Simulation of drying and flooding in a tidal embayment using the level set method

A thesis submitted to the Delft Institute of Applied Mathematics in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE in APPLIED MATHEMATICS

by

NICO VAN DEN HEUVEL

Delft, the Netherlands October 2010

Copyright © 2010 by Nico van den Heuvel. All rights reserved.
MSc THESIS APPLIED MATHEMATICS

“Simulation of drying and flooding in a tidal embayment using the level set method”

NICO VAN DEN HEUVEL

Delft University of Technology

Daily supervisor          Responsible professor
Dr. H.M. Schuttelaars     Prof.dr.ir. A.W. Heemink

Second supervisor
Dr.ir. F.J. Vermolen

October 2010              Delft, the Netherlands
## Contents

1 Introduction  
   1.1 Tides  

2 Mathematical model  

3 Analytical solutions  
   3.1 Periodic analytical solution  
      3.1.1 Frictionless case  
      3.1.2 With friction  
   3.2 Full analytical solution  

4 Numerical solution method  
   4.1 Linearised model with fixed embayment length  
      4.1.1 Discretisation  
   4.2 Nonlinear model with fixed embayment length  
   4.3 Nonlinear model with a moving boundary  
      4.3.1 With fixed cell averages in space  
      4.3.2 With moving cell averages  
   4.4 Level set method  
      4.4.1 Extension of front velocity  
      4.4.2 Reinitialisation  
      4.4.3 Process overview / Integration with model equations  
   4.5 Time integration  
   4.6 Choice of initial value for new cell  
   4.7 Diagonalisation for a test problem  

5 Numerical results  
   5.1 Comparison of analytical and numerical solution  
      5.1.1 Frictionless case  
      5.1.2 With friction  
   5.2 Numerical results for the moving boundary problem  

6 Conclusion  

7 Recommendations  

References  

A Depth averaging of the model equations  
   A.1 Bottom stress  

B Calculations for case 2  

C MATLAB-code
1 Introduction

In this thesis the water movement in a tidal embayment is considered. Tidal embayments occur in several places along coastal seas. These embayments can have rivers with little discharge, like the Western Scheldt, or have no rivers, like most of the embayments in the Wadden Sea. Tidal embayments are important for economics and ecology; for its economic value and for safety reasons it is important that the land used by people will not be flooded. Also winning of natural resources below the embayments makes it necessary to understand the physics of the embayment and the possible influences of the winning. The circulation of drained heated water from power plants is of interest as well. Ecology is valuable as well. Much live around the the sea depends on the intertidal zone: the birds catch their food at the plates and seals rest on these during low tide. Also the time of drying and flooding is important, since longer drying times will give change for vegetation to grow. Transportation and deposition of pollution is another topic making the study of tidal flows important.

In embayments complex processes take place since there is interaction between water motion and sediment transport; the water motion will cause sediment to be transported. But also the transport of sediment will change the bathymetry of the domain and hence influence the water motion.

Figure 1 shows a satellite image of an embayment in the Wadden Sea near Ameland. From this picture two embayments can be distinguished. Each embayment has its own connection with the North Sea. Below the island just right of the middle there is a so called tidal high, which can be seen as the border between the two embayments, since at that point there is almost no water flow. The embayment on the left hand side stretches to the left side of the connection with the sea beyond the edge of the picture.

1.1 Tides

Tides are excited by celestial forces. The dominant causes for tides on earth are the moon and the sun. The differences in period of the lunar and solar tides cause the spring-neap tide cycle. Depending on the location on earth, the tide can dominantly be diurnal (one high-tide per day)
Table 1: First nine tidal components for Vlieland Haven.

<table>
<thead>
<tr>
<th>Component name</th>
<th>Type</th>
<th>Description</th>
<th>period (h)</th>
<th>factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>M2</td>
<td>SD</td>
<td>principal lunar</td>
<td>12.42</td>
<td>100</td>
</tr>
<tr>
<td>S2</td>
<td>SD</td>
<td>principal solar</td>
<td>12.00</td>
<td>27.58</td>
</tr>
<tr>
<td>N2</td>
<td>SD</td>
<td>larger lunar elliptic</td>
<td>12.66</td>
<td>15.83</td>
</tr>
<tr>
<td>O1</td>
<td>D</td>
<td>principal solar</td>
<td>25.82</td>
<td>12.63</td>
</tr>
<tr>
<td>MU2</td>
<td>SD</td>
<td>lunar evectional</td>
<td>12.87</td>
<td>10.13</td>
</tr>
<tr>
<td>K2</td>
<td>SD</td>
<td>luni-solar</td>
<td>11.97</td>
<td>9.77</td>
</tr>
<tr>
<td>K1</td>
<td>D</td>
<td>luni-solar</td>
<td>23.93</td>
<td>9.04</td>
</tr>
<tr>
<td>L2</td>
<td>SD</td>
<td>smaller lunar elliptic</td>
<td>12.19</td>
<td>8.53</td>
</tr>
<tr>
<td>M6</td>
<td></td>
<td>lunar overtide</td>
<td>4.14</td>
<td>6.01</td>
</tr>
</tbody>
</table>

or semi-diurnal (two high-tides per day) or a combination of both. In The Netherlands the tides are of the semi-diurnal type. Tides can be analysed by decomposition in harmonic components. If \( \zeta \) is the tidal elevation, it can be written as

\[
\zeta = \sum_{i=1}^{N} c_i \sin(\omega_i (t - \varphi_i)),
\]

where \( \omega_i \) is the frequency of the \( i \)th-component, \( \varphi_i \) its phase and \( c_i \) its amplitude. Table 1 gives the nine most dominant tidal components and their amplitudes for the tide in Vlieland Haven, based on a tidal analysis using predicted values from the website [14] at every ten minutes over a two year period. The column factor gives the amplitude of the components, relative to the amplitude of the M2 component, where the factor of M2 is set to 100. Hence \( c_i = \text{factor}_i/100 \times \text{M2 height} \).

Near the shore, due to boundaries of the sea and sloping bottoms, the tides are modified by the bottom friction and nonlinear interactions, resulting in the generation of overtides. The latter have higher frequencies (with different phase). A combination of these tidal constituents results in a large contribution to the transport of sediment. For a more detailed description of tides, see also reference [2], Chapter 2.

To get a correct description of the tidal motion in the shallower areas of tidal embayments, various mathematical techniques have been proposed, see for instance [15]. A well-established mathematical technique, the so-called level set method, has not been applied to the problem of drying and flooding in coastal areas. The level set method is used to solve problems with a moving boundary, for example when the domain consists of two different states (and hence equations), separated by the moving boundary.

The goal of this thesis is to investigate the application of the level set method on the shallow water equations with a moving coastal boundary. The focus will lie on the water motion itself, not on the sediment transport. We consider a rectangular embayment without discharge due to a river. At the land side the position of the water front will change due to fluctuations of the water level. We assume that the water motion in the tidal embayment is only driven by the prescribed tidal elevation at the seaward boundary.

The organisation of the thesis is as follows. In Chapter 2 we describe the mathematical-physical model we use to model the hydrodynamics in a tidal embayment. In Chapter 3 analytical solutions for special cases will be derived and discussed. In Chapter 4 a discretisation is given for several cases, the level set method is discussed and the time integration method will be discussed. Chapter 5 gives a comparison between the analytic and numerical method, for the
cases in which we have an analytical solution. Furthermore results will be presented for the numerical method for other cases and with the level set method being used. The conclusions of the applied analysis are given in Chapter 6 and some recommendations will be made in Chapter 7.
2 Mathematical model

The situation that we consider in this study is sketched in Figure 3. In the topview the domain is given, which is a rectangular domain. At the sea side is an open boundary. In the sideview the sea is on the left hand side and the land on the right hand side. The bottom profile is denoted by $h$, with $h = 0$ by definition at sea side. The height of the mean sea level (MSL) above $h = 0$ is given by $H$. The variation of the sea level around this mean is the variable $\zeta$. We assume that the bottom profile at the land side is such that the water level will never exceed the highest land level within the domain.

To model the water motion in such a tidal embayment, the depth-averaged shallow water equations (for a derivation, see Appendix A) can be used:

$$\frac{\partial \zeta}{\partial t} - \frac{\partial h}{\partial t} + \frac{\partial}{\partial x}((H - h + \zeta)u) + \frac{\partial}{\partial y}((H - h + \zeta)v) = 0, \hspace{1cm} (2.1a)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + g \frac{\partial \zeta}{\partial x} + \frac{\hat{c}_d}{H - h + \zeta} |\tilde{u}|u = 0, \hspace{1cm} (2.1b)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + g \frac{\partial \zeta}{\partial y} + \frac{\hat{c}_d}{H - h + \zeta} |\tilde{u}|v = 0. \hspace{1cm} (2.1c)$$

Here we neglected the viscosity terms and the Coriolis forces, and we assume a fixed bottom profile. Equation (2.1a) is the continuity equation, equations (2.1b) and (2.1c) are the momentum equations in $x$ and $y$-direction. These three variables are the unknowns. The velocity vector $[u, v]^T$ is denoted by $\tilde{u}$. The time is denoted by $t$, and $x$ and $y$ are the spatial coordinates. Further, $H$ is the (tidally averaged) depth of the basin at the entrance, $h$ is the bottom elevation and $\hat{c}_d$ is the friction parameter.

In the continuity equation, the first term is the change in time of the sea surface elevation, the second is the change in time of the bottom elevation, the third term is the spatial derivative in $x$-direction of the longitudinal water flux and the final term is the spatial derivative in $y$-direction of the lateral water flux. In the momentum equation in $x$-direction, the first term is the local acceleration, the second and third term are the advective contributions. The fourth term is the pressure gradient and the fifth term models the bottom friction. The expression $H - h + \zeta$ is the local depth. For the momentum equation in $y$-direction we have the same type of terms.

Neglecting the variations in the lateral direction, we have

$$\frac{\partial \zeta}{\partial t} - \frac{\partial h}{\partial t} + \frac{\partial}{\partial x}((H - h + \zeta)u) = 0, \hspace{1cm} (2.2a)$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + g \frac{\partial \zeta}{\partial x} + \frac{\hat{c}_d}{H - h + \zeta} |u|u = 0. \hspace{1cm} (2.2b)$$

We make these equations dimensionless by introducing the following non-dimensional variables

$$\tilde{x} = \frac{x}{L}, \hspace{1cm} \tilde{t} = \omega t, \hspace{1cm} \tilde{u} = \frac{H}{A \omega L} u,$$

$$\tilde{\zeta} = \frac{\zeta}{A}, \hspace{1cm} \tilde{h} = \frac{h}{A}.$$  

Here $L$ is the tidally averaged length of the basin, $H$ is the depth of the basin at the entrance, $A$ is the amplitude of the tidal constituent at the entrance of the basin with period $\frac{2\pi}{\omega}$ and $\omega$ is the radial frequency of the considered tidal constituent. The scaling of the velocity $u$ has been chosen such that the first and last term in the continuity equation (2.2a) balance. This is
useful since we are interested in long waves. One could also choose the first and third term in the momentum equation (2.2b) to balance. This would be useful if one is interested in gravity waves, but we are not so much interested in gravity waves. Using this scaling in equations (2.2) we obtain the following dimensionless system of equations:

$$\frac{\partial \tilde{\zeta}}{\partial \tilde{t}} - \frac{H}{A} \frac{\partial \tilde{h}}{\partial \tilde{t}} + \frac{\partial}{\partial \tilde{x}} \left( (1 - \tilde{h} + \frac{A}{H} \tilde{\zeta}) \tilde{u} \right) = 0,$$

(2.3a)

$$\frac{\partial \tilde{u}}{\partial \tilde{t}} + \frac{A}{H} \tilde{u} \frac{\partial \tilde{u}}{\partial \tilde{x}} + \frac{gh}{\omega^2 L^2} \frac{\partial \tilde{\zeta}}{\partial \tilde{x}} + \frac{AL}{H^2} \frac{\tilde{c}_d}{1 - \tilde{h} + \frac{A}{H} \tilde{\zeta}} |\tilde{u}| \tilde{u} = 0.$$  

(2.3b)

After defining the dimensionless parameters $\varepsilon = \frac{A}{H}$ (the ratio of the tidal amplitude and the water depth), $\lambda = \frac{\omega L}{\sqrt{gH}}$ (the ratio of the embayment length $L$ and the frictionless tidal wave-length $\frac{\sqrt{gH}}{\omega}$), $c_\ast^d = \frac{AL}{H^2} \tilde{c}_d$ and dropping the tildes, we get

$$\frac{\partial \zeta}{\partial t} - \frac{1}{\varepsilon} \frac{\partial h}{\partial t} + \frac{\partial}{\partial x} ((1 - h + \varepsilon \zeta) u) = 0,$$

(2.4a)

$$\frac{\partial u}{\partial t} + \varepsilon u \frac{\partial u}{\partial x} + \frac{1}{\lambda^2 \partial x} \frac{\partial \zeta}{\partial x} + \frac{c_\ast^d}{1 - h + \varepsilon \zeta} |u| u = 0.$$  

(2.4b)

In Table 2 a summary is given of the dimensionless parameters.
<table>
<thead>
<tr>
<th>parameter</th>
<th>definition</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε</td>
<td>$\frac{A}{H}$</td>
<td>0.1</td>
</tr>
<tr>
<td>λ</td>
<td>$\frac{\omega L}{\sqrt{g H}}$</td>
<td>0.0141</td>
</tr>
<tr>
<td>$\tilde{c}_d$</td>
<td>$\frac{A L}{H^2} \tilde{c}_d$</td>
<td>0.025</td>
</tr>
<tr>
<td>$r$</td>
<td>$\frac{8}{3\pi} \tilde{c}_d = \frac{8}{3\pi} \frac{A L}{H^2} \tilde{c}_d$</td>
<td>0.0212</td>
</tr>
</tbody>
</table>

Table 2: Definition of parameters and their typical values.

