this is physically speaking incorrect. Among the available time integration methods, the RK1 (Forward Euler) and the RK3b schemes are examples satisfying property (3.24) [6]. These schemes will therefore be used in this report. We introduce the discrete time

$$t(n) = n\Delta t, \quad n = 0, 1, 2... \tag{3.25}$$

Then, the RK1 time integration method for this problem is given by

$$\mathbf{q}_i^{n+1} = \mathbf{q}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}(\mathbf{q}^n_{i+\frac{1}{2}}) - \mathbf{F}(\mathbf{q}^n_{i-\frac{1}{2}})\right) + \Delta t \mathbf{s}_i^n, \tag{3.26}$$

where $\mathbf{s}_i$ is the approximation of the source term in the i-th cell component wise given by

$$\mathbf{s}^n_{1,i} = 0, \quad \mathbf{s}^n_{2,i} = g q^n_{1,i} \frac{h^n_{i+1} - h^n_{i-1}}{2\Delta x} \Delta x, \tag{3.27}$$

where $g = 9.8$ N/kg. Since we will for a large part of the report be dealing with the limited second order upwind scheme for the space discretization, we include another time integration method. This is the explicit RK3b scheme [6], and it is given by

$$\mathbf{q}_i^{n+1} = \mathbf{q}_i^n + \frac{1}{6}(\mathbf{k}_1 + \mathbf{k}_2 + 4\mathbf{k}_3). \tag{3.28}$$

If we introduce a vector function **G** defined by

$$\mathbf{G}(\mathbf{q}^n)_i = \frac{\Delta t}{\Delta x}(\mathbf{F}(\mathbf{q}^n_{i-\frac{1}{2}}) - \mathbf{F}(\mathbf{q}^n_{i+\frac{1}{2}})) + \Delta t \mathbf{s}^n_i, \tag{3.29}$$

hen the vectors $\mathbf{k}_1$, $\mathbf{k}_2$ and $\mathbf{k}_3$ are given by

$$\mathbf{k}_1 = \mathbf{G}(\mathbf{q}^n) \tag{3.30}$$
$$\mathbf{k}_2 = \mathbf{G}(\mathbf{q}^n + \mathbf{k}_1) \tag{3.31}$$
$$\mathbf{k}_3 = \mathbf{G}(\mathbf{q}^n + \frac{1}{4}(\mathbf{k}_1 + \mathbf{k}_2)). \tag{3.32}$$

3.2.5 Time step size

Time integration methods come with criteria on the step size Δt to guarantee (absolute) stability of the method. For the RK1 method, we require that for each cell face the time step Δt:

$$\Delta t < \frac{\Delta x}{\max(\left|u \pm \sqrt{gH}\right|)}. \tag{3.33}$$

Note that the denominator of the RHS of inequality (3.33) contains the eigenvalues of matrix A, as given in (3.1). The flux difference splitting scheme also has a step size criterion for stability. This criterion is more restrictive than (3.33). For each cell face we need

$$\Delta t < \frac{\Delta x}{2 \cdot \max(|u \pm \sqrt{gH}|)}. \tag{3.34}$$

Then combining (3.33) and (3.34) gives the following requirement for the time step Δt for using RK1 on the spatially discretized one dimensional SWE:

$$\Delta t < \frac{\Delta x}{2 \cdot \max_i(|u^n_{i+\frac{1}{2}} \pm \sqrt{gH^n_{i+\frac{1}{2}}}|)} = t_{max,1}, \quad i \in \{0, 1, 2, ..., \mathrm{N}\}. \tag{3.35}$$

The RK3 method also has a restriction on the step size Δt to guarantee (absolute) stability of the method. We can perform a similar linear stability analysis on RK3 as seen in [7] on RK4. We introduce the following equation, known as the test equation, which plays an important role in the stability analysis of time integration methods:

$$\begin{cases} y' & = \lambda y, \quad t > t_0 \\ y(t_0) & = y_0 \end{cases}, \tag{3.36}$$

where $\lambda \in \mathbb{R}$. The RK3b method has the following Butcher tableau [6], which is a systematic way to write down the relevant coefficients in the time integration methods of the Runge-Kutta family.

0	0	0	0
1	1	0	0
1/2	1/4	1/4	0
	1/6	1/6	2/3

Therefore, if we want to time integrate the test equation (3.36) using RK3b, the solution at t_{n+1} is approximated by:

$$w_{n+1} = w_n + \frac{1}{6}(k_1 + k_2 + 4k_3), \tag{3.37}$$

with

$$k_1 = \Delta t f(t_n, w_n) = \lambda \Delta t w_n, \tag{3.38}$$
$$k_2 = \Delta t f(t_n + \Delta t, w_n + k_1) = \lambda \Delta t (w_n + \lambda \Delta t w_n), \tag{3.39}$$
$$k_3 = \Delta t f(t_n + \frac{1}{2}\Delta t, w_n + \frac{1}{4}(k_1 + k_2) = \lambda \Delta t (w_n + \frac{1}{2}\lambda \Delta t w_n + \frac{1}{4}(\lambda \Delta t)^2). \tag{3.40}$$

It can then be verified that

$$w_{n+1} = \left(1 + \lambda \Delta t + \frac{1}{2}(\lambda \Delta t)^2 + \frac{1}{6}(\lambda \Delta t)^3\right) w_n. \tag{3.41}$$

For stability, we require for the expression which acts as the multiplicative factor for w_n in (4.37) that

$$-1 < 1 + \lambda \Delta t + \frac{1}{2}(\lambda \Delta t)^2 + \frac{1}{6}(\lambda \Delta t)^3 < 1. \tag{3.42}$$

Since $\Delta t > 0$, we observe that for $\lambda > 0$, (3.42) can never be satisfied. Therefore, we consider the case $\lambda < 0$. Let $x = \lambda \Delta t$, then (3.42) can be split into 2 separate inequalities:

$$\begin{cases} 1 + \frac{1}{2}x + \frac{1}{6}x^2 & > 0, \\ 2 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 & > 0. \end{cases} \tag{3.43}$$

The first inequality in (3.43) is satisfied for all x. For the second inequality in (3.43), it can be shown that this holds for

$$x > -1 - \frac{1}{\sqrt[3]{\sqrt{17} - 4}} + \sqrt[3]{\sqrt{17} - 4} \approx -2.513. \tag{3.44}$$

Hence, one will find the following stability requirement for RK3b:

$$\Delta t < \frac{5}{2} \cdot \frac{1}{|\lambda|}. \tag{3.45}$$

Note that the quantity found in (3.44) has been replaced in (3.45) by $\frac{5}{2}$. This is done for simplicity reasons. Transitioning back to the SWE, λ represents the eigenvalues of matrix A given in (3.1). I.e. $\lambda = u \pm \sqrt{gH}$. A combination of (3.34) and (3.45) yields the following time step criterion for RK3b:

$$\Delta t < \frac{\min(5, \Delta x)}{2 \cdot \max_i(|u^n_{i+\frac{1}{2}} \pm \sqrt{gH^n_{i+\frac{1}{2}}}|)} = \Delta t_{max,2}, \quad i \in \{0, 1, 2, ..., \mathrm{N}\}. \tag{3.46}$$