As boundary conditions we prescribe at the seaward boundary ($x = 0$)

$$\zeta(0, t) = \zeta_0(t),$$

i.e., $\zeta$ has a Dirichlet boundary condition and $u$ is free. Most of the time we will use $\zeta_0 = \sin(t)$. For the landward boundary we have a fixed boundary at $x = 1$ where $\zeta$ is free and we have a homogeneous Dirichlet condition for $u$,

$$u(1, t) = 0.$$  

In other cases we will have a moving landward boundary condition. In such a case the boundary is located where $1 - h + \varepsilon \zeta = 0$ and the condition is

$$\left(1 - h(x) + \varepsilon \zeta(x, t)\right) u(x, t) = 0.$$  

This actually requires that $u(x, t)$ is finite on the landward boundary. We use the following initial conditions, stating that the variables $\zeta$ and $u$ are given at $t = 0$,

$$\zeta(x, 0) = f_1(x), \quad u(x, 0) = f_2(x),$$

with $f_1(x)$ and $f_2(x)$ prescribed functions of the spatial variable.

We want to linearise the nonlinear dependency of the bottom friction term on the velocity, $\tilde{c}_d^* |u|u = ru$. We want to choose $r$ such, that the dissipation of energy over a tidal period is the same for the original nonlinear friction parametrisation and for the model with the linearised friction term. This process used to determine $r$ is called the Lorentz linearisation, see for instance Chapter 5 in [3]. Using this process, one find that $r = \frac{8}{3\pi} \tilde{c}_d^*$, and hence the friction term reads as

$$\frac{AL}{H^2} \tilde{c}_d \frac{|u|u}{1 - h} u = \frac{r}{1 - h} u.$$  

We will not consider any morphodynamical changes either, since we take the bottom fixed in time ($\frac{\partial h}{\partial t} = 0$) throughout the thesis. The resulting dimensionless system of equations with linearised friction reads as

$$\frac{\partial \zeta}{\partial t} + \frac{\partial u}{\partial x} \left( (1 - h + \varepsilon \zeta) u \right) = 0,$$

$$\frac{\partial u}{\partial t} + \varepsilon u \frac{\partial u}{\partial x} + \frac{1}{\lambda^2} \frac{\partial \zeta}{\partial x} + \frac{r}{1 - h + \varepsilon \zeta} u = 0.$$  

In the following chapter we choose the parameters such that we have a linear system and obtain analytical solutions.
3 Analytical solutions

In this chapter, we consider the linear equations in an embayment with fixed length in order to derive analytical solutions. To be able to derive the analytical solution, we assume a spatially uniform bottom profile. Since $h(x = 0) = 0$ by definition we must now have $h(x) \equiv 0$. First we consider a periodic equilibrium solution and further on we will consider the full analytical solution including initial conditions and decaying solutions towards the periodic solution. We take $\varepsilon = 0$ in system (2.10) and obtain the following linear dimensionless system

\[
\begin{align*}
\frac{\partial \zeta}{\partial t} + \frac{\partial u}{\partial x} &= 0, \\
\frac{\partial u}{\partial t} + \frac{1}{\lambda^2} \frac{\partial \zeta}{\partial x} + ru &= 0.
\end{align*}
\]

(3.1a) \hspace{1cm} (3.1b)

As boundary conditions we have conditions (2.5) and (2.6)

$$\zeta(0, t) = \sin(t), \quad \text{and} \quad u(1, t) = 0.$$  

(3.2)

We use the following initial conditions in the transient solution

$$\zeta(x, 0) = f_1(x), \quad u(x, 0) = f_2(x),$$  

(3.3)

with $f_1(x)$ and $f_2(x)$ prescribed functions of the spatial variable.

The solution of this system of equations can been regarded as the sum of a periodic solution and other transient solutions. Hence, the total solution will decay in the limit for $t \to \infty$ towards the periodic solution, due to friction. We will discuss the two types of solutions separately.

3.1 Periodic analytical solution

As a first step, we will look for a periodic solution. This periodicity is due to the boundary condition for $\zeta$ at $x = 0$, which is periodic in time. As the solution is periodic, we will not prescribe initial conditions for this case, as they most likely do not correspond with the periodic solution. Instead, we use periodic conditions in time as initial condition, i.e.

$$\zeta(x, t + 2\pi) = \zeta(x, t), \quad u(x, t + 2\pi) = u(x, t).$$

First we consider the case without friction ($r = 0$), in Section 3.1.2 the case with friction ($r > 0$) will be discussed.

3.1.1 Frictionless case

Neglecting the bottom friction ($r = 0$), the system of equations (3.1) reduces to

\[
\begin{align*}
\frac{\partial \zeta}{\partial t} + \frac{\partial u}{\partial x} &= 0, \\
\frac{\partial u}{\partial t} + \frac{1}{\lambda^2} \frac{\partial \zeta}{\partial x} &= 0,
\end{align*}
\]

(3.4a) \hspace{1cm} (3.4b)

with boundary conditions

$$\zeta(0, t) = \sin(t), \quad u(1, t) = 0.$$  

(3.5)
To solve this system analytically we first differentiate the momentum equation (3.4b) with respect to \( t \) and the continuity equation (3.4a) with respect to \( x \). Combing these expressions results in
\[
\frac{\partial^2 u}{\partial t^2} - \frac{1}{\lambda^2} \frac{\partial^2 u}{\partial x^2} = 0. \tag{3.6}
\]

We try to find a solution in the form \( u(x,t) = T(t) \cdot y(x) \), where \( T(t) = a \cos(t) + b \sin(t) \).

So we assume we have a periodic solution and we want to find \( a, b \) and \( y(x) \). Substitution in equation (3.6) and division by \( T \) gives
\[
- y + \frac{1}{\lambda^2} \frac{d^2 y}{dx^2} = 0. \tag{3.7}
\]

The general solution is given by
\[
y(x) = c_1 e^{i\lambda x} + c_2 e^{-i\lambda x}, \tag{3.8}
\]

Using the boundary condition for \( u \), requires \( y(1) = 0 \):
\[
0 = y(1) = c_1 e^{i\lambda} + c_2 e^{-i\lambda}, \tag{3.9}
\]
\[
c_2 = -c_1 e^{2i\lambda}. \tag{3.10}
\]

This gives
\[
y(x) = c_1 \left( e^{i\lambda x} - e^{2i\lambda} e^{-i\lambda x} \right), \tag{3.11}
\]
or
\[
y(x) = c_1 e^{i\lambda} \left( e^{-i\lambda(1-x)} - e^{i\lambda(1-x)} \right). \tag{3.12}
\]

Using this expression we find that
\[
u(x,t) = (a \sin t + b \cos t) c_1 e^{i\lambda} \left( e^{-i\lambda(1-x)} - e^{i\lambda(1-x)} \right). \tag{3.13}
\]

This can be rewritten as
\[
u(x,t) = \left( \hat{a} \sin t + \hat{b} \cos t \right) \sin(\lambda(1-x)) \tag{3.14}
\]
where \( \hat{a} = -2iac_1 e^{i\lambda} \) and \( \hat{b} = -2ibc_1 e^{i\lambda} \). Now we can use the continuity equation (3.4a) to determine \( \zeta, \hat{a} \) and \( \hat{b} \).

\[
\frac{\partial \zeta}{\partial t} = -\frac{\partial u}{\partial x} = \lambda \left( \hat{a} \sin t + \hat{b} \cos t \right) \cos(\lambda(1-x)). \tag{3.15}
\]

Integrating equation (3.15) with respect to time, gives
\[
\zeta(x,t) = \lambda \left( -\hat{a} \cos t + \hat{b} \sin t \right) \cos(\lambda(1-x)) + f(x). \tag{3.16}
\]

From the momentum equation (3.4b) it follows that \( \frac{df(x)}{dx} \equiv 0 \), so \( f(x) \) is a constant, say \( f(x) = cf \). The boundary condition reads as
\[
\zeta(0,t) = \sin t, \tag{3.17}
\]
so we have
\[
\sin t = \lambda \left( -\hat{a} \cos t + \hat{b} \sin t \right) \cot (\lambda) + c_f. \tag{3.18}
\]
Since the time average over one period of the left hand side is zero, also the right hand side average should be zero, so \( c_f = 0 \). For the other unknown constants, it follows that
\[
\hat{a} = 0, \quad \hat{b} = (\lambda \cot (\lambda))^{-1}. \tag{3.19}
\]
Hence the analytical solution for the sea surface elevation is given by
\[
\zeta(x, t) = \sin t \frac{\cos (\lambda (1 - x))}{\cos (\lambda)}, \tag{3.20}
\]
and for the velocity by
\[
u(x, t) = \frac{\cos t \sin (\lambda (1 - x))}{\lambda} \cdot \frac{\cos (\lambda)}{\cos (\lambda)}. \tag{3.21}
\]

### 3.1.2 With friction

In case of the linear dimensionless equations with friction, we have the following system of equations:

\[
\begin{align*}
\frac{\partial \zeta}{\partial t} + \frac{\partial u}{\partial x} & = 0 \tag{3.22a} \\
\frac{\partial u}{\partial t} + \frac{1}{\lambda^2} \frac{\partial \zeta}{\partial x} + r u & = 0, \tag{3.22b}
\end{align*}
\]

with boundary conditions
\[
\zeta(0, t) = \sin (t), \quad u(t, 1) = 0. \tag{3.23}
\]

To solve this system analytically, we derive an equation for the unknown velocity \( u \) only. To this end we differentiate the momentum equation (3.22b) with respect to \( t \) and the continuity equation (3.22a) with respect to \( x \). Combing these expressions gives
\[
\frac{\partial^2 u}{\partial t^2} - \frac{1}{\lambda^2} \frac{\partial^2 u}{\partial x^2} + r \frac{\partial u}{\partial t} = 0. \tag{3.24}
\]

Assume we have a periodic solution, and we can write \( u(x, t) = \Re (e^{i \sigma t} y(x)) \). From the continuity equation we find that \( \zeta \) has the same period \( \sigma \) as \( u \), and from the boundary condition for \( \zeta \) it follows that \( \sigma = 1 \). We first write \( u(x, t) = e^{i t} y(x) \) for simplicity, and at the end we take the real part. Then we have
\[
- e^{i t} y(x) - \frac{1}{\lambda^2} e^{i t} \frac{d^2 y}{d x^2} + ir e^{i t} y(x) = 0. \tag{3.25}
\]

Division by \( e^{i t} \) and rearranging equation (3.25) gives
\[
\frac{d^2 y}{d x^2} + \lambda^2 (1 - ir) y = 0. \tag{3.26}
\]

The general solution for this equation is given by \( y(x) = c_1 e^{\mu x} + c_2 e^{-\mu x} \), where \( \mu \) satisfies
\[
\mu^2 = \lambda^2 (1 - ir). \tag{3.27}
\]
Hence $\mu$ is given by
\[
\mu = \pm \lambda \frac{\sqrt{2}}{2} \left( \sqrt{1 + \sqrt{1 + r^2} - i\sqrt{1 + \sqrt{1 + r^2}}} \right),
\] (3.28)
where we used that
\[
\sqrt{\alpha + i\beta} = \pm \frac{\sqrt{2}}{2} \left( \sqrt{\alpha + \sqrt{\alpha^2 + \beta^2} + i\frac{\beta}{|\beta|} \sqrt{-\alpha + \sqrt{\alpha^2 + \beta^2}}} \right).
\]

Using the landward boundary condition we have
\[
y(1) = 0 \Rightarrow c_2 = -c_1 e^{2i\mu}.
\]
So for the velocity $u$ we have
\[
u(x, t) = c_1 e^{it} \left( e^{i\mu x} - e^{2i\mu} e^{-i\mu x} \right)
\] (3.29)
\[
u(x, t) = -c_1 e^{it} \left( -e^{-i\mu(1-x)} + e^{i\mu(1-x)} \right)
\] (3.30)
\[
u(x, t) = -2ic_1 e^{i\mu} e^{it} \sin(\mu(1-x))
\] (3.31)
\[
u(x, t) = c_3 e^{it} \sin(\mu(1-x)),
\] (3.32)
where $c_3 = -2ic_1 e^{i\mu}$. Now we can determine the water level elevation $\zeta$ by using equation (3.22a)
\[
\frac{\partial \zeta}{\partial t} = -\frac{\partial u}{\partial x} = c_3 e^{it} \cos(\mu(1-x)).
\] (3.33)
Integration with respect to time gives
\[
\zeta(x, t) = -ic_3 \mu e^{it} \cos(\mu(1-x)).
\] (3.34)
We use the seaward boundary condition $\zeta(0, t) = \sin(t)$ to determine the constant $c_3$:
\[
-ic_3 \mu e^{it} \cos(\mu) = \sin(t).
\] (3.35)
We only have to satisfy the boundary condition for the real part of $\zeta$, as mentioned before. So we satisfy the boundary condition if we take
\[
-ic_3 \mu \cos(\mu) = -i,
\] (3.36)
or equivalently
\[
c_3 = \frac{1}{\mu \cos(\mu)}.
\] (3.37)
As final analytical expressions for the sea surface elevation $\zeta$ and the velocity $u$ we have
\[
\zeta(x, t) = \text{Re} \left\{ -i e^{it} \frac{\cos(\mu(1-x))}{\cos(\mu)} \right\}
\] (3.38a)
\[
u(x, t) = \text{Re} \left\{ \frac{1}{\mu} e^{it} \frac{\sin(\mu(1-x))}{\cos(\mu)} \right\},
\] (3.38b)
where we take the real part of the solutions as mentioned above. Note that if we take $r = 0$ we have the same solution as we had before.
In the formula for $\zeta$ (3.38a) we can identify $\frac{1}{\cos(\mu)}$ as the complex wave amplitude, $\text{Re}(\mu)$ as the wave number and $\text{Im}(\mu)$ as one over the friction scale.
3.2 Full analytical solution

In order to apply the method of separation of variables, we make system (3.1) homogeneous by subtracting the periodic solution; define

\[ z(x, t) = \zeta(x, t) - \zeta_{\text{per}}(x, t), \]  
\[ v(x, t) = u(x, t) - u_{\text{per}}(x, t), \]  

where \( \zeta_{\text{per}} \) and \( u_{\text{per}} \) are the solutions found in the previous section. Then we find as homogeneous system for \( z \) and \( v \)

\[
\begin{align*}
\frac{\partial z}{\partial t} + \frac{\partial v}{\partial x} &= 0, \quad (3.41a) \\
\frac{\partial v}{\partial t} + \frac{1}{\lambda^2} \frac{\partial z}{\partial x} + rv &= 0, \quad (3.41b)
\end{align*}
\]

with boundary conditions

\[ z(0, t) = 0, \quad v(1, t) = 0. \]  