4 Test case: Tank of water with moving bottom

This report focuses on testing the shallow water equations on scenarios with moving bottoms. The test case is a scenario on which the one-dimensional SWE will be tested using the numerical solvers as described above. We consider a large, rectangular tank filled with water from the side. The tank has a length of 100 metres and is filled with water until the total water height is 50 metres. The tank and the numerical solvers have been implemented in MATLAB. The initial situation looks the following:

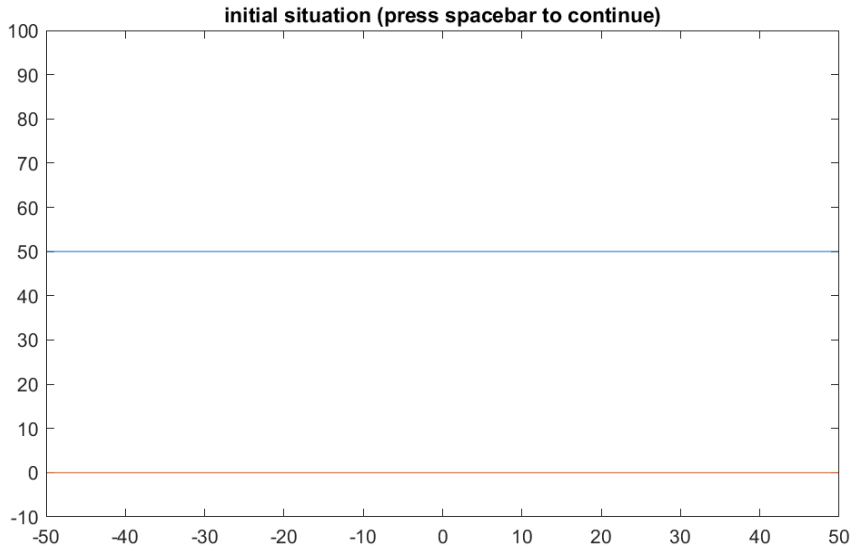

Figure 4.1: Initial situation of the rectangular tank filled with water. The orange line represents the bottom, the blue line is the water height level.

In other words, for the initial situation we have $h(x,0) = 0,\ \eta(x,0) = 50$ m for $x \in [-50, 50]$. We set the initial velocity equal to 0 ($u = 0$). Hence, there is no background flow. The computational domain is partitioned into 400 cells, resulting in $\Delta x = 0.25$ m. At $x = -50$ m and $x = 50$ m, we have the homogeneous Neumann boundary condition

$$\mathbf{q}_x = 0. \tag{4.1}$$

This means we have open boundaries, and that the derivative of the height of the water and the velocity are equal to 0. We discuss two types of deformations of the bottom in the simulation: Instantaneous deformation, and deformation over time. We assume that we can reshape the bottom of the tank in any form at any time during the simulation. Both time integration methods will be tested. For RK1, we use $\Delta t = 0.0005$ s for the second order central scheme since this method does not use flux difference splitting. Similarly, for RK3b, we test the case $\Delta t = 0.001$ s to see whether the time step can be increased. If not, $\Delta t = 0.0005$ s will be shown too. For the upwind schemes, which do use flux difference splitting, we use a time step size depending on the time integration method. RK1 uses a time step of $\Delta t = 0.9\Delta t_{max,1}$, RK3b uses a time step of $\Delta t = 0.9\Delta t_{max,2}$. Both types of deformation will be handled using both proposed time integration methods. For the second order upwind unlimited scheme, we set $\kappa = \frac{1}{3}$ to achieve the highest order of accuracy.

4.1 Instantaneous deformation

This type of deformation occurs as soon as the simulation starts. The deformation that has occurred then remains fixed for the duration of the simulation. Firstly, we consider an instantaneous deformation of

the bottom of the tank. This is in the form of a rectangular bump given by

$$h(x,t) = \begin{cases} -10, & \text{if } x \in [-10,10] \\ 0, & \text{else} \end{cases}, \quad \text{for } t \in [0,1]. \tag{4.2}$$

4.1.1 The RK1 method (Forward Euler)

The RK1 method is considered to be a less accurate method compared to a higher order method such as the RK3b method. For $\mathcal{O}(\Delta x^2)$ or $\mathcal{O}\Delta(x^3)$ space discretization methods, i.e. second order upwind (un)limited, it makes more sense to use the RK3b method. We can expect that RK1 does not handle these methods well.

Second order central scheme

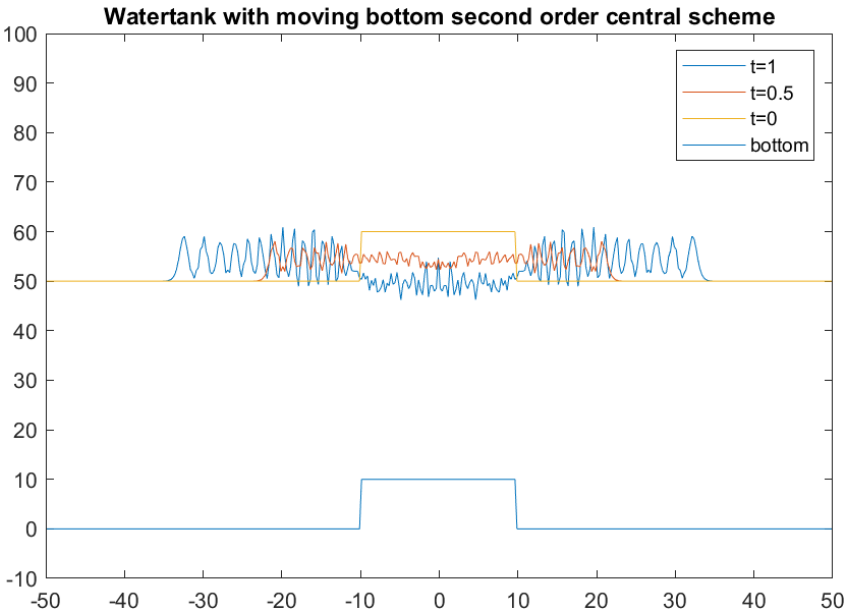

Figure 4.2: Instantaneous deformation with second order central and RK1.

With a time step of $\Delta t = 0.0005$, figure 4.2 shows that the second order central scheme here produces spurious oscillations, which increase over time in the simulation. Above that, it shows behavior which is typical for this scheme; the water surface seems to be 'nervous'. The surface contains sharp angles, giving it a non-smooth look.

First order upwind scheme

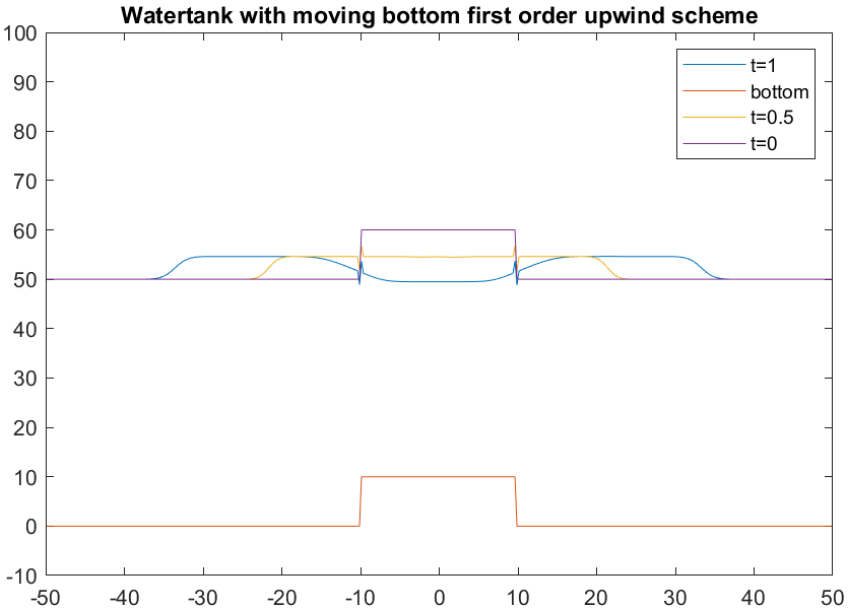

Figure 4.3: Instantaneous deformation with first order upwind and RK1.

This scheme overall provides a more smooth representation of the water surface compared to the second order central scheme. This is explainable by the facts that the upwind schemes use the flux difference splitting scheme to compute the flux across the cell faces and upwind schemes take into account the direction of flow of the water. A few interesting observations can be made. The method characterizes itself with an odd shape of the water surface for the x-values where the deformation of the bottom both starts and ends, where, for both sides, a noticeable, fixed oscillation is created. The relatively large jumps and drops in the water level are explained by the sudden change in bottom height. The implementation of this method in MATLAB does not account for this.

Second order upwind schemes

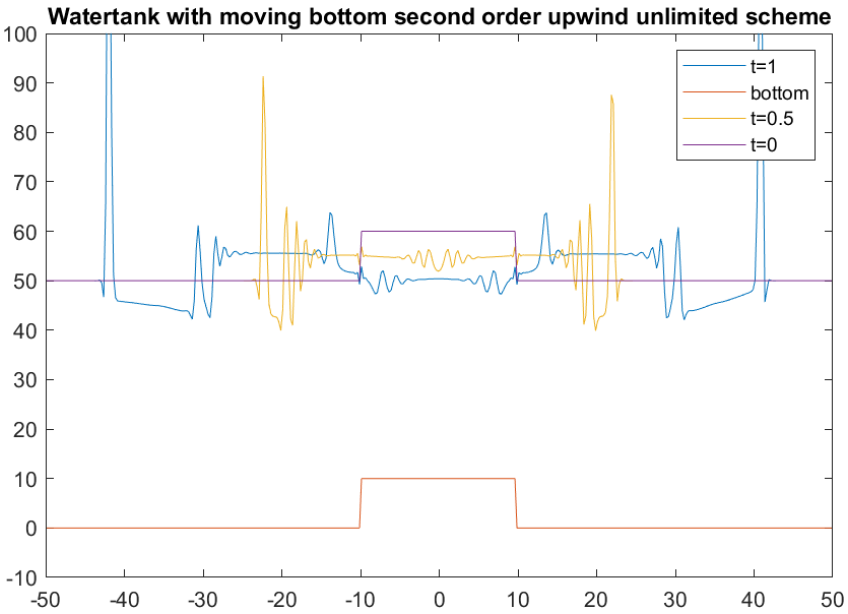

Figure 4.4: Instantaneous deformation with unlimited second order upwind and RK1.

As mentioned in the introduction of this section, we already expected that RK1 combined with the second order upwind unlimited scheme should not go together well due to the higher order of accuracy of the upwind scheme. Figure 4.4 justifies this hypothesis. Spurious oscillations already present themselves relatively early on in the simulation at $t = 0.5$. By the end of the simulation ($t = 1$), the largest oscillation seen in $t = 0.5$ is enhanced to the point where its peak is not even visible anymore in the plot. We therefore consider the combination of RK1 and second order upwind unlimited not a reliable method to simulate the test case. This should already give a hint for the results for the limited scheme. We have introduced three different limiter functions to prevent non-positive discretized which might occur in the the unlimited second order upwind scheme. The performances of these limiter functions are seen below.

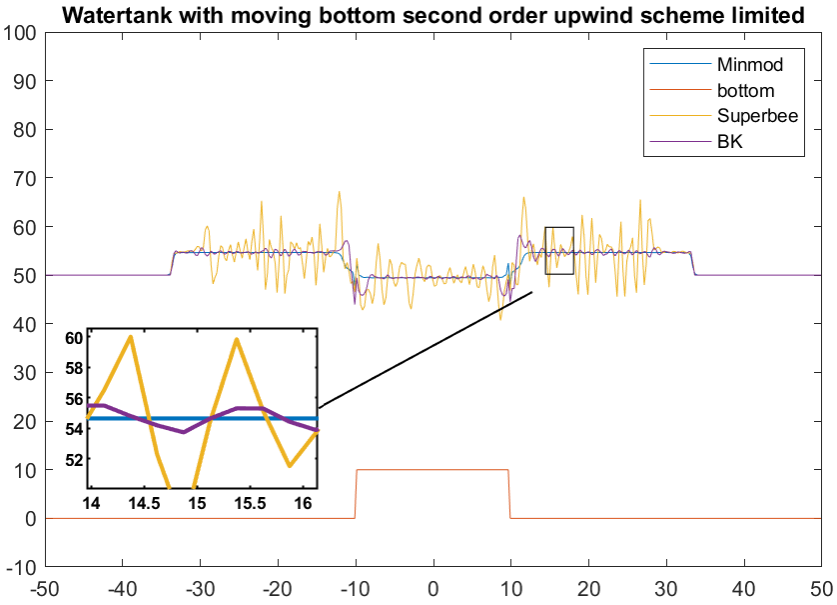

Figure 4.5: Instantaneous deformation with limited second order upwind and RK1 ($t = 1$).

What is immediately visible from figure 4.5, is the performance of the Superbee limiter function. It is able to contain the oscillations to some extent, however they are still very visible throughout the simulation. The next noticeable line is the purple graph. This is the performance of the BK limiter. It suppresses oscillations well, but minor oscillations are still visible, especially near the boundaries of the bump. The Minmod limiter seems to perform best in this situation. The blue graph is not as visible as the other graphs when moving away from the bump, indicating good performance. The enhancement clearly shows that Minmod ensures that the water surface at the top of the wave remains straight, giving it a somewhat similar look to figure 4.3. Minmod and BK show minor differences near the wave front. The difference between the second order upwind limited with Minmod and the first order upwind scheme shows itself on the wave fronts travelling to the boundaries of the water tank. The limited second order upwind scheme produces much steeper wave fronts compared to the first order upwind scheme.