(3.42)

From the initial conditions 3.3, the initial conditions are given by

\[ z(x, 0) = f_1(x) - \zeta_{\text{per}}(x, 0) := f_3(x), \quad v(x, 0) = f_2(x) - u_{\text{per}}(x, 0) := f_4(x). \]  

(3.43)

To obtain a single equation for \( v \), we differentiate equation (3.41b) with respect to \( t \) and equation (3.41a) with respect to \( x \) and combine the two results:

\[
\frac{d^2 v}{dt^2} - \frac{1}{\lambda^2} \frac{d^2 v}{dx^2} + r \frac{dv}{dt} = 0.
\]

We separate the variables of \( v \) by writing \( v(x, t) = T(t) y(x) \) and look for non-trivial solutions. Substitution of this expression in the previous equation, dividing by \( T \) and \( y \), gives

\[
\frac{\frac{d^2 T}{dt^2} + r \frac{dT}{dt}}{T} - \frac{1}{\lambda^2} \frac{d^2 y}{dx^2} = 0.
\]

Since the first term is a function of \( t \) only and the second term is a function of \( x \) only, both should be constant. So we have

\[
\frac{\frac{d^2 T}{dt^2} + r \frac{dT}{dt}}{T} = \frac{1}{\lambda^2} \frac{d^2 y}{dx^2} = K.
\]

(3.44)

Here \( K \in \mathbb{R} \), since we can write the boundary value problem for \( y(x) \) as a Sturm-Liouville eigenvalue problem with the boundary conditions that follow. As boundary condition, we had \( v(1, t) = 0 \). As \( T(t)y(1) = 0 \) this implies \( y(1) = 0 \) since we look for non-trivial solutions. For the other boundary condition we have \( z(0, t) \). We want to translate this into a condition for \( y \) (or \( T \)). From equation (3.41a), we find that

\[
z(x, t) = - \int \frac{\partial}{\partial x} \left( T(t)y(x) \right) dt = - \frac{dy}{dx} \int T \ dt.
\]

(3.45)
So \(z(0, t) = 0\) gives \(-\frac{dy}{dx}\bigg|_{x=0} \int T dt = 0\). If \(\int T dt = 0\), then \(T \equiv 0\) and we have the trivial solution, hence \(\frac{dy}{dx}\bigg|_{x=0} = 0\). Now we solve equation (3.44) for \(y\) first. We have

\[
\frac{d^2y}{dx^2} = K\lambda^2 y,
\]

with as general solution (when \(K\) is nonzero)

\[
y(x) = c_1 e^{\lambda \sqrt{K} x} + c_2 e^{-\lambda \sqrt{K} x}.\tag{3.46}
\]

We have to adjust this solution to the boundary conditions \(y(1) = 0\) and \(\frac{dy}{dx}\bigg|_{x=0} = 0\). The first gives \(c_2 = -c_1 e^{2\lambda \sqrt{K}}\), so

\[
y(x) = c_1 \left(e^{\lambda \sqrt{K} x} - e^{2\lambda \sqrt{K}} e^{-\lambda \sqrt{K} x}\right),\tag{3.47}
\]

\[
\frac{dy}{dx}(x) = \lambda \sqrt{K} c_1 \left(e^{\lambda \sqrt{K} x} + e^{2\lambda \sqrt{K}} e^{-\lambda \sqrt{K} x}\right).\tag{3.48}
\]

The second boundary condition gives \(\frac{dy}{dx}(0) = \lambda \sqrt{K} c_1 \left(1 + e^{2\lambda \sqrt{K}} e\right) = 0\). Since \(c_1 = 0\) would lead to the trivial solution, we should have

\[
\lambda \sqrt{K} \left(1 + e^{2\lambda \sqrt{K}} e\right) = 0.\tag{3.49}
\]

This equation determines the possible values for the separation constant \(K\), also known as the eigenvalues. If \(K=0\) then we have \(y(x) = \tilde{c}_1 + \tilde{c}_2 x\). The first boundary condition gives that \(\tilde{c}_1 + \tilde{c}_2 = 0\), whereas the second boundary condition gives \(\tilde{c}_2 = 0\), which together gives the trivial solution for \(y(x)\). So \(K \neq 0\). As condition for \(K\) we now have

\[
1 + e^{2\lambda \sqrt{K}} = 0.
\]

This gives \(2\lambda \sqrt{K} = i\pi + 2ik\pi\), where \(k \in \mathbb{Z}\), or equivalently \(\sqrt{K} = (k + \frac{1}{2}) \frac{i\pi}{2\lambda}\). We observe from the general solution in equation (3.46) that the replacement of \(\sqrt{K}\) by \(-\sqrt{K}\) does not make a difference. For negative values of \(k\) we have the same absolute value of \(\sqrt{K}\) as for taking \(-k - 1\), so we can restrict ourselves to \(k \in \mathbb{N}\). The solution is now given by

\[
y_k(x) = c_{1k} \left(e^{i\pi (k + \frac{1}{2}) x} - e^{2i\pi (k + \frac{1}{2}) e^{-i\pi (k + \frac{1}{2}) x}}\right),\tag{3.50}
\]

which we can rewrite as

\[
y_k(x) = -2ic_{1k} e^{i\pi (k + \frac{1}{2})} \sin \left(\pi \left(k + \frac{1}{2}\right)(1 - x)\right) = -2ic_{1k} (-1)^k \sin \left(\pi \left(k + \frac{1}{2}\right)(1 - x)\right).\tag{3.51}
\]

Now we found the space-dependent part of the solution \(y(x)\), we want to solve the time-dependent part \(T(t)\). From equation (3.44) we have for the time dependent part

\[
\frac{d^2T}{dt^2} + r \frac{dT}{dt} - KT = 0,
\]

12
with $K = -(k + \frac{1}{2})^2 \frac{\pi^2}{\lambda^2}$. Substitution of the test function $t = e^{st}$ we get the characteristic equation $s^2 + rs - K = 0$, which has as solutions
\begin{equation}
s_{1,2} = -\frac{r \pm \sqrt{r^2 + 4K}}{2}.
\end{equation}

The general solution is given by
\begin{equation}
T(t) = e^{-\frac{r}{2}t} \left( c_3 e^{\frac{1}{2} \sqrt{r^2 + 4K} t} + c_4 e^{-\frac{1}{2} \sqrt{r^2 + 4K} t} \right).
\end{equation}

Since $K < 0$ we have $\text{Re} \left( \sqrt{r^2 + 4K} \right) < |r|$, which implies that there is always damping in the system whenever $r > 0$. For realistic values of the parameters we can have the cases that $r^2 > 4K$ and $r^2 < 4K$. In order to treat both cases, define
\begin{equation}
k_1 = \min \left\{ k : r^2 - 4 \left( k + \frac{1}{2} \right)^2 \frac{\pi^2}{\lambda^2} < 0, \ k \in \mathbb{N} \right\}.
\end{equation}

The function $T(t)$ can be rewritten as
\begin{equation}
T(t) = \sum_{k=0}^{k_1-1} e^{-\frac{r}{2}t} \left( \tilde{c}_5 \cosh \left( \frac{t}{2} \sqrt{r^2 + 4K(k)} \right) + \tilde{c}_6 \sinh \left( \frac{t}{2} \sqrt{r^2 + 4K(k)} \right) \right) + \sum_{k=k_1}^{\infty} e^{-\frac{r}{2}t} \left( \tilde{c}_5 \cos \left( \frac{t}{2} \sqrt{-4K(k) - r^2} \right) + \tilde{c}_6 \sin \left( \frac{t}{2} \sqrt{-4K(k) - r^2} \right) \right),
\end{equation}

where $K(k) = -(k + \frac{1}{2})^2 \frac{\pi^2}{\lambda^2}$. The function $v(x, t)$ is given by
\begin{equation}
v(x, t) = \sum_{k=0}^{k_1-1} e^{-\frac{r}{2}t} \sin \left( \frac{\pi}{2} \right) (1-x) \left( c_{5k} \cosh(A(k)t) + c_{6k} \sinh(A(k)t) \right) + \sum_{k=k_1}^{\infty} e^{-\frac{r}{2}t} \sin \left( \frac{\pi}{2} \right) (1-x) \left( c_{5k} \cos(B(k)t) + c_{6k} \sin(B(k)t) \right),
\end{equation}

where
\begin{align*}
c_{5k} &= -2i c_1 (-1)^k (c_3 + c_4) \quad \forall k \in \mathbb{N}, \\
c_{6k} &= \begin{cases} 
-2i c_1 (-1)^k (c_3 - c_4) & \text{if } k < k_1 \\
2c_1 (-1)^k (c_3 - c_4) & \text{if } k \geq k_1.
\end{cases} \\
A(k) &= \frac{1}{2} \sqrt{r^2 - 4 \left( k + \frac{1}{2} \right)^2 \frac{\pi^2}{\lambda^2}}, \\
B(k) &= \frac{1}{2} \sqrt{4 \left( k + \frac{1}{2} \right)^2 \frac{\pi^2}{\lambda^2} - r^2}.
\end{align*}

From this we have
\begin{equation}
v(x, 0) = \sum_{k=0}^{\infty} c_{5k} \sin \left( \pi \left( k + \frac{1}{2} \right) (1-x) \right).
\end{equation}

As the next step we determine $z(x, t)$. Using equation (3.55), and observing that the integration constant should equal zero, we have that $z(x, t) = - \int \frac{\partial v}{\partial x} \, dt$. The constant is zero, since if we
Using orthogonal properties of the sine and cosine functions, we have a single function \( z(x, t) = z_e(x) \), then from the boundary condition we have that \( z_e(0) = 0 \). From the continuity equation (3.41a) it follows then that \( v(x, t) = v(t) \), and the boundary condition gives \( v \equiv 0 \). This leads to a contradiction in the momentum equation (3.41b).

Using the expression for \( v \), this results in

\[
z(x, t) = \sum_{k=0}^{k_1-1} \frac{1}{2} \int -\cos\left(\pi \left( k + \frac{1}{2} \right) (1-x) \right) \pi \left( k + \frac{1}{2} \right) e^{-\frac{\lambda^2}{t}} \left( c_{5k} \cosh(A(k)t) + c_{6k} \sinh(A(k)t) \right) dt \\
+ \sum_{k=k_1}^{\infty} \frac{1}{2} \int -\cos\left(\pi \left( k + \frac{1}{2} \right) (1-x) \right) \pi \left( k + \frac{1}{2} \right) e^{-\frac{\lambda^2}{t}} \left( c_{5k} \cosh(B(k)t) + c_{6k} \sinh(B(k)t) \right) dt \\
= \sum_{k=0}^{k_1-1} \cos\left(\pi \left( k + \frac{1}{2} \right) (1-x) \right) \frac{-\lambda^2}{2\pi(k + \frac{1}{2})} e^{-\frac{\lambda^2}{t}} \left( c_{5k} \left( r \cosh(A(k)t) + 2A(k) \sinh(A(k)t) \right) + c_{6k} \left( r \sinh(A(k)t) + 2A(k) \cosh(A(k)t) \right) \right) \\
+ \sum_{k=k_1}^{\infty} \cos\left(\pi \left( k + \frac{1}{2} \right) (1-x) \right) \frac{-\lambda^2}{2\pi(k + \frac{1}{2})} e^{-\frac{\lambda^2}{t}} \left( c_{5k} \left( r \cosh(B(k)t) - 2B(k) \sinh(B(k)t) \right) + c_{6k} \left( r \sinh(B(k)t) + 2B(k) \cosh(B(k)t) \right) \right),
\]

where we have used integration by parts to determine the integral of the time dependent part and the equalities

\[
\frac{-2\pi(k + \frac{1}{2})}{r^2 - 4A^2(k)} = \frac{-2\pi(k + \frac{1}{2})}{r^2 + 4B^2(k)} = \frac{-\lambda^2}{2\pi(k + \frac{1}{2})}.
\]

At \( t = 0 \) we have

\[
z(x, 0) = \sum_{k=0}^{k_1-1} \cos\left(\pi \left( k + \frac{1}{2} \right) (1-x) \right) \frac{-\lambda^2}{2\pi(k + \frac{1}{2})} \left( rc_{5k} + 2A(k)c_{6k} \right) \\
+ \sum_{k=k_1}^{\infty} \cos\left(\pi \left( k + \frac{1}{2} \right) (1-x) \right) \frac{-\lambda^2}{2\pi(k + \frac{1}{2})} \left( rc_{5k} + 2B(k)c_{6k} \right).
\]

Using the boundary conditions we can now determine the constants

\[
f_4(x) = v(x, 0) = \sum_{k=0}^{\infty} c_{5k} \sin\left(\pi \left( k + \frac{1}{2} \right) (1-x) \right),
\]

\[
f_3(x) = z(x, 0) = \sum_{k=0}^{k_1-1} \cos\left(\pi \left( k + \frac{1}{2} \right) (1-x) \right) \frac{-\lambda^2}{2\pi(k + \frac{1}{2})} \left( rc_{5k} + 2A(k)c_{6k} \right) \\
+ \sum_{k=k_1}^{\infty} \cos\left(\pi \left( k + \frac{1}{2} \right) (1-x) \right) \frac{-\lambda^2}{2\pi(k + \frac{1}{2})} \left( rc_{5k} + 2B(k)c_{6k} \right).
\]

Using orthogonal properties of the sine and cosine functions,

\[
\int_0^1 \sin\left(\pi \left( k + \frac{1}{2} \right) (1-x) \right) \sin\left(\pi \left( m + \frac{1}{2} \right) (1-x) \right) dx = \begin{cases} 
0 & \text{if } k \neq m \\
\frac{1}{2} & \text{if } k = m,
\end{cases}
\]

\[
\int_0^1 \cos\left(\pi \left( k + \frac{1}{2} \right) (1-x) \right) \cos\left(\pi \left( m + \frac{1}{2} \right) (1-x) \right) dx = \begin{cases} 
0 & \text{if } k \neq m \\
\frac{1}{2} & \text{if } k = m,
\end{cases}
\]

14
we find the formulas

\[ 2 \int_0^1 f_4(x) \sin \left( \pi \left( m + \frac{1}{2} \right) (1 - x) \right) \, dx = c_{5m} \] (3.58)

\[ 2 \int_0^1 f_3(x) \cos \left( \pi \left( m + \frac{1}{2} \right) (1 - x) \right) \, dx = \begin{cases} \frac{-\lambda^2}{2\pi(m + \frac{1}{2})} (rc_{5m} + 2A(m)c_{6m}) & \text{if } m < k_1 \\ \frac{-\lambda^2}{2\pi(m + \frac{1}{2})} (rc_{5m} + 2B(m)c_{6m}) & \text{if } m \geq k_1 \end{cases} \] (3.59)

\[ c_{5m} = 2 \int_0^1 f_4(x) \sin \left( \pi \left( m + \frac{1}{2} \right) (1 - x) \right) \, dx \] (3.60)

\[ c_{6m} = \begin{cases} \left( f_0^1 f_3(x) \cos \left( \pi \left( m + \frac{1}{2} \right) (1 - x) \right) \, dx \frac{4\pi(m + \frac{1}{2})}{\lambda^2} - rc_{5m} \right) \frac{1}{2A(m)} & \text{if } m < k_1 \\ \left( f_0^1 f_3(x) \cos \left( \pi \left( m + \frac{1}{2} \right) (1 - x) \right) \, dx \frac{4\pi(m + \frac{1}{2})}{\lambda^2} - rc_{5m} \right) \frac{1}{2B(m)} & \text{if } m \geq k_1 \end{cases} \] (3.61)

Figure 4: Plot of \( u(x, t) \) as function of \( t \) for several locations, small \( r \).

The part of the solution with \( k \leq k_1 \) decays very quickly without oscillating, whereas the other part oscillates with decaying amplitude.

For the case without friction (\( r = 0 \)), the formulas remains valid, (with \( k_1 = 0 \), and can be summarised as

\[ v(x, t) = \sum_{k=0}^{\infty} \sin \left( \pi \left( k + \frac{1}{2} \right) (1 - x) \right) \left( c_{5k} \cos \left( \frac{\pi}{\lambda} \left( k + \frac{1}{2} \right) t \right) + c_{6k} \sin \left( \frac{\pi}{\lambda} \left( k + \frac{1}{2} \right) t \right) \right), \]

\[ z(x, t) = \sum_{k=0}^{\infty} \cos \left( \pi \left( k + \frac{1}{2} \right) (1 - x) \right) \lambda \left( c_{5k} \sin \left( \frac{\pi}{\lambda} \left( k + \frac{1}{2} \right) t \right) - c_{6k} \cos \left( \frac{\pi}{\lambda} \left( k + \frac{1}{2} \right) t \right) \right), \]
(a) $z = \zeta$ damp

(b) $v = u$ damp

Figure 5: Plot of decaying solution for $r$ from simulation and two initial conditions, straight line and parabola.

Figure 6: Analytical solution for $\zeta$ for short time, several locations.
where

\[ c_{5m} = 2 \int_0^1 f_4(x) \sin \left( \pi \left( m + \frac{1}{2} \right) (1 - x) \right) \, dx \quad \text{for } m \in \mathbb{N} \]

\[ c_{6m} = \frac{-2}{\lambda} \int_0^1 f_3(x) \cos \left( \pi \left( m + \frac{1}{2} \right) (1 - x) \right) \, dx \quad \text{for } m \in \mathbb{N}. \]

In this case there is no damping as we can see since the time behaviour is fully periodic for each eigenmode.
4 Numerical solution method

In this chapter we will derive the numerical method to obtain the solution. First we will give the finite volume discretisation for the embayment with a fixed length and with a moving boundary. In Section 4.4 we will present the level set method, and we will discuss the time integration method and present an overview of the process.

4.1 Linearised model with fixed embayment length

4.1.1 Discretisation

We want to solve the cross-sectionally averaged linear equations (3.1) numerically. To this end we use a finite volume discretisation in space and a forward Euler or modified Euler method in time. For the space domain \( x \in [0, L] \) (\( L = 1 \), in our dimensionless case) we use a uniform grid with \( N \) points, \( \Delta x = \frac{1}{N-1} \), \( x_1 = 0 \), \( x_N = L \) and \( x_i = (i-1)\Delta x \), for \( i = 1 \ldots N \). See also Figure 7. In this figure we see the gridpoints \( x_i \) with \( x_1 = 0 \) and \( x_N = L \). As an example cell \( C_3 \) of length \( \Delta x \) centered around gridpoint \( x_3 \) is indicated.

We integrate each equation over a cell, for an internal cell \( i \) we have

\[
\int_{C_i} \left[ \frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}((1 - h)u) \right] \, dx = 0. \tag{4.1}
\]

where \( C_i = [x_i - \frac{1}{2}\Delta x, x_i + \frac{1}{2}\Delta x] \). Assuming we can interchange the integration and differentiation we can write

\[
\frac{d}{dt} \int_{C_i} \zeta \, dx + \int_{C_i} \frac{\partial}{\partial x}((1 - h)u) \, dx = 0. \tag{4.2}
\]

Now define the cell averaged quantities

\[
\zeta_i = \frac{1}{\Delta x} \int_{C_i} \zeta(x, t) \, dx, \quad u_i = \frac{1}{\Delta x} \int_{C_i} u(x, t) \, dx, \tag{4.3}
\]

where \( C_i = [x_i - \frac{1}{2}\Delta x, x_i + \frac{1}{2}\Delta x] \). Then we have

\[
\Delta x \frac{d\zeta_i}{dt} + \int_{C_i} \frac{\partial}{\partial x}((1 - h)u) \, dx = 0, \tag{4.4}
\]

\[
\Delta x \frac{d\zeta_i}{dt} + [(1 - h)u]_{x_i + \frac{1}{2}\Delta x}^{x_i + \frac{1}{2}\Delta x} \, dx = 0. \tag{4.5}
\]

Figure 7: Grid layout for a fixed domain.
We use linear interpolation from the cell values,

\[ 0 = \Delta x \frac{d\zeta_i}{dt} + [(1 - h)u_{i+\frac{1}{2}} + \frac{1}{2}(1 - h_{i+1}) u_{i+1} + (1 - h_i) u_{i}] \]

\[ - \frac{1}{2}(1 - h_{i-1}) u_{i-1} + (1 - h_i) u_i + (1 - h_{i+1}) u_{i+1} \]

\[ = \Delta x \frac{d\zeta_i}{dt} + \frac{1}{2}(1 - h_{i+1}) u_{i+1} + (1 - h_i) u_i + \frac{1}{2}(1 - h_{i+1}) u_{i+1} - (1 - h_{i-1}) u_{i-1}. \] (4.6)

For the momentum equation (3.1b) we get

\[ \frac{d}{dt} \int_{C_i} u \, dx + \int_{C_i} \left[ \frac{1}{\lambda^2} \frac{\partial \zeta}{\partial x} + \frac{r}{1 - h} u \right] \, dx = 0. \] (4.7)

\[ \Delta x \frac{du_i}{dt} + \frac{1}{\lambda^2} \int_{C_i} \frac{\partial \zeta}{\partial x} \, dx + \Delta x \frac{r}{1 - h_i} u_i = 0. \] (4.8)

In the same way as for the continuity equation, we get

\[ \Delta x \frac{du_i}{dt} + \frac{1}{2\lambda^2} (\zeta_{i+1} - \zeta_{i-1}) + \Delta x \frac{r}{1 - h_i} u_i = 0. \] (4.9)

The equations (4.6) and (4.9) result in the following discrete system for internal cells:

\[ \frac{d\zeta_i}{dt} = -\frac{1}{2\Delta x} ((1 - h_{i+1}) u_{i+1} - (1 - h_{i-1}) u_{i-1}), \] (4.10a)

\[ \frac{du_i}{dt} = -\frac{1}{2\Delta x \lambda^2} (\zeta_{i+1} - \zeta_{i-1}) - \frac{r}{1 - h_i} u_i. \] (4.10b)

As mentioned before we have as boundary conditions

\[ \zeta(0, t) = \zeta_0, \quad u(L, t) = 0. \] (4.11)

The two Dirichlet conditions define two of our variables, so for \( \zeta_1 \) and for \( u_N \) we do not need equations. For the other two variables at the boundary we derive a discretisation using a boundary cell, whose length is half the length of an internal cell:

\[ \int_{L - \frac{1}{2}\Delta x}^{L} \frac{\partial \zeta}{\partial t} + (1 - h) u \frac{\partial}{\partial x} \, dx = 0, \] (4.12)

\[ \int_{0}^{\frac{1}{2}\Delta x} \left[ \frac{\partial u}{\partial t} + \frac{1}{\lambda^2} \frac{\partial \zeta}{\partial x} + \frac{r}{1 - h} u \right] \, dx = 0. \] (4.13)

Define the cell averaged quantities for cells at the boundary

\[ \zeta_N = \frac{2}{\Delta x} \int_{L - \frac{1}{2}\Delta x}^{L} \zeta(x, t) \, dx, \quad u_1 = \frac{2}{\Delta x} \int_{0}^{\frac{1}{2}\Delta x} u(x, t) \, dx. \] (4.14)

Note that \( L - \Delta x = (N - \frac{3}{2}) \Delta x \). Then the equations become

\[ \frac{1}{2} \Delta x \frac{d\zeta_N}{dt} + \left[ (1 - h) u \right]_{L - \frac{1}{2}\Delta x}^{L} = 0, \] (4.15)

\[ \Delta x \frac{du_1}{dt} + \frac{1}{\lambda^2} \left[ \zeta \right]_{0}^{\frac{1}{2}\Delta x} + \Delta x \frac{r}{2} \frac{u_1}{1 - h_1} = 0. \] (4.16)
This can be rewritten as

\[
\frac{1}{2} \Delta x \frac{d \zeta_N}{dt} - \frac{1}{2} ((1 - h_{N-1})u_{N-1} + (1 - h_N)u_N) + (1 - h_N)u_N = 0,
\]

resulting in

\[
\frac{d \zeta_N}{dt} = \frac{1}{\Delta x} ((1 - h_{N-1})u_{N-1} - (1 - h_N)u_N),
\] (4.17)

for the continuity equation and

\[
\frac{d u_1}{dt} + \frac{2}{\Delta x} \frac{1}{\lambda^2} \left( \frac{1}{2} (\zeta_2 + \zeta_1) - \zeta_1 \right) + \frac{r}{1 - h_1} u_1 = 0,
\]

which results in

\[
\frac{d u_1}{dt} = -\frac{1}{\Delta x} \frac{1}{\lambda^2} (\zeta_2 - \zeta_1) - \frac{r}{1 - h_1} u_1,
\] (4.18)

for the momentum equation. The resulting discretised system follows from equations (4.10a), (4.10b), (4.17) and (4.18), and reads

\[
\begin{align*}
\frac{d \zeta_i}{dt} &= -\frac{1}{2 \Delta x} ((1 - h_{i+1})u_{i+1} - (1 - h_{i-1})u_{i-1}), \quad i = 2, \ldots, N - 1 \\
\frac{d u_i}{dt} &= -\frac{1}{2 \Delta x} \frac{1}{\lambda^2} (\zeta_{i+1} - \zeta_{i-1}) - \frac{r}{1 - h_i} u_i, \quad i = 2, \ldots, N - 1 \\
\frac{d \zeta_N}{dt} &= \frac{1}{\Delta x} ((1 - h_{N-1})u_{N-1} - (1 - h_N)u_N), \\
\frac{d u_1}{dt} &= -\frac{1}{\Delta x} \frac{1}{\lambda^2} (\zeta_2 - \zeta_1) - \frac{r}{1 - h_1} u_1, \\
\zeta_1 &= \sin(t), \\
u_N &= 0.
\end{align*}
\] (4.19) (4.20) (4.21) (4.22) (4.23) (4.24)

### 4.2 Nonlinear model with fixed embayment length

As general dimensionless equations we found equations (2.10):

\[
\begin{align*}
\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x} ((1 - h + \varepsilon \zeta)u) &= 0, \quad (4.25a) \\
\frac{\partial u}{\partial t} + \varepsilon u \frac{\partial u}{\partial x} + \frac{1}{\lambda^2} \frac{\partial \zeta}{\partial x} + \frac{r}{1 - h + \varepsilon \zeta} u &= 0. \quad (4.25b)
\end{align*}
\]

Now, we consider the nonlinear equations in which the depth depends on the wave height, but still we neglect the nonlinear advective terms. Again we take a fixed domain. As system of equations we now have

\[
\begin{align*}
\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x} ((1 - h + \varepsilon \zeta)u) &= 0, \quad (4.26a) \\
\frac{\partial u}{\partial t} + \frac{1}{\lambda^2} \frac{\partial \zeta}{\partial x} + \frac{r}{1 - h + \varepsilon \zeta} u &= 0. \quad (4.26b)
\end{align*}
\]
We have the boundary conditions

\[ \zeta(0, t) = \zeta_0(t), \quad u(L, t) = 0. \] (4.27)

Usually \( \zeta_0(t) = \sin(t) \). We can use the same derivation for the discretisation as in the previous section, resulting in the following discretised system:

\[
\begin{align*}
\frac{d\zeta_i}{dt} &= -\frac{1}{2\Delta x}((1 - h_{i+1} + \varepsilon \zeta_{i+1})u_{i+1} - (1 - h_{i-1} + \varepsilon \zeta_{i-1})u_{i-1}), \quad i = 2, \ldots, N - 1 \quad (4.28) \\
\frac{du_i}{dt} &= -\frac{1}{2\Delta x} \frac{1}{\lambda^2} (\zeta_{i+1} - \zeta_{i-1}) - \frac{r}{1 - h_i + \varepsilon \zeta_i} u_i, \quad i = 2, \ldots, N - 1 \quad (4.29) \\
\frac{d\zeta_N}{dt} &= \frac{1}{\Delta x}((1 - h_{N-1} + \varepsilon \zeta_{N-1})u_{N-1} - (1 - h_N + \varepsilon \zeta_N)u_N), \\
\frac{du_1}{dt} &= -\frac{1}{\Delta x} \frac{1}{\lambda^2} (\zeta_2 - \zeta_1) - \frac{r}{1 - h_1 + \varepsilon \zeta_1} u_1, \\
\zeta_1 &= \sin(t), \\
u_N &= 0. 
\end{align*}
\]

4.3 Nonlinear model with a moving boundary

In order to treat the moving boundary, we need to redefine the grid. The mean seawater level is at \( x = 1 \), but the water front will reach beyond \( x = 1 \). Therefore, we extend the grid to \( x = 3/2 \). Now \( \Delta x = \frac{3}{2(N-1)} \) and \( x_i = (i-1)\Delta x \) for \( i = 1 \ldots N \). Although we have a moving boundary, the gridpoints will be kept fixed. The movement of the boundary will be dealt with only by changing the discretisation locally. The situation is given in Figure 8.