4.1.2 The RK3b method

Second order central scheme

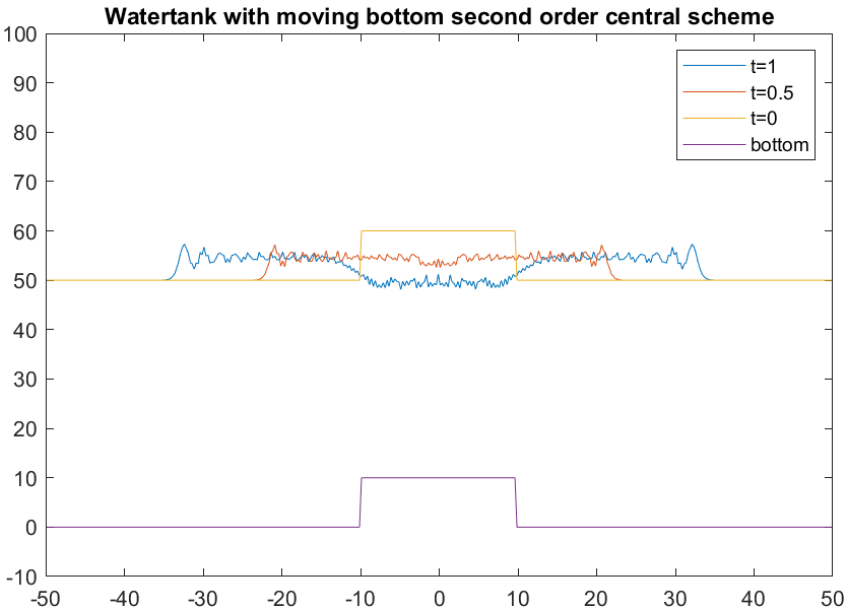

Figure 4.6: Instantaneous deformation with second order central and RK3b.

As a reminder, note that the all lines except for the line for the bottom of the tank in figure 4.6 represent the water surface at different times. The purple line represents the water surface at $t = 0$, and is identically shaped to the bottom deformation. The water surface at $t = 0.5$ and $t = 1$ approximated by this scheme has physically speaking unnatural shapes. The water surface contains sharp angles in itself at these time steps, giving it a non-smooth look. This is not uncommon for the second order central scheme. Above that, the scheme creates oscillations which increase in size when shifting away from the middle of the tank.

First order upwind scheme

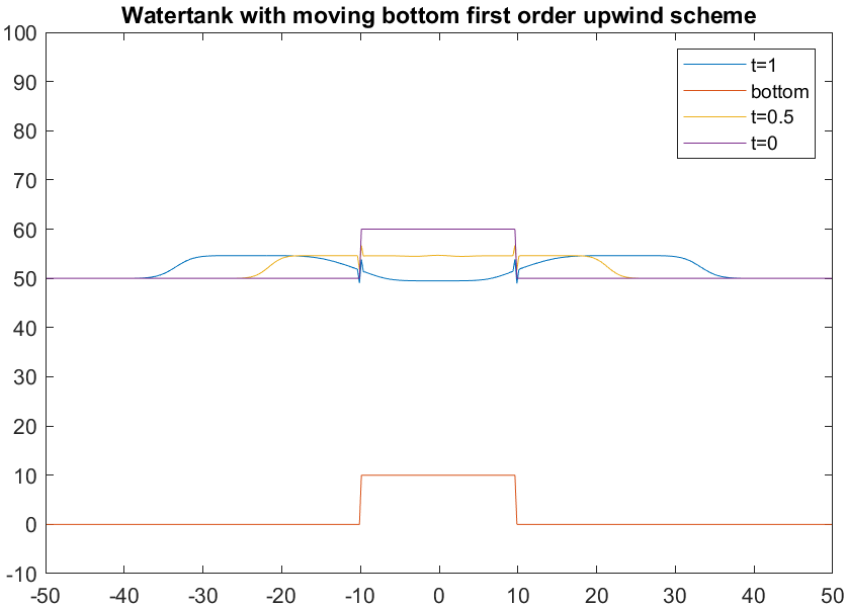

Figure 4.7: Instantaneous deformation with first order upwind and RK3b.

The image in figure 4.7 shows a very similar, if not identical, image compared to figure 5.3. This implies that the first order upwind scheme works well together with both the RK1 and the RK3b time integration methods. Since the first order upwind scheme, as the name says, is a $\mathcal{O}(\Delta x)$ space discretization method, Forward Euler can be used as an accurate time integration method, with appropriate time step choice. Notice again the less steep wave fronts the first order upwind schemes create.

Second order upwind schemes

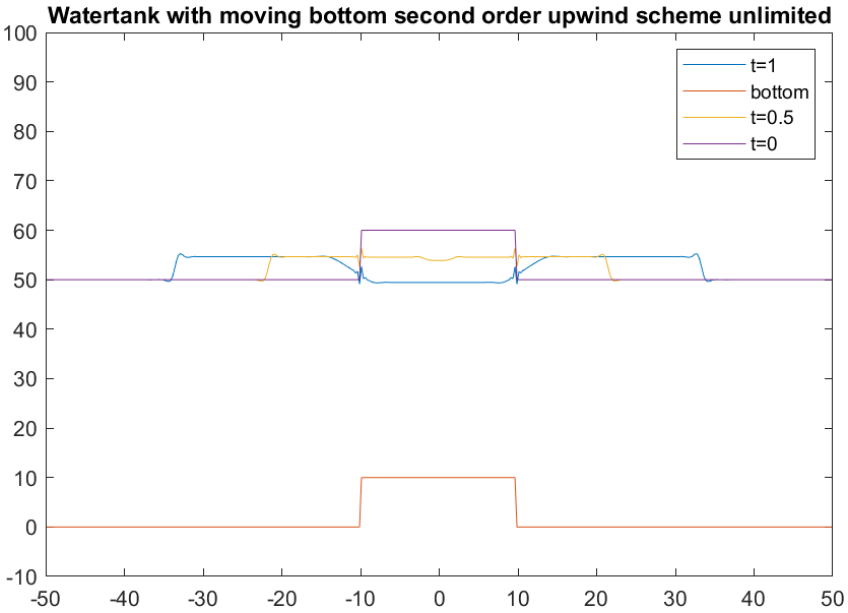

Figure 4.8: Instantaneous deformation with unlimited second order upwind and RK3b.

The hypothesis mentioned in the beginning of this section on the RK1 method is again justified by figure 4.8. Compare this figure with figure 4.4. The RK3b method works together with unlimited second order upwind a lot better than the RK1 method does. Figure 4.8 shows a smooth, moreover realistic, development of the water surface over time as a result of the incurred deformation. It has minor differences compared to figure 4.7, i.e. first order upwind. This is noticeable at the top of the wave front. Here, small peaks are observable in figure 4.8, whereas in figure 4.7, such an image is not present. The limited version of this scheme should therefore only have minor impact on the water surface.

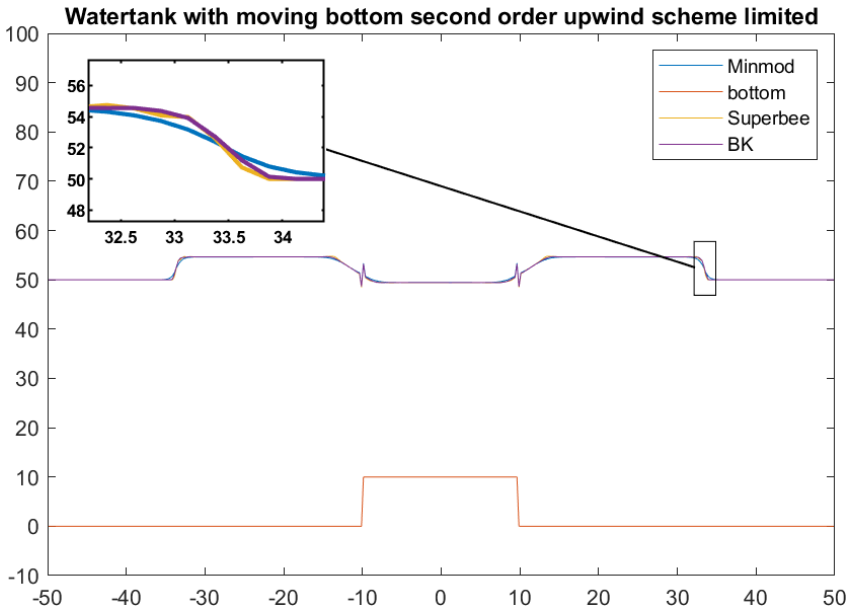

Figure 4.9: Instantaneous deformation with limited second order upwind and RK3b ($t = 1$).

As expected, the limiter functions do not have a major influence on the water surface. All limiter functions perform practically the same, where they smooth out the small peaks at the top of the wave front from figure 4.8, creating a similar image as seen in figure 4.7 for $t = 1$, only with significantly steeper wave fronts.

4.1.3 Comparison

From the results of an instantaneous deformation, we see that, out of all the options, the second order central scheme for both time integration methods seem to provide the worst solution. The methods are characterized by their small, if not increasing, sharp-angled oscillations which provide a non-smooth look to the water surface. Extending the time limit of the simulation would lead to heavily increasing oscillations for the chosen time step. The results for the first order upwind schemes show minor to no differences regarding the shape of the water surface throughout the simulation when comparing between RK1 and RK3b. RK1 is computationally less expensive, however for this test case the computational time is practically the same for both time integration methods. For the second order upwind schemes, be it unlimited or limited, we definitely prefer the use of RK3b as a suitable time integration method, due to the fact that we are using higher order space discretization methods in this case. This is justified by figures 4.8 and 4.9, as they both produce smooth, realistic results. Should one choose to implement RK1 for the limited second order upwind schemes, choosing the Minmod limiter function produces relatively the best results. For RK3b, the choice of limiter function is in this case insignificant, as all produce practically the same result.

4.2 Deformation over time

As the name suggests, we also consider the case that the bottom may deform during the simulation. Now, the same deformation of the bottom as seen in the section on the instantaneous bottom will occur, but will develop over time. The plots that are presented in this section are results of simulating 2 seconds of the water tank that incurs a deformation in its bottom. The deformation occurs in stages depending on the value of t. By stages, it is meant that the velocity of deforming vertically upwards differs, depending on the stage. The total deformed bottom over time can be described as follows.

$$h(x,t) = \begin{cases} -2.5 \cdot 3t & \text{if } x \in [-10,10],\ t \in \left[0, \frac{1}{3}\right], \\ -2.5 - 5 \cdot 3(t - \frac{1}{3}) & \text{if } x \in [-10,10],\ t \in \left(\frac{1}{3}, \frac{2}{3}\right], \\ -7.5 - 2.5 \cdot 3(t - \frac{2}{3}) & \text{if } x \in [-10,10],\ t \in \left(\frac{2}{3}, 1\right], . \\ -10 & \text{if } x \in [-10,10],\ t \in (1,2], \\ 0 & \text{else} \end{cases} \tag{4.3}$$

Again, the results will be categorized according to the used time integration method.

4.2.1 The RK1 method (Forward Euler)

Second order central scheme

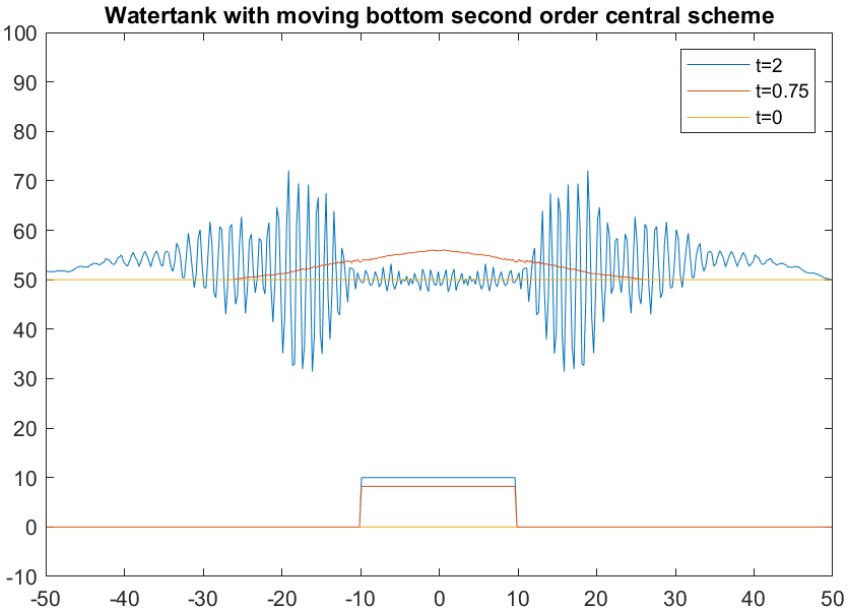

Figure 4.10: Deformation over time with second order central ($\Delta t = 0.001$) and RK1.

From figure 4.10, we see that for the first second in the simulation, the second order central scheme is able to simulate the water surface rather well. However, advancing to the next second, small oscillations start to occur at the water surface away from the deformation, which are heavily increased in size until finally put to a hard stop by reaching the time limit. We suspect that the time step $\Delta t = 0.001$ is too large a step for a stable RK1 time integration method. Therefore, we try the same simulation, now with a time step of $\Delta t = 0.0005$.

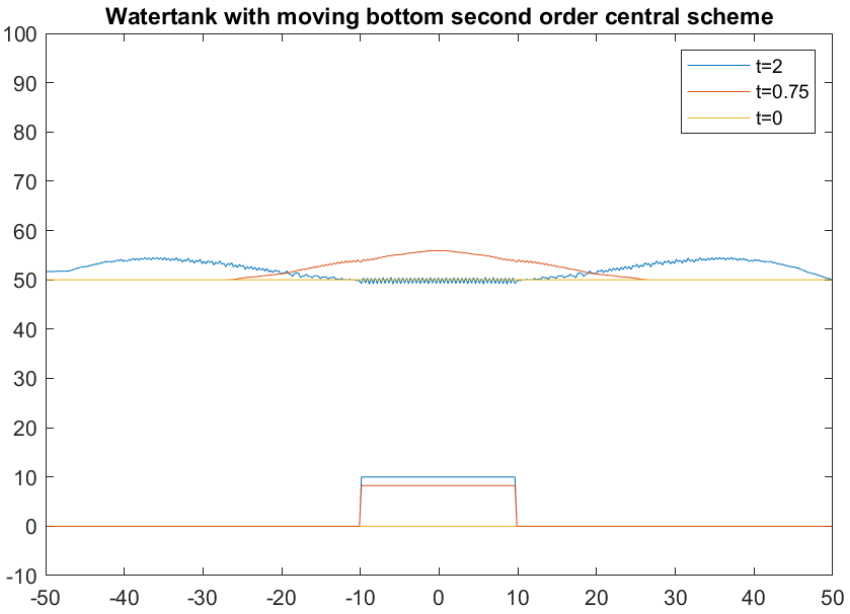

Figure 4.11: Deformation over time with second order central ($\Delta t = 0.0005$) and RK1.