The computational domain reaches from the seaward boundary at \( x = 0 \), to the moving boundary, which is denoted by \( \Gamma(t) \). The moving landward boundary \( \Gamma(t) \) is that location where the depth is zero:

\[ \Gamma(t) := \{ x \in [0, \infty) \mid 1 - h(x) + \varepsilon \zeta(x, t) = 0 \} . \] (4.34)

First, we will derive the discretisation for a single time instance. At this time instance we know

![Figure 8: Grid layout for the problem with a moving boundary.](image)
the location of the boundary (and is fixed). The equations we consider are given by

\[
\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x} ((1 - h + \varepsilon \zeta) u) = 0, \quad (4.35a)
\]

\[
\frac{\partial u}{\partial t} + \frac{1}{\lambda^2} \frac{\partial \zeta}{\partial x} + \frac{r}{1 - h + \varepsilon \zeta} u = 0. \quad (4.35b)
\]

As boundary conditions we prescribe

\[
\zeta(0, t) = \zeta_0(t), \quad (4.36)
\]

\[
u(x, t)(1 - h(x) + \varepsilon \zeta(x, t)) = 0, \quad \text{at } x = \Gamma(t). \quad (4.37)
\]

The computational domain is between \(x = 0\) and the point \(x\) where \(1 - h + \varepsilon \zeta = 0\), i.e. \(\Gamma(t)\). For internal points and the left boundary, the discretisation remains the same as in Section 4.2 in equations (4.28), (4.29) and (4.31). Now suppose that at a certain moment the point \(\Gamma(t)\) lies inside the interval \(i + 1:\)

\[
x_{i+1} - \frac{\Delta x}{2} < \Gamma \leq x_{i+1} + \frac{\Delta x}{2}.
\]

For the moving boundary, we need to consider two cases, see Figure 9:

- **Case 1**, \(\Gamma < x_{i+1}\)
  
  Here we extend cell \(C_i\) to the boundary \(\Gamma\), i.e. \(C_i = [x_{i-\frac{1}{2}}, \Gamma]\) is the boundary cell and we define \(a = \frac{1}{\Delta x} (\Gamma - x_{i+\frac{1}{2}})\). In this case \(0 < a < \frac{1}{2}\).

- **Case 2**, \(\Gamma \geq x_{i+1}\)
  
  Here we have cell \(C_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]\) as an ordinary internal cell and we shorten cell \(C_{i+1} = [x_{i+\frac{1}{2}}, \Gamma]\) as the boundary cell. In this case we define \(a = \frac{1}{\Delta x} (\Gamma - x_{i+\frac{1}{2}})\). This is the same definition, but now it is the complete length of the cell, whereas in the first case it is only part of the cell’s length. Here we have \(\frac{1}{2} \leq a \leq 1\).

### 4.3.1 With fixed cell averages in space

In the above mentioned cases of changing cells we can choose where in space we locate the cell average, since the boundary cell moves. In this section we will keep the cell averages fixed in space at the original gridpoints. In Section 4.3.2 we will move the cell averages to the middle of the actual cell. In both cases we will discuss the boundary cell, since this is the only cell for which we have to make a choice, the internal cells will remain unchanged. We will describe the derivations for case 1 here and give similar derivations for case 2 in the Appendix B.

**Case 1**

To discretise the equations (4.35) in case 1 we define

\[
\zeta_i = \frac{1}{(1 + a)\Delta x} \int_{(i-\frac{1}{2})\Delta x}^{(i+\frac{1}{2}+a)\Delta x} \zeta(x, t)dx, \quad (4.38)
\]

and

\[
u_i = \frac{1}{(1 + a)\Delta x} \int_{(i-\frac{1}{2})\Delta x}^{(i+\frac{1}{2}+a)\Delta x} u(x, t)dx. \quad (4.39)
\]

Now integrate the continuity equation (4.35a) over boundary cell \(C_i\),

\[
\int_{C_i} \frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x} ((1 - h + \varepsilon \zeta) u) \, dx = 0. \quad (4.40)
\]
Figure 9: Situations of the grid at the moving boundary.

Since the boundary of the integral depends on time, we cannot simply interchange integration and differentiation. Instead, we discretise the differentiation with respect to time inside the integral. Here, the superscript indices denote the time level. Considering equation (4.40), we get

\[
\int_{c_i} \frac{\zeta^j - \zeta^{j-1}}{\Delta t} \, dx + \int_{c_i} \frac{\partial}{\partial x} ((1 - h + \varepsilon \zeta) u) \, dx = 0, \quad (4.41)
\]
\[
\frac{(1 + a) \Delta x}{\Delta t} (\zeta^j - \zeta^{j-1}) + [(1 - h + \varepsilon \zeta) u]_{x_i - \frac{1}{2}} = 0. \quad (4.42)
\]

Note that we still have to choose the time level for the second term. We will make this choice later on. At the moving boundary at \( x = \Gamma \), using boundary condition (4.37), we get

\[
(1 + a) \frac{\Delta x}{\Delta t} (\zeta^j - \zeta^{j-1}) = (1 - h + \varepsilon \zeta) u |_{x_i - \frac{1}{2}}. \quad (4.43)
\]

We have a choice whether we want to keep the cell averages in the gridpoints, or whether we want to have them in the middle of the cell. Only for the cell at the boundary this makes a difference. In this section we will keep the cell averages fixed in the gridpoints, and in the next section we will take the averages in the middle of the cell. As before we will use linear interpolation between the averages of two adjacent cells to get the value at the common edge of the two cells. Using this procedure, equation (4.43) gives us for the continuity equation

\[
(1 + a) \frac{\Delta x}{\Delta t} (\zeta^j - \zeta^{j-1}) = (1 - h + \varepsilon \zeta) u |_{x_i - \frac{1}{2}}
\]
\[
= \frac{1}{2} ((1 - h_{i-1} + \varepsilon \zeta_{i-1}) u_{i-1} + (1 - h_i + \varepsilon \zeta_i) u_i). \quad (4.44)
\]

For the momentum equation (4.35b), we integrate over the boundary cell and rewrite to obtain

\[
\int_{c_i} \frac{\partial u}{\partial t} \, dx + \frac{1}{\lambda^2} [\zeta] c_i + \int_{c_i} \frac{r}{1 - h + \varepsilon \zeta} u \, dx = 0. \quad (4.45)
\]
Again we have to be careful if we want to interchange integration and differentiation with respect to \( t \), so we discretise the time derivative inside the integral,

\[
\int_{c_i} \frac{u^j - u^{j-1}}{\Delta t} \, dx + \frac{1}{\lambda^2} \left[ \zeta \right]_{c_i} + \int_{c_i} \frac{r}{1 - h + \varepsilon \zeta} u \, dx = 0, 
\]

(4.46)

\[
(1 + a) \Delta x \frac{u^j - u^{j-1}}{\Delta t} + \frac{1}{\lambda^2} \left[ \zeta \right]_{c_i} + \int_{c_i} \frac{r}{1 - h + \varepsilon \zeta} u \, dx = 0. 
\]

(4.47)

Use of the midpoint rule for the integral gives after rearrangement

\[
(1 + a) \frac{\Delta x}{\Delta t} (u^j_i - u^{j-1}_i) = -\frac{1}{\lambda^2} \left( \zeta|_{x=\Gamma} - \zeta|_{x=x_{i-1/2}} \right) - (1 + a) \Delta x \frac{r}{1 - h_i + \varepsilon \zeta_i} u_i. 
\]

(4.48)

To rewrite the second term consisting of \( \zeta|_{x=\Gamma} \), we use that \( 1 - h + \varepsilon \zeta = 0 \) at \( x = \Gamma = x_i + (\frac{1}{2} + a) \Delta x \) and use linear interpolation (since \( h \) is also a grid function). Then we get

\[
\zeta|_{x=\Gamma} = \frac{h - 1}{\varepsilon} \left|_{x=\Gamma} \right. = \frac{1}{\varepsilon} \left( h_{i+1} - h_i \right) \frac{(\frac{1}{2} + a) \Delta x}{\Delta x} + h_i - 1 
= \frac{1}{\varepsilon} \left( \left( \frac{1}{2} - a \right) h_i + \left( \frac{1}{2} + a \right) h_{i+1} \right) - 1. 
\]

(4.49)

With linear interpolation we get

\[
\zeta|_{x=x_{i-1/2}} = \frac{1}{2} (\zeta_{i-1} + \zeta_i). 
\]

(4.50)

Now by substitution of equation (4.49) and (4.50) into equation (4.48) we arrive at the discretisation for the momentum equation given by

\[
(1 + a) \frac{\Delta x}{\Delta t} (u^j_i - u^{j-1}_i) = -\frac{1}{\varepsilon \lambda^2} \left( \left( \frac{1}{2} - a \right) h_i + \left( \frac{1}{2} + a \right) h_{i+1} \right) - 1 
+ \frac{1}{2 \lambda^2} (\zeta_{i-1} + \zeta_i) - (1 + a) \Delta x \frac{r}{1 - h_i + \varepsilon \zeta_i} u_i. 
\]

(4.51)

**Final results for case 1 and case 2**

Summarised, we have for the equations with a moving boundary, of which the location is known, the following discretisation for the boundary cell in the two cases:

**Case 1**

\[
(1 + a) \frac{\Delta x}{\Delta t} (\zeta^j_i - \zeta^{j-1}_i) = \frac{1}{2} \left( (1 - h_{i+1} + \varepsilon \zeta_{i+1}) u_{i+1} + (1 - h_i + \varepsilon \zeta_i) u_i \right), 
\]

(4.52)

\[
(1 + a) \frac{\Delta x}{\Delta t} (u^j_i - u^{j-1}_i) = -\frac{1}{\varepsilon \lambda^2} \left( \left( \frac{1}{2} - a \right) h_i + \left( \frac{1}{2} + a \right) h_{i+1} \right) - 1 
+ \frac{1}{2 \lambda^2} (\zeta_{i-1} + \zeta_i) - (1 + a) \Delta x \frac{r}{1 - h_i + \varepsilon \zeta_i} u_i. 
\]

(4.53)

Case 2, from Appendix B equations (B.7) and (B.13)
\[ a \frac{\Delta x}{\Delta t} \left( \zeta_{i+1}^j - \zeta_{i+1}^{j-1} \right) = \frac{1}{2} \left( (1 - h_i + \varepsilon \zeta_i) u_i + (1 - h_{i+1} + \varepsilon \zeta_{i+1}) u_{i+1} \right), \] (4.54)

\[ a \frac{\Delta x}{\Delta t} \left( u_{i+1}^j - u_{i+1}^{j-1} \right) = -\frac{1}{\varepsilon \lambda^2} \left( \left( \frac{3}{2} - a \right) h_{i+1} + \left( a - \frac{1}{2} \right) h_{i+2} - 1 \right) + \frac{1}{2 \lambda^2} (\xi_i + \zeta_{i+1}) - \frac{r a \Delta x}{1 - h_{i+1} + \varepsilon \zeta_{i+1}} u_{i+1}. \] (4.55)

We can take the formulas for the two cases together, if we define

\[ b = \begin{cases} 1 + a & \text{for case 1} \\ a & \text{for case 2} \end{cases} \quad \text{and} \quad k = \begin{cases} i & \text{for case 1} \\ i + 1 & \text{for case 2}. \end{cases} \] (4.56)

Then we have the following discrete equations at the moving boundary for fixed cell averages (in both cases),

\[ b \frac{\Delta x}{\Delta t} \left( \zeta_k^j - \zeta_k^{j-1} \right) = \frac{1}{2} \left( (1 - h_{k-1} + \varepsilon \zeta_{k-1}) u_{k-1} + (1 - h_k + \varepsilon \zeta_k) u_k \right), \] (4.57)

\[ b \frac{\Delta x}{\Delta t} \left( u_k^j - u_k^{j-1} \right) = -\frac{1}{\varepsilon \lambda^2} \left( \left( \frac{3}{2} - b \right) h_k + \left( b - \frac{1}{2} \right) h_{k+1} - 1 \right) + \frac{1}{2 \lambda^2} (\xi_{k-1} + \zeta_k) - \frac{r b \Delta x}{1 - h_k + \varepsilon \zeta_k} u_k. \] (4.58)

### 4.3.2 With moving cell averages

Instead of keeping the cell averages at fixed points in space, we could also choose to keep the averages at the middle of the cell. So the boundary cell is the only cell with a change. Then we have to adapt the interpolations. We also need to adapt the cell average when a cell splits (we have a new cell) or when two cells merge (we have a cell less). In this section we will describe the difference compared to keeping the averages fixed as described in the previous section.