The result is shown in figure 4.11. This image indicates the implemented RK1 time integration method is a stable one during this simulation. Overall, it shows a nice representation of the development of the water surface as a result of this deformation. The only detail that is visible throughout the entire water surface, is the typical appearance of using the second order central scheme, which are the tiny, contained oscillations which give a non-smooth look to the surface. The downside of using this time step is that significant increase in computational time.

First order upwind scheme

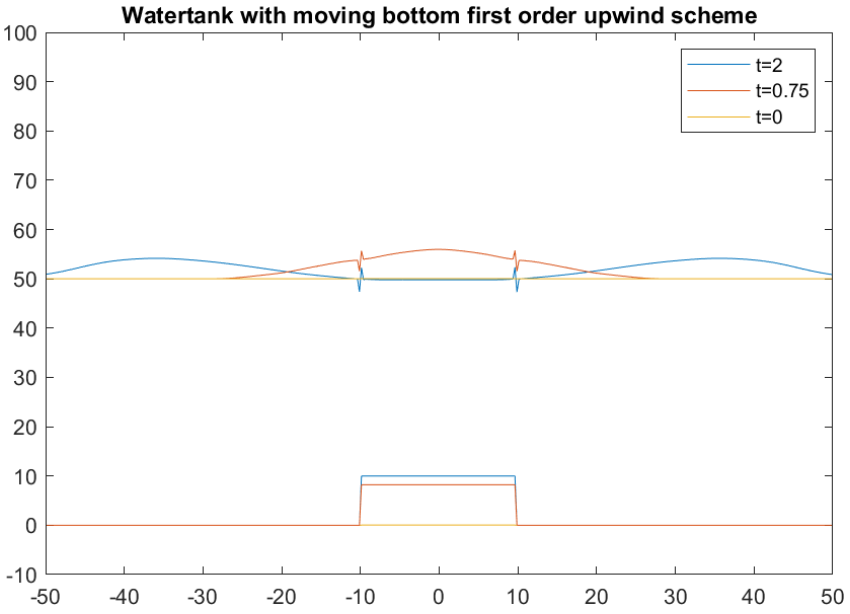

Figure 4.12: Deformation over time with first order upwind and RK1.

The image produced in figure 4.12 might be characterized by calling it a smooth representation of what is seen in figure 4.11. First order upwind here shows a good, realistic representation of the water surface.

The only downside of the implementation in MATLAB is the visible distortions at the boundaries of the rectangular bump.

Second order upwind schemes

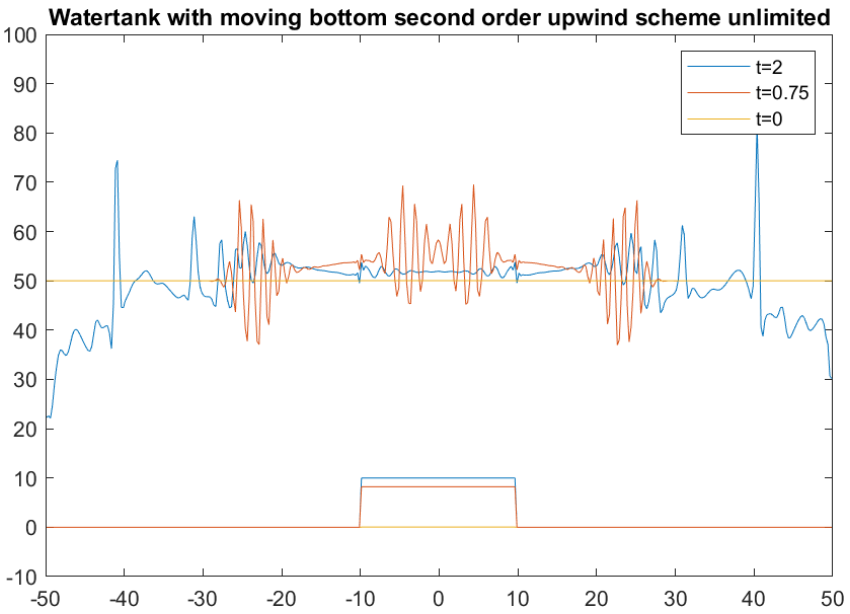

Figure 4.13: Deformation over time with unlimited second order upwind and RK1.

Again, we see similar results in figure 4.13 when comparing it to figure 4.4, which shows the performance of the unlimited second order upwind scheme using RK1. Already early on in the simulation, when the bump is still developing, large oscillations present themselves in the water surface, which increase in size when continuing the simulation. For the same reason as given in the section of the instantaneous deformation, this combination of space discretization scheme and time integration method do not work well together, and already provide insight on the results for limited second order upwind.

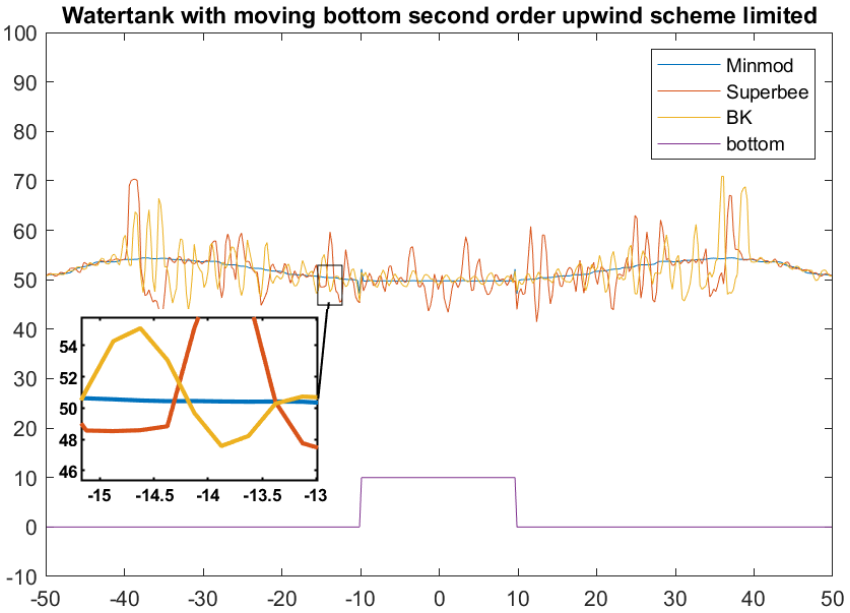

Figure 4.14: Deformation over time with limited second order upwind and RK1 ($t = 2$).

Figure 4.14 provides somewhat similar results to figure 4.5, which also shows the performance of the limiter functions for limited second order upwind using RK1. Again, the use of the Superbee limiter function is an immediate eyecatcher in figure 4.14. It shows many oscillations throughout the water surface, but is able to create a somewhat accurate, overall image of the trend of the water surface, with help of the graphs of the other limiter functions. The use of the BK limiter function shows somewhat worse results if comparing its performance in figure 4.5, with larger oscillations. Again, Minmod seems to come out on top in terms of performance, showing a smooth water surface.

4.2.2 The RK3b method

Second order central scheme

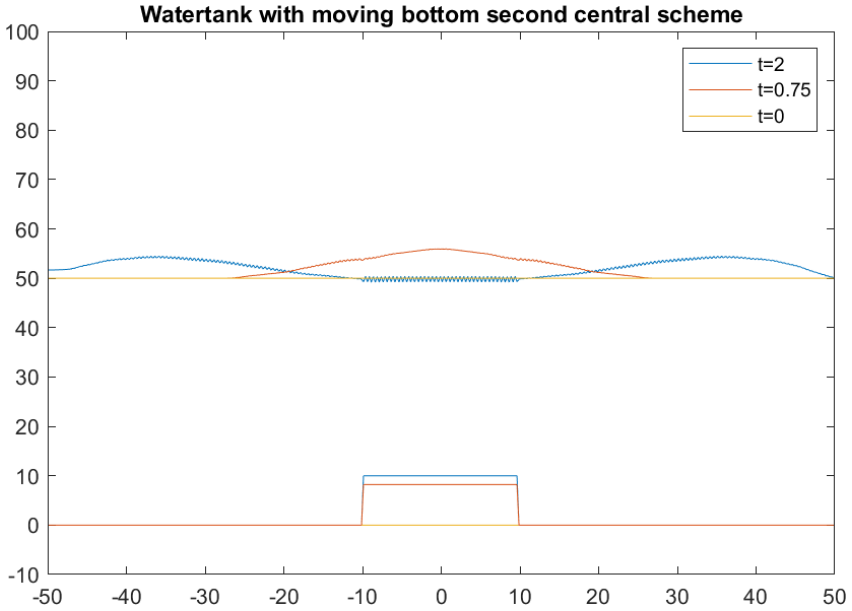

Figure 4.15: Deformation over time with second order central and RK3b.

Note that the time step size used here is $\Delta t = 0.001$. The result is shown in figure 4.15, which shows that RK3b behaves stable throughout the simulation. It produces an overall realistic representation of the water surface over time, with only the typical aspect of the second order central scheme showing throughout the water surface.

First order upwind scheme

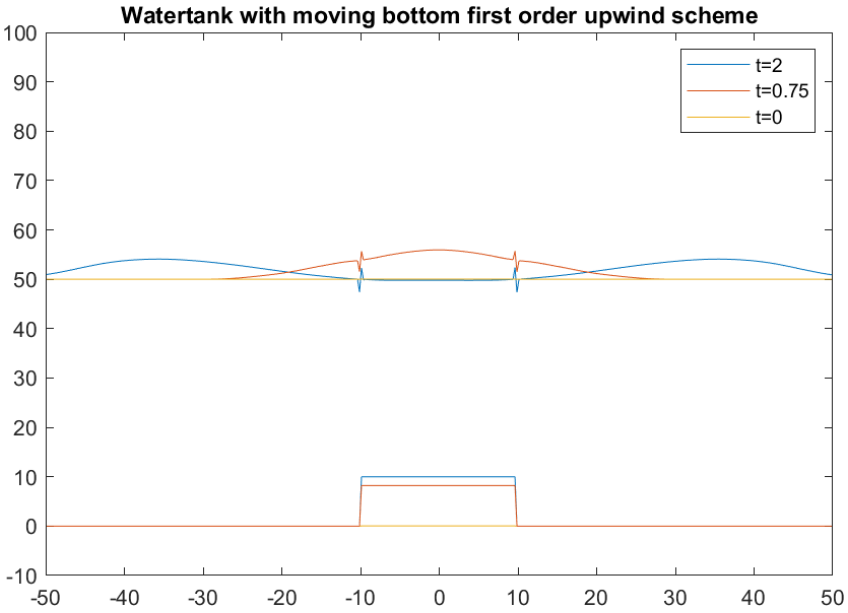

Figure 4.16: Deformation over time with first order upwind and RK3b.

Again, it a similar result when comparing the time integration methods for instantaneous deformation using first order upwind. The results in figure 4.16 are similar, if not identical, to the results in figure 4.12, showing a good, realistic representation of the water surface apart from the small distortions near the boundaries of the bump. The small to no differences in the results indicate that RK1 is a suitable for the first order upwind scheme, for the same reason as given in the previous section on instantaneous deformation.

Second order upwind schemes

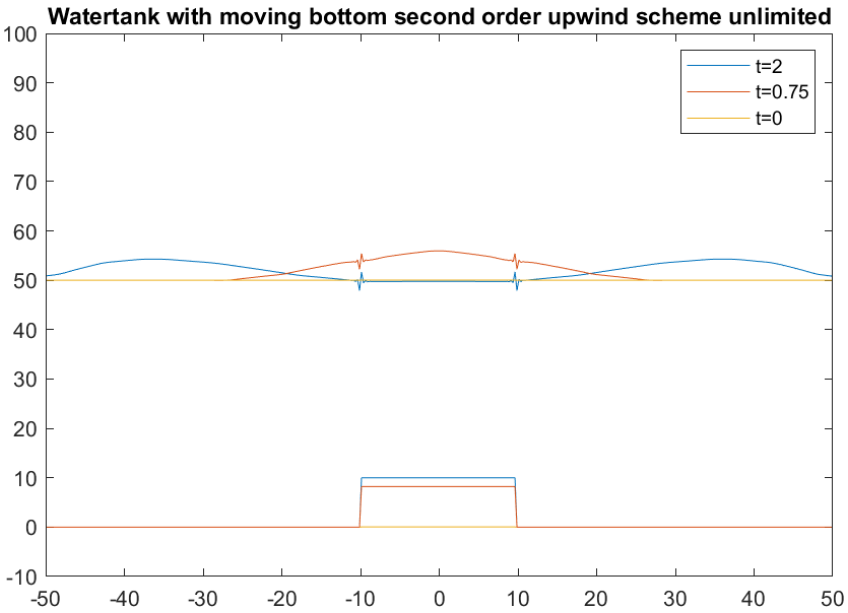

Figure 4.17: Deformation over time with unlimited second order upwind and RK3b.

The unlimited second order upwind scheme performs much better with RK3b as seen in figure 4.17, if we compare this with the use of RK1 in figure 4.13. The figure shows a smooth water surface apart from the small distortions, and looks similar to figure 4.16, which shows the results for the first order upwind. Therefore, the limiter functions in the limited version of this scheme will not play an impactful role for the results.