For case 1 we found (equation (4.43)),

\[ (1 + a) \frac{\Delta x}{\Delta t} \left( \zeta_i^j - \zeta_i^{j-1} \right) = (1 - h + \varepsilon \zeta) u \big|_{x_{i-\frac{1}{2}}}. \] (4.59)

At the interface of two adjacent cells we will interpolate (as before) to get the value at that point. We use linear interpolation between the adjacent cell averages. As mentioned we will put the cell averages in the middle of the corresponding cells, also in the boundary cell. Since the width of this last cell is not constant, we have to adapt the interpolation. In this case the width of the last cell is \((1 + a) \Delta x\). The situation is sketched in Figure 10, indicated is the middle of the boundary cell. Note that the common cell edge stay at the same position \(x_{i-\frac{1}{2}}\). The
interpolation for a general function \( f \) is given as
\[
f_{\text{lin}}(x) = \left( f_i - f_{i-1} \right) \frac{1}{(1 + \frac{a}{2})\Delta x}(x - x_{i-1}) + f_{i-1},
\]
\[
f_{\text{lin}}(x_{i-\frac{1}{2}}) = \left( f_i - f_{i-1} \right) \frac{1}{(1 + \frac{a}{2})\Delta x}(\frac{1}{2}\Delta x) + f_{i-1}
\]
\[
= \frac{1}{2 + a}(f_i - f_{i-1}) + f_{i-1}
\]
\[
= \frac{1}{2 + a}((1 + a)f_{i-1} + f_i).
\]

One can verify that with \( a = 0 \) we have the usual average. Using this linear interpolation we get for the continuity equation
\[
(1 + a)\Delta x \frac{\Delta t}{\Delta x} \left( \zeta_j^i - \zeta_j^{i-1} \right) = (1 - h + \varepsilon \zeta)u|_{x_{i-\frac{1}{2}}}
\]
\[
= \frac{1}{2 + a}((1 + a)(1 - h_{i-1} + \varepsilon \zeta_{i-1})u_{i-1} + (1 - h_i + \varepsilon \zeta_i)u_i).
\]

For the momentum equation we have equation (4.48),
\[
(1 + a)\Delta x \frac{\Delta t}{\Delta x} \left( u_j^i - u_j^{i-1} \right) = -\frac{1}{\lambda^2} \left( \zeta|_{x=\Gamma} - \zeta|_{x=x_{i-\frac{1}{2}}} \right) - (1 + a)\Delta x \frac{r}{1 - h_i + \varepsilon \zeta_i} u_i,
\]

and equation (4.49) remains unchanged. With the formula for linear interpolation (4.61) we get
\[
\zeta|_{x=x_{i-\frac{1}{2}}} = \frac{1}{2 + a}((1 + a)\zeta_{i-1} + \zeta_i)).
\]

Now by substitution of equation (4.49) and (4.64) into equation (4.63) we arrive at the following discretisation for the momentum equation
\[
(1 + a)\Delta x \frac{\Delta t}{\Delta x} \left( u_j^i - u_j^{i-1} \right) = -\frac{1}{\varepsilon \lambda^2} \left( \left( \frac{1}{2} - a \right) h_i + \left( \frac{1}{2} + a \right) h_{i+1} \right) - (1 + a)\Delta x \frac{r}{1 - h_i + \varepsilon \zeta_i} u_i.
\]

Summarised, we have for the equations with a moving (but given) boundary the following discretisation for the boundary cell in the two cases:

Case 1
\[(1 + a) \frac{\Delta x}{\Delta t} \left( \zeta_j^i - \zeta_j^{i-1} \right) = \frac{1}{2 + a} \left( (1 + a)(1 - h_{i-1} + \varepsilon \zeta_{i-1})u_{i-1} + (1 - h_i + \varepsilon \zeta_i)u_i \right), \quad (4.66)\]

\[(1 + a) \frac{\Delta x}{\Delta t} \left( u_j^i - u_j^{i-1} \right) = -\frac{1}{\varepsilon \lambda^2} \left( \left( \frac{3}{2} - a \right) h_{i+1} + \left( a - \frac{1}{2} \right) h_{i+2} - 1 \right) + \frac{1}{\lambda^2} \frac{1}{1 + a} \left( (1 + a) \zeta_i + \zeta_{i+1} \right) - (1 + a) \Delta x \frac{r}{1 - h_i + \varepsilon \zeta_i} u_i. \quad (4.67)\]

Case 2, from Appendix B equations (B.17) and (B.20)

\[a \frac{\Delta x}{\Delta t} \left( \zeta_{i+1}^j - \zeta_{i+1}^{j-1} \right) = \frac{1}{1 + a} \left( a(1 - h_i + \varepsilon \zeta_i)u_i + (1 - h_{i+1} + \varepsilon \zeta_{i+1})u_{i+1} \right), \quad (4.68)\]

\[a \frac{\Delta x}{\Delta t} \left( u_{i+1}^j - u_{i+1}^{j-1} \right) = -\frac{1}{\varepsilon \lambda^2} \left( \left( \frac{3}{2} - a \right) h_{i+1} + \left( a - \frac{1}{2} \right) h_{i+2} - 1 \right) + \frac{1}{\lambda^2} \frac{1}{1 + a} \left( a \zeta_i + \zeta_{i+1} \right) - \frac{r a \Delta x}{1 - h_{i+1} + \varepsilon \zeta_{i+1}} u_{i+1}. \quad (4.69)\]

As before we can take the formulas for the two cases together using the same definition (4.56),

\[b = \begin{cases} 
1 + a & \text{for case 1} \\
1 & \text{for case 2}
\end{cases} \quad \text{and} \quad k = \begin{cases} 
i & \text{for case 1} \\
 i + 1 & \text{for case 2}
\end{cases}. \quad (4.70)\]

Then we have as the discrete equations at the moving boundary for moving cell averages (in both cases),

\[b \frac{\Delta x}{\Delta t} \left( \zeta_k^j - \zeta_k^{j-1} \right) = \frac{1}{1 + b} \left( b(1 - h_{k-1} + \varepsilon \zeta_{k-1})u_{k-1} + (1 - h_k + \varepsilon \zeta_k)u_k \right), \quad (4.71)\]

\[b \frac{\Delta x}{\Delta t} \left( u_k^j - u_k^{j-1} \right) = -\frac{1}{\varepsilon \lambda^2} \left( \left( \frac{3}{2} - b \right) h_k + \left( b - \frac{1}{2} \right) h_{k+1} - 1 \right) + \frac{1}{\lambda^2} \frac{1}{1 + b} \left( b \zeta_k + \zeta_{k+1} \right) - \frac{r b \Delta x}{1 - h_k + \varepsilon \zeta_k} u_k. \quad (4.72)\]

4.4 Level set method

The level set method uses a continuous function \( \phi \) in order to trace the moving boundary \( \Gamma(t) \), in our problem the water front at the shore, as its zero level set:

\[\phi(x, t) = 0 \iff x \in \Gamma(t). \quad (4.73)\]

The function \( \phi \) is called the level set function. The level set method was first introduced by Osher and Sethian [5]. The motion of the level set function is determined by

\[\frac{\partial \phi}{\partial t} + v_n \| \nabla \phi \| = 0, \quad (4.74)\]
where $v_n$ denotes the normal component of the front velocity. This equation is only valid at the boundary $\Gamma(t)$, because the velocity $v_n$ is only defined at the boundary. If we extend the velocity $v_n$ continuously over the whole domain $\Omega$ and call the result the vector field $\mathbf{v}$, we obtain a generalised equation for the level set function:

$$
\frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0.
$$

(4.75)

This equation is known as the level set equation. We numerically solve this equation using a forward Euler time integration and a first order upwind method for the spatial discretisation. For our one dimensional problem this will give the discretisation

$$
\phi^{j+1}_i - \phi^j_i = - \left( -v_1^j \phi_{i-1}^j + |v_1| \phi_i^j + v_1^- \phi_{i+1}^j \right),
$$

(4.76)

where $v$ is the continuous extension of the velocity at the boundary, $v_1^+ = \max\{0, v_1\}$ and $v_1^- = \min\{v_1, 0\}$. Note that $|v_1| = v_1^+ - v_1^-$. 

### 4.4.1 Extension of front velocity

In order to solve the equation for the level set function on the whole domain, we need to extend the front velocity continuously. In one dimension this is an easy task, since the moving boundary consists of one point, so the continuation can be just a function that is constant over space in which its value is dictated by the interface velocity. In higher dimensions this is a more complicated task and, in order to test whether we can handle such a situation, we will use a method that works in all dimensions. An extension of the front velocity can be obtained by solving

$$
\frac{\partial v}{\partial \tau} + \text{Sign} \left( \phi \frac{\partial \phi}{\partial x} \right) \frac{\partial v}{\partial x} = 0,
$$

(4.77a)

$$
v = u|_{\Gamma} \text{ at } x = \Gamma(t),
$$

(4.77b)

where $\tau$ is a pseudo-time, i.e. an artificial variable just to extend the velocity continuously. The velocity from the model equations at the interface is denoted by $u|_{\Gamma}$. Equation (4.77) is valid for a one dimensional problem. For more dimensions one has to choose in which way one wants the extension. If we use extension in the Cartesian directions for a two dimensional problem we have to solve the equations

$$
\frac{\partial v_1}{\partial \tau} + \text{Sign} \left( \phi \frac{\partial \phi}{\partial x} \right) \frac{\partial v_1}{\partial x} = 0,
$$

(4.78a)

$$
\frac{\partial v_2}{\partial \tau} + \text{Sign} \left( \phi \frac{\partial \phi}{\partial y} \right) \frac{\partial v_2}{\partial y} = 0,
$$

(4.78b)

$$
v = u|_{\Gamma} \text{ for } \mathbf{x} \in \Gamma(t),
$$

(4.78c)

where $u_1$ is the (extended) velocity in the $x$-direction and $u_2$ in the $y$-direction, $\mathbf{v} = [v_1, v_2]^T$. These equation come from [7], Section 4.2.1. These equations are such that the velocity is extended away from the interface. An alternative is the use of an extension in the direction normal to the interface. 

Equations (4.77) for the extension of the velocity can be discretised using a forward Euler time integration and a first order upwind method for the spatial discretisation. We will discuss only the discretisation for the one dimensional problem. In order to be able to use the ‘boundary’
condition at the interface, we adjust the grid in the vicinity of the interface: we shift the shared boundary of the control volumes corresponding to the two gridpoints adjacent to the interface onto the interface position. In this way it is easy to use the prescribed velocity at the interface. Say the interface position, which is \( x = \Gamma(t) \), lies between the gridpoints \( x_i \) and \( x_{i+1} \), see Figure 11. Redefine
\[
v_i = \frac{1}{\frac{1}{2} + a} \Delta x \int_{x_i - \frac{1}{2} \Delta x}^{\Gamma(t)} v \, dx, \tag{4.79}
\]
\[
v_{i+1} = \frac{1}{\frac{3}{2} - a} \Delta x \int_{\Gamma(t)}^{x_{i+1} + \frac{1}{2} \Delta x} v \, dx, \tag{4.80}
\]
where \( a \) is defined as the fraction of the gridspacing to the left of the interface, \( a := \frac{\Gamma(t) - x_i}{\Delta x} \).

![Diagram](image_url)

**Figure 11:** Grid adjustment for the interface.

Using a forward Euler and first order upwind method, we get the same type of discretisation as in equation (4.76), except for the two points near the interface. These discretisations are given by
\[
\frac{dv_i}{d\tau} = -\frac{1}{\frac{1}{2} + a} \Delta x v_i + \frac{u|_\Gamma}{\frac{1}{2} + a} \Delta x, \tag{4.81}
\]
\[
\frac{dv_{i+1}}{d\tau} = -\frac{1}{\frac{3}{2} - a} \Delta x v_{i+1} + \frac{u|_\Gamma}{\frac{3}{2} - a} \Delta x. \tag{4.82}
\]

The absolute value of the sign function is one, so from convection equation (4.77a) we have that the numerical speed is 1. Since the numerical speed during continuation of the velocity is 1, we have the condition \( \frac{\Delta \tau}{\Delta x} \leq 1 \), so we need to choose a value for the timestep that is slightly smaller than the grid spacing.

The initial condition for \( v \) does not make much difference, so it might be taken zero. Also one could use the extended velocity of the previous timestep for \( t \).

In our case the front velocity will be continued over the whole domain, although it suffices to extend the velocity only in a band around the interface position. Most of the times it is only continued in a band, since this is less computational work and hence it makes the computations faster.

### 4.4.2 Reinitialisation

After updating the level set function using equation (4.75), the level set function might not be a signed distance function anymore. Since flat or steep gradients might give less accurate approximations, we reinitialise the level set function to a signed distance function using the equation
\[
\frac{\partial \phi}{\partial \tau}(x, \tau) = S(\phi(x, t)) \left( 1 - \left| \frac{\partial \phi}{\partial x}(x, \tau) \right| \right), \tag{4.83}
\]
with initial condition $\varphi(x, 0) = \phi(0, t)$. Here $\tau$ is pseudo-time and $S$ is a smoothed sign function, given by

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + (2\Delta x)^2}}.$$  \hfill (4.84)

For two values of $\Delta x$ this smoothed sign function is plotted in Figure 12. The interface position is not advected by equation (4.83), since at that point $\phi = 0$, so $S(\phi)=0$. This equation has a stable steady-state solution, which is $\left| \frac{\partial \varphi}{\partial x} \right| = 1$, see [6], Section 7.4, page 66. After this equilibrium is thought to be reached, one sets $\varphi = \phi$.

Equation (4.83) needs a delicate numerical treatment, but can be solved using a monotone Godunov upwind scheme. The term $G(\varphi) := 1 - \left| \frac{\partial \varphi}{\partial x} (x, \tau) \right|$ is approximated by

$$G(\varphi_i) = \begin{cases} 
1 - \sqrt{\max(a_+^2, b_+^2)}, & \text{if } \varphi_i > 0, \\
1 - \sqrt{\max(a_-^2, b_-^2)}, & \text{if } \varphi_i < 0, \\
0, & \text{otherwise},
\end{cases}$$

where $a = D^\varphi \varphi_i$ and $b = D^\varphi_+ \varphi_i$ are used to denote the left or right sided derivatives of $\varphi$ with respect to $x$, and the notation $a_- = \min(a, 0)$ and $a_+ = \max(a, 0)$ is used. The time integration for the reinitialisation is done using a third order TVD Runge-Kutta scheme, whereas a fifth order WENO scheme is used for the spatial derivatives. Both can be found in [6], Section 3.4 and 3.5.

4.4.3 Process overview / Integration with model equations

The level set method can be used for the physical system as described in Section 4.3 applying the following recipe:

Set index $i = 1$, $t_1 = 0$ and initialise the variables and $\phi$ and $\Gamma(0)$ using the initial condition for $\zeta$.

1. Calculate $\Gamma(t_i)$ and adapt the variables to the new size.

2. Solve the variables for the next time level $t_{i+1}$.  

Figure 12: Smoothed sign function from equation (4.84) for two values of $\Delta x$. 
3. Extend the front velocity over the domain.