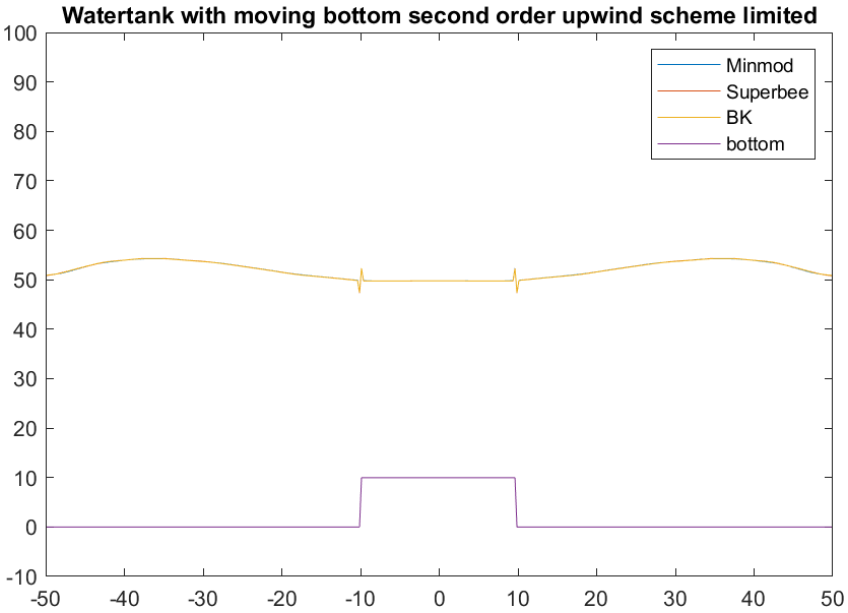

Figure 4.18: Deformation over time with limited second order upwind and RK3b ($t = 2$).

What we expected is indeed the case, as seen in figure 4.18. The performance of the limiter functions are non-distinguishable, as the water surface graphs all seem to overlap. I.e., the choice of limiter function in limited second order upwind does not influence the results we will obtain for the water surface.

4.2.3 Comparison

The use of the second order central scheme is for both time integration methods is again not a reliable choice to obtain accurate results due to its typical appearance. Above that, the computational time for second order central is significantly longer compared to the upwind schemes, due to the fact that the upwind schemes have a variable, but larger time step size. Again, for the first order upwind schemes, we overall find for both RK1 and RK3b a good, realistic representation of the water surface. The unlimited second order upwind scheme does not perform well with RK1, as expected, but produces a similar result compared to the first order upwind scheme with RK3b. The choice of limiter function does not matter for RK3b, but again Minmod seems to be the best option if RK1 with limited second order upwind is desired. In all cases, the impact of adding a higher velocity of deformation stage on the bump provides most change to the initial deformation of the water surface, and when the deformation decelerates to its last stage, the bump of the deformation of the water surface drops.

5 Conclusion & further research

The purpose of this paper was exploring the possibility of using the shallow water equations on a body of water for which the bottom incurs a deformation which occurs either instantaneously or over time. The chosen route became the implementation of finite volume integration, as we found that waves in shallow water are apt to becoming steeper, possibly creating breaking wave fronts. Finite volume integration handles the discontinuities better than standard finite difference methods.

The implementation of the problem in MATLAB came with proposing several numerical schemes for both flux approximation and time integration. To approximate the fluxes, the solutions at the cell faces need to be computed first. We proposed 4 different schemes for this. These were the second order central scheme, first order upwind scheme, unlimited second order upwind scheme and limited second order upwind scheme, together with the three different limiter functions Minmod, Superbee and BK. Secondly, the flux across the cell faces need to be calculated. We proposed 2 different ways to do so. For the second order central scheme, the flux is directly evaluated using the found flux function in the derivation of the one-dimensional shallow water equations. For the other cell face approximation schemes, we turned to flux difference splitting. Lastly, time integration is needed. The time integration methods proposed were the RK1, or Forward Euler, and the RK3b schemes. The numerical schemes were tested and compared in a test case. The test case was a 100 metres wide tank of water, of which the bottom incurred either an instantaneous deformation or a deformation over time, eventually resulting in the same deformation; a rectangular bump of 20 metres wide and 10 metres high.

The results between both types of deformation show similar results. The second order central scheme does not perform well for either deformation and either time integration method. Although in some cases providing an overall nice solution for the water surface, the scheme characterized itself by its nervous appearance. Thus, this scheme together with a direct flux evaluation is not a suitable option for the problem. Turning towards the schemes using flux difference splitting, we found interesting results. The first order upwind scheme is suitable for both RK1 and RK3b, as this scheme is an $\mathcal{O}(\Delta x)$ space discretization method. This scheme provided smooth, realistic results for the water surface in both deformation cases. The unlimited second order upwind scheme does not perform well with RK1 for both deformation cases, as this was a third order accurate method in our case. The limited version of this scheme with RK1 only seemed to work properly with the use of the Minmod limiter function. The second order upwind schemes performed better using RK3b as the time integration method for both deformation cases, providing good, realistic representations of the water surface given the deformation. The use of a limiter function in this case has little to no impact on the water surface.

As a continuation on this research project, one could extend this project by considering a similar problem for the two-dimensional shallow water equations. This report has only considered a problem on which the one-dimensional shallow water equations are applicable. For example, the 2D water tank could firstly be extended to a 3D water tank, on which one could implement tons of different interesting bottom deformation to simulate the behavior of the water surface using the two-dimensional shallow water equations, for which a challenge would be to find a suitable bottom function $h(x, z, t)$. Above that, the only proper flux approximation method that has been proposed here is the flux difference splitting scheme. This scheme is based on the linearized shallow water equations, making it less accurate. For another good flux approximation method, one could read about the Godunov scheme, for which a well explained version is available in a report by a fellow student [8].

Bibliography

[1] A.J. de Saint-Venant et al. Théorie du mouvement non permanent des eaux avec applications aux crues des rivières et à l'introduction des marées dans leur lit. *Comptes Rendus de l'Académie des Sciences de Paris* vol. 73, 1871.

[2] C. L. Fefferman. Existence and smoothness of the Navier-Stokes equation. *The millennium prize problems* 57:67, 2006.

[3] J.J. Stoker. *Water Waves: The Mathematical Theory with Applications*. Interscience Publishers, 1957.

[4] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, Berlin, Third edition, 2009.

[5] B. Koren. *A robust upwind discretization method for advection, diffusion and source terms*. Centrum voor Wiskunde en Informatica Amsterdam, 1993.

[6] W. Hundsdorfer, B. Koren, J. Verwer, et al. A positive finite-difference advection scheme. *Journal of Computational Physics*, 117(1):35-46, 1995.

[7] C. Vuik, F. Vermolen, M. van Gijzen., M. Vuik. *Numerical Methods for Ordinary Differential Equations*. Delft Academic Press, Delft, Second edition, 2018.

[8] R. Hoekstra, *Finite Volume Integration of the shallow water equations, applied to the wave behind a submarine*. Eindhoven University of Technology Department of Mathematics, 2020.