4. Calculate $\phi(t_{i+1})$.

5. Reinitialise $\phi(t_{i+1})$.

6. Set the time level $t_{i+1} = t_i + \Delta T$ and index $i = i + 1$, and repeat from point 1.

### 4.5 Time integration

For the obtained semi-discretisation we still have to perform a time integration. For a general semi-discretised system as
\[
\frac{dy}{dt} = Ay + f, \tag{4.85}
\]
we will use the Crank-Nicholson method, also known as the $\theta$-method or trapezoidal rule,
\[
\frac{y^{n+1} - y^n}{\Delta t} = (1 - \theta) (Ay^n + f^n) + \theta (Ay^{n+1} + f^{n+1}). \tag{4.86}
\]
Rewriting gives
\[
(I - \Delta t \theta A)y^{n+1} = (I + \Delta t(1 - \theta)A)y^n + \Delta t(1 - \theta)f^n + \Delta t \theta f^{n+1}. \tag{4.87}
\]
If $\theta = 0$ we have the explicit Euler forward scheme and for $\theta = 1$ we have the (implicit) Euler backward method. One should have $0 \leq \theta \leq 1$. For all choices of $\theta$ the error is of first order, except for $\theta = \frac{1}{2}$, when we have a second order accurate time integration method. In order to be stable, the Euler forward method has a strict condition on the timestep, whereas the Euler backward method gives an unconditionally stable scheme.

### 4.6 Choice of initial value for new cell

For the moving boundary we have to make a choice during some time steps if the interface moves into another control volume. Since for the update of each cell average we need a value at the previous time, we are faced with a complication if after the previous timestep the numerical domain is extended with an extra cell. At the previous time level, there was no cell average to compute, but we need an old value for this new cell at the current time level. We have a couple of possible choices. First we can linear extrapolate the adjacent two cell averages at the old time level, so we get
\[
y_{i+1} = 2y_i - y_{i-1}, \tag{4.88}
\]
where $y = \zeta$ or $u$. For $\zeta$ we can also make another choice, since we know that the depth at the interface is zero. The situation is sketched in Figure 13, where the little circle indicates the desired value. So the water level at the interface is given by $\zeta|_{\Gamma} = \frac{h_{i-1}}{\varepsilon}$. Then we can determine the new value, which lies between $x_i$ and $\Gamma(t_{n+1})$, by interpolation, as follows
\[
\zeta_{i+1} = \zeta_i + \frac{\zeta|_{\Gamma} - \zeta_i}{\frac{1}{2} + a} \cdot 1, \tag{4.89}
\]
which gives, after substitution of $\zeta|_{\Gamma}$ and interpolation of the bottom profile (since this is only a grid function as well),
\[
\zeta_{i+1} = \zeta_i + \frac{1}{\frac{1}{2} + a} \left[ \frac{1}{\varepsilon} \left( -1 + \left( \frac{3}{2} - a \right) h_{i+1} + \left( a - \frac{1}{2} \right) h_{i+2} \right) - \zeta_i \right]. \tag{4.90}
\]
Note that this choice guarantees the depth to be positive in our new grid point, whereas the other standard interpolation might give a negative depth.

For $u$, we can do the same, if we assume that $u$ at the interface does not change, which is a good assumption, since we stay at the old time level during interpolation. Hence, for $u$ we have

$$u_{i+1} = u_i + \frac{u|_{\Gamma} - u_i}{2} + a.$$  

(4.91)

### 4.7 Diagonalisation for a test problem

To find an alternative for the central difference scheme and possibly a more optimal numerical scheme, we try to diagonalise our system. To try the idea of diagonalisation, we apply it on a (probably) similar test system. This system is given by

$$\frac{\partial \zeta}{\partial t} + \alpha \frac{\partial u}{\partial x} = 0,$$  

(4.92a)

$$\frac{\partial u}{\partial t} + \beta \frac{\partial \zeta}{\partial x} = 0.$$  

(4.92b)

It can be rewritten in matrix-vector form as

$$\frac{\partial y}{\partial t} + A \frac{\partial y}{\partial x} = 0,$$  

(4.93)

where

$$y = \begin{pmatrix} \zeta \\ u \end{pmatrix}, \quad \text{and} \quad A = \begin{pmatrix} 0 & \alpha \\ \beta & 0 \end{pmatrix}.$$  

This system can be diagonalised which results into

$$\frac{\partial w}{\partial t} + \Lambda \frac{\partial w}{\partial x} = 0,$$  

(4.94)

where

$$w = R^{-1}y, \quad \Lambda = \begin{pmatrix} \sqrt{\alpha \beta} & 0 \\ 0 & -\sqrt{\alpha \beta} \end{pmatrix} = R^{-1}AR, \quad \text{and} \quad R = \begin{pmatrix} \sqrt{\alpha} & -\sqrt{\alpha} \\ \sqrt{\beta} & \sqrt{\beta} \end{pmatrix}.$$  

The vector $w$ contains diagonalised variables, $\Lambda$ is a diagonal matrix with the eigenvalues as its entries and $R$ is the matrix of corresponding eigenvectors. This diagonalised system is easy to solve, since it is decoupled and consists of scalar advection equations. Since for each equation there is a distinct direction, indicated by the sign of the eigenvalue, we can use a standard
upwind discretisation. For the boundary conditions of the diagonalised system we assume that \( w_1(x = 0) \) and \( w_2(x = 1) \) are known.

We are interested in the stability and maximal timestep for the numerical method. For the forward Euler time integration, the condition is \( \sqrt{\alpha \beta \Delta t \Delta x} < 1 \).

For application of the diagonalisation to a nonlinear system, the diagonalisation should be performed at every timestep. Also note that in order to apply such a scheme to our original system, the boundary conditions should be rewritten into the diagonalised variables. This is not a trivial task. This is difficult, since we have one condition for \( \zeta(0, t) \) at the seaward boundary and one condition for \( u(1, t) \) at the landward boundary, but these result into two conditions for both diagonalised variables at each boundary. But this does not give a condition for a single variable. A possible solution is to use the characteristics and trace one variable back from one to the other boundary, in order to get conditions for just one variable. Further research might be carried out on the boundary conditions and on real application of the method on the original system.

For a test system with a friction term included, the diagonalisation is exactly the same. The friction term is affected by the diagonalisation, but the form is the same. In the original system (4.93) we add a term \( F_y \), where \( F = \left( \begin{smallmatrix} 0 & 0 \\ 0 & r \end{smallmatrix} \right) \) as in the physical problem. This term results in an extra term in equation (4.94) given by \( R^{-1}F Rw \), this means that friction is affecting both diagonalised variables. For the Euler forward method we have that \( \Delta t < \frac{2\Re(\mu)}{\mu \Delta x^2} \), where \( \mu \) are the eigenvalues of the spatially discretised system. \( \Delta t \) must be chosen such that the inequality holds for all eigenvalues \( \mu \).
5 Numerical results

In this chapter, the numerical results will be presented and discussed. First we will make a comparison between the analytical and numerical solution on a domain with fixed boundaries. Further on, the numerical solutions for the domain with a moving boundary will be given.

5.1 Comparison of analytical and numerical solution

5.1.1 Frictionless case

Neglecting nonlinear terms and bottom friction ($r = 0$), the system of equations (3.1) reduces to

\[
\frac{\partial \zeta}{\partial t} + \frac{\partial u}{\partial x} = 0 \quad (5.1a)
\]

\[
\frac{\partial u}{\partial t} + \frac{1}{\lambda^2} \frac{\partial \zeta}{\partial x} = 0, \quad (5.1b)
\]

with boundary conditions

\[
\zeta (0, t) = \sin (t), \quad u(1, t) = 0. \quad (5.2)
\]

As periodic analytical solution we found in Section 3.1

\[
\zeta (x, t) = \sin t \cos(\lambda (1 - x)), \quad (5.3a)
\]

\[
u(x, t) = \frac{\cos t \sin (\lambda (1 - x))}{\lambda} \cos(\lambda t). \quad (5.3b)
\]

In Table 3 the difference between the analytical and numerical solution is shown. The simulation is done using the following parameters: $H = 7$, $r = 0$. As the initial condition for the numerical

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$N$</th>
<th>Mstep</th>
<th>$\Delta t$</th>
<th>$\Delta x$</th>
<th>Scale</th>
<th>Max. average</th>
<th>Abs. average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>1</td>
<td>0.05</td>
<td>2.45</td>
<td>$\zeta$</td>
<td>8.327 \cdot 10^{-6}</td>
<td>5.636 \cdot 10^{-6}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>2.917 \cdot 10^{-2}</td>
<td>1.489 \cdot 10^{-2}</td>
</tr>
<tr>
<td>1</td>
<td>50</td>
<td>1</td>
<td>0.0005</td>
<td>0.0245</td>
<td>$\zeta$</td>
<td>1.084 \cdot 10^{-7}</td>
<td>6.655 \cdot 10^{-8}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>2.918 \cdot 10^{-4}</td>
<td>1.489 \cdot 10^{-4}</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>1</td>
<td>0.0005</td>
<td>0.45</td>
<td>$\zeta$</td>
<td>2.058 \cdot 10^{-6}</td>
<td>1.144 \cdot 10^{-6}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>2.918 \cdot 10^{-4}</td>
<td>1.621 \cdot 10^{-4}</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>1</td>
<td>0.0005</td>
<td>0.0495</td>
<td>$\zeta$</td>
<td>8.499 \cdot 10^{-8}</td>
<td>5.659 \cdot 10^{-8}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>2.918 \cdot 10^{-4}</td>
<td>1.474 \cdot 10^{-4}</td>
</tr>
<tr>
<td>0</td>
<td>50</td>
<td>100</td>
<td>5 \cdot 10^{-7}</td>
<td>2.45 \cdot 10^{-5}</td>
<td>$\zeta$</td>
<td>7.068 \cdot 10^{-8}</td>
<td>3.596 \cdot 10^{-8}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>3.944 \cdot 10^{-7}</td>
<td>1.893 \cdot 10^{-7}</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>50</td>
<td>100</td>
<td>0.005</td>
<td>0.245</td>
<td>$\zeta$</td>
<td>9.003 \cdot 10^{-8}</td>
<td>4.940 \cdot 10^{-8}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>3.989 \cdot 10^{-6}</td>
<td>2.279 \cdot 10^{-6}</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>50</td>
<td>1</td>
<td>0.005</td>
<td>0.245</td>
<td>$\zeta$</td>
<td>1.052 \cdot 10^{-7}</td>
<td>5.668 \cdot 10^{-7}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>4.785 \cdot 10^{-6}</td>
<td>2.617 \cdot 10^{-6}</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>50</td>
<td>1</td>
<td>0.0005</td>
<td>0.0245</td>
<td>$\zeta$</td>
<td>7.119 \cdot 10^{-8}</td>
<td>3.614 \cdot 10^{-8}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>2.580 \cdot 10^{-7}</td>
<td>9.672 \cdot 10^{-8}</td>
</tr>
</tbody>
</table>

Table 3: Difference between analytical and numerical solution, without friction, for several values of the parameters.
method, we take the periodic analytical solution at \( t = 0 \). The parameter \( \text{Mstep} \) denotes how often the computed solution is stored. Since for some case the timestep becomes so small, that the computed solution cannot be stored at all the timelevels. If \( \text{Mstep}=1 \), at all the computed timestep the solutions are stored. In the column 'Max. average' the maximal value over time of the mean absolute error in place is shown. The maximum average is defined as

\[
\text{MA}(\zeta) = \max_{1 \leq j \leq n} \frac{1}{N - 1} \sum_{i=2}^{N} \left| \zeta_{i}^{j} - \zeta_{\text{anal}} \right|.
\] (5.4)

In the column 'Abs. average' the average value of the average absolute error is given. The absolute average is defined as

\[
\text{AA}(\zeta) = \frac{1}{n(N - 1)} \sum_{j=1}^{n} \sum_{i=2}^{N} \left| \zeta_{i}^{j} - \zeta_{\text{anal}} \right|.
\] (5.5)

The error is computed over one period (2\( \pi \) time), where we take the second period of the simulation in order to have less influence of initial conditions and initial oscillation. In the first period of the simulation there is in some case some oscillation which is quickly decaying. We can see there is overall a good fit, especially for \( \zeta \).

The simulation for \( \theta = 0 \) is done with \( \text{Mstep}=100 \) and cost about 632 seconds and the postprocess another 88 seconds. Saving the vector at all time steps is not possible because of the memory. The difference in the error in comparison with saving at all timesteps is small. The timestep should at least be this small in order to have a stable solution, i.e. a solution which does not grow with time.

Figure 14 shows the difference between the analytical and numerical solution at two time steps.
Figure 14: Difference of analytical and numerical solution, explicit Euler, at t=7.7 and t=12.
Table 4: Difference between analytical and numerical solution, with friction, for several values of some parameters.

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>$N$</th>
<th>$\Delta t$</th>
<th>$\frac{\Delta t}{N}$</th>
<th>Scale</th>
<th>Max. average</th>
<th>Abs. average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>0.0005</td>
<td>0.0245</td>
<td>$\zeta$</td>
<td>2.948 $\cdot 10^{-7}$</td>
<td>1.688 $\cdot 10^{-7}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>2.921 $\cdot 10^{-4}$</td>
<td>1.490 $\cdot 10^{-4}$</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>0.005</td>
<td>0.045</td>
<td>$\zeta$</td>
<td>7.006 $\cdot 10^{-6}$</td>
<td>4.011 $\cdot 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>2.927 $\cdot 10^{-3}$</td>
<td>1.625 $\cdot 10^{-3}$</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>0.0005</td>
<td>0.0495</td>
<td>$\zeta$</td>
<td>1.713 $\cdot 10^{-7}$</td>
<td>1.105 $\cdot 10^{-7}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>2.918 $\cdot 10^{-4}$</td>
<td>1.474 $\cdot 10^{-4}$</td>
</tr>
<tr>
<td>1</td>
<td>50</td>
<td>0.005</td>
<td>0.245</td>
<td>$\zeta$</td>
<td>1.566 $\cdot 10^{-6}$</td>
<td>1.046 $\cdot 10^{-6}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>2.918 $\cdot 10^{-3}$</td>
<td>1.489 $\cdot 10^{-3}$</td>
</tr>
<tr>
<td>1</td>
<td>500</td>
<td>0.0005</td>
<td>0.2495</td>
<td>$\zeta$</td>
<td>1.499 $\cdot 10^{-7}$</td>
<td>9.997 $\cdot 10^{-8}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>2.918 $\cdot 10^{-4}$</td>
<td>1.462 $\cdot 10^{-4}$</td>
</tr>
<tr>
<td>0</td>
<td>50</td>
<td>5 $\cdot 10^{-7}$</td>
<td>2.45 $\cdot 10^{-5}$</td>
<td>$\zeta$</td>
<td>1.239 $\cdot 10^{-6}$</td>
<td>1.958 $\cdot 10^{-7}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>6.654 $\cdot 10^{-5}$</td>
<td>1.197 $\cdot 10^{-5}$</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>50</td>
<td>0.005</td>
<td>0.245</td>
<td>$\zeta$</td>
<td>2.719 $\cdot 10^{-7}$</td>
<td>1.125 $\cdot 10^{-7}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>6.460 $\cdot 10^{-6}$</td>
<td>1.446 $\cdot 10^{-6}$</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>500</td>
<td>0.0005</td>
<td>0.2495</td>
<td>$\zeta$</td>
<td>2.304 $\cdot 10^{-9}$</td>
<td>1.060 $\cdot 10^{-9}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$u$</td>
<td>3.714 $\cdot 10^{-8}$</td>
<td>1.270 $\cdot 10^{-8}$</td>
</tr>
</tbody>
</table>

5.1.2 With friction

In case of the linear dimensionless equations with friction, we have the following system of equations:

\[
\begin{align*}
\frac{\partial \zeta}{\partial t} + \frac{\partial u}{\partial x} &= 0 \quad (5.6a) \\
\frac{\partial u}{\partial t} + \frac{1}{\lambda^2} \frac{\partial \zeta}{\partial x} + ru &= 0 \quad (5.6b)
\end{align*}
\]

with boundary conditions

\[
\zeta(0, t) = \sin(t), \quad u(1, t) = 0. \quad (5.7)
\]

As periodic analytical solution we found in Section 3.1 equations (3.38):

\[
\begin{align*}
\zeta(x, t) &= \text{Re} \left\{ -ie^{it}\cos(\mu(1-x)) \right\} \quad (5.8a) \\
u(x, t) &= \text{Re} \left\{ \frac{1}{(1-h)\mu} e^{it\sin(\mu(1-x))} \right\} \quad (5.8b)
\end{align*}
\]

where $\mu$ is given by

\[
\mu = \pm \frac{\lambda \sqrt{2}}{2} \left( \sqrt{1 + \sqrt{1 + r^2}} - i \sqrt{1 + \sqrt{1 + r^2}} \right) \quad (5.9)
\]

In Table 4 the difference between the analytical and numerical solution is shown. Note that using the trapezoidal rule the difference for $\zeta$ and $u$ is more of the same order, than using Euler forward method. With trapezoidal the order of the difference is almost the same, where for the Euler forward method the difference for $u$ is much larger.
5.2 Numerical results for the moving boundary problem

The numerical simulation is performed for different bottom profiles. In Figure 15 the various bottom profiles used are plotted. Remember that $h(x)$ is the bottom profile, where we plot $-H + h$ in the figure, since this gives the coordinates (and hence 0 corresponds to the mean seawater level). In the dimensionless case we have always $H = 1$. Bottom 1 is the standard bottom with linear slope. Bottom 2 is a special profile, the profile with a bump in the middle of the domain. Bottom 3 is a test bottom for a more realistic profile, but it is far from a usable bottom which need to have $h(x = 0) = 0$. Bottom 4 is a bottom with linear slope in the beginning and a tidal flat at the end, but quite high above the mean seawater level. Bottom 6 is the most realistic bottom profile, with linear slope at the beginning, a tidal flat around the mean seawater level and then a concave upward part at the end of the domain. See also [11] for measurements and analysis of bottom profiles in the south San Francisco Bay.

The choice of the initial condition can be done in several ways. First, one can just choose an initial condition of which one thinks that it fits the situation. Another way, in case of trying to obtain the periodic solution is the following one. First a quite good guess for the initial condition is made. Then the simulation is run for one period to $t = 2\pi$. The obtained solution at $t = 2\pi$ is then taken as initial condition for a new simulation. In this way, the effect of the original condition is much less, since the initial oscillations due to the initial conditions are already damped for one period. Actually, if the same simulation is done with the new initial condition, it is just as starting at $t = 2\pi$, but the difference is that this initial solution is also a good guess for different, but similar cases. Such as changing some parameter values such as the number of gridpoints $N$, the friction coefficient $r$ and at least the timestep $\Delta t$. The advantage is that only once the extra simulation of one period has to be done, whereas starting at $t = 2\pi$...
with the old initial condition computing the extra period has to be done every simulation again. The parameters in the simulation has been taken realistic, except for the friction coefficient \( r \). Taking realistic friction values will lead to extremely small values for \( \Delta t \), and hence longer computing time. The value for \( r \) just in the simulations is taken one hundred times larger as the realistic value.

The parameters for the numerical simulations itself are chosen as follows. The number of grid-points is set to \( N = 400 \), but also \( N = 100 \) or intermediate values might be used. However, when we take \( N = 100 \), the results are sometimes a little less smooth, especially when there is very shallow water over a large area or when at some places the velocity changes very quickly. Most of the times \( \Delta t \) has to be very small, for instance \( \Delta t = 1 \cdot 10^{-5} \), but often is set a factor 10 smaller for more accurate results. Also note that \( \Delta t \) depends on the choice of the number of gridpoints \( N \).

The results for the bottom with linear slope are depicted in Figure 16. On the horizontal axis is the position \( x \) and on the vertical axis the time \( t \). On the left hand side is the solution for \( \zeta(x, t) \) given, and the corresponding solution for \( u(x, t) \) is on the right hand side. The white area inside figure corresponds to the on-shore area. The colors indicate the value of the function, where dark blue means low values and a red color indicate high values. The black lines in the colored area are the contour lines. For the solutions of other bottom profiles this works the same.

A horizontal cross-section gives the profile in \( x \)-direction at a certain time, whereas a vertical cross-section gives the behaviour in time at a certain place. For instance, the values at the vertical axis give the boundary condition and the values on the horizontal axis the initial condition. Also for two horizontal cross-sections (at \( t = \frac{3}{2} \pi \) and \( t = 2 \pi \)) the result is given in Figure 17.

There are some minor wiggles in the velocity \( u \) in an area at the interface for certain periods of time. These wiggles are visible as black shuddering lines in a neighborhood of the interface, indicating that the variable is not smooth. In general the behaviour looks like the analytical solution which a flat bottom. The most important differences occur near the landward boundary. First of all it is moving, but also the sea surface elevation \( \zeta(x, t) \) and the velocity \( u(x, t) \) behave different near the boundary. Near the landward boundary, \( \zeta \) reacts slower to the forcing (lower than surrounding when tide rises and higher than surrounding when tide falls), and \( u \) is always curved towards zero velocity.

Figure 16: Solution for a linear sloping bottom.
Figure 17: Solution for linear sloping bottom at two time levels.
Figure 18: The boundary $\Gamma(t)$ for the linear sloping bottom.

The moving boundary is depicted in Figure 18, but can also be seen from the picture of the solution in Figure 16, since it is the boundary of the domain. The curve is smooth, but when one zooms in, for example on the indicated area, one can detect some irregularities, as shown in Figure 19(a). The irregularities can better be visualised by considering the time derivative of $\Gamma(t)$, as depicted in Figure 19(b). This derivative is obtained by using numerical central differences after the simulation has been run. The unevennesses take place at the moment the boundary crosses the line of a gridcell change. However, the irregularities do not occur due to numerical errors in the continuation of the boundary velocity or the reinitialisation of the level set function $\phi$. It appears that the derivative of $\Gamma$ is identical to calculated velocity $u$ at the boundary at each timestep, as it should be from the model we use. So the irregularities in $\Gamma(t)$ stem from the calculation of the velocity due to the extrapolation, in combination with the change of a boundary cell.

The choice of $\Delta t$ is essential, since for $\theta = 0.55$ and too large values of $\Delta t$ there enter unstable wiggles from the seaside into the domain.
Figure 19: Irregularities in the boundary $\Gamma(t)$. 

(a) Zoomed area with visible irregularities.

(b) Time derivative indicating irregularities at short times.
In case of a bottom with a bump (bottom 2), the result is plotted in Figure 20. The derivative \( \frac{\partial \zeta}{\partial x} \) attains values between -2 and 2 and the minimum depth at the bump is around 0.063, which corresponds to 0.63 meters. The bottom profile is given by
\[
h = 20.33x^4 + 87.37x^3 - 164.54x^2 + 71.12x - 0.6.
\]

The extreme values for the velocity occur at the bump, and the velocities at these locations are plotted in Figure 21.

For bottom 3, the solution is depicted in Figure 22. In contrast with the previous cases, the shape of the moving boundary is not symmetric: due to the more flat shaped bottom in this area, the water front sticks to bottom. Hence, after the high tide, the water front moves much slower back in the direction of the sea. At a certain moment, the new tide front from the sea arrives at the moving boundary and the latter catch up its original speed again. This effect will also happen for the bottom profiles 4 and 6.

For bottom 4, the result is presented in Figure 23 and 24. Normally the code is run up to \( t = 3\pi \), and it is assumed that periodic solution has been reached. But for this bottom the solution is also computed for an additional period of \( 2\pi \), in order to check this assumption. From the...
Figure 22: Bottom 3.

Figure 23: Bottom 4, periodic part.

Figure 24: Bottom 4 with initial oscillation.
computations it is found that indeed the difference between the period for $t \in [\pi, 3\pi]$ and the period $t \in [3\pi, 5\pi]$ is very small.
For this bottom, the movement of the boundary is similar to that in the case of bottom 3, but the solution is different. The behaviour of the solution near the seaward boundary is in this case more similar to that of a linear sloping bottom, whereas the irregular part of the solution is more shifted towards the landward boundary.
For the realistic bottom 6, the solution is given in Figure 25. In this case, the solution stays more similar to the solution of bottom 1, but the part of the solution near the moving boundary is even more irregular than for bottom 4. This is due to the bottom profile, since the flat part of the bottom is even more flat (but still sloping) and because the boundary moves in both the concave and the convex part of the bottom.

Figure 25: Realistic bottom 6.
6 Conclusion

In this thesis the ability of applying the level set method on the shallow water equations was investigated. The water motion in an tidal embayment is modeled by the depth-averaged shallow water equations. The dimensionless equations can be solved analytically only for a few bottom profiles, both on the fixed grid and with a moving boundary. Using a numerical code and including the level set method, the equations can be solved for every type of bottom which give rise to only one interface point. The system is solved by using a finite volume discretisation with a central difference scheme.

The obtained solution is quite satisfactory. The level set method is well able to capture the moving boundary. For the solution there are some little complications at the boundary, since for certain values of time, the velocity has some small wiggles near the landward boundary. However, for smaller values of $\Delta t$ and a higher number of gridpoints, the wiggles tends to decrease, but still they are present.

One also can take advective terms into account. First results indicates that the results are qualitative the same as without advection.

7 Recommendations

Some aspects can be improved. At first the wiggles at the boundary may be smoothed somehow (since the trend seems to be fine) in order to get a smooth solution. Another option may be to diagonalise the system at each time step, so that the characteristic directions can be obtained, and it will be possible then to use upwind difference instead of central differences. Also this gives the opportunity to apply slope limiters for the differences. This also will lead to non oscillating solutions.

An interesting extension is the introduction of a trace system for new boundaries. This will give the possibility of catching topological changes in the domain, such may arise when the bump in the used bottom with a bump is enlarged, such that it dries during low water. In that case the code can be run on two separated domains until the water rises above the bump again, when the tracing system must be able to merge the domains together again.

An obvious other step is the extension to two spatial dimensions, such that no longer a cross-sectionally but only a depth-averaged system can be considered. The moving boundary will be a moving curve (like a real coastline), instead of a single point. It might be useful to use a finite-element discretisation instead of a finite volume one, since it is easier to use the finite-element method for a problem with complex or irregular domains. However, the idea of the code will remain the same.

Changing the boundary condition is another improvement. Until now, only one tidal constituent has been prescribed at the seaward boundary. However, other tidal components and especially overides may be added to the boundary condition. This might have interesting consequences for the solution. As well, the sediment transport might be computed from the solution for $u$ (if it is smooth), and the overides might have a strong impact in this case.

Also the actual code can be made more efficient and faster. Some parameters might be chosen such that the same result is obtained, but costs less computational time. For instance, the pseudo timestepping in the level set part of the code might be optimised. Another way to improve the efficiency is to extend the velocity only in a small band around the interface instead of the full domain. Then also the reinitialisation has to be done only in this small band. And also the reinitialisation can be limited to only once in a couple of timesteps, instead of doing it every timestep.
The code might be improved in the problem when the boundary moves into another gridcell. At the moment the change of gridcells is done at the change of the time level. However it might be possible to predict the time at which the boundary crosses the point of a gridcell change, by linear interpolation of the boundary point and the speed of the boundary. Then the value for the new cell at the new time level can be integrated only over the period of time in which the boundary is in the new cell. This might give even more accurate results.
References


A Depth averaging of the model equations

In this section we want to derive a two dimensional system from the three dimensional Boussinesq equations, taking out the vertical components by averaging over the depth. Consider the equations, which come from \[12\], Chapter 3.

\[
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \tag{A.1a}
\]

\[
\frac{Du}{Dt} = -\frac{1}{\rho_0} \frac{\partial p}{\partial x} + \frac{1}{\rho_0} \left( \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} \right), \tag{A.1b}
\]

\[
\frac{Dv}{Dt} = -\frac{1}{\rho_0} \frac{\partial p}{\partial y} + \frac{1}{\rho_0} \left( \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{yz}}{\partial z} \right), \tag{A.1c}
\]

\[
\frac{Dw}{Dt} = -\frac{1}{\rho_0} \frac{\partial p'}{\partial z} - \frac{\rho' g}{\rho_0} + \frac{1}{\rho_0} \left( \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} \right), \tag{A.1d}
\]

where the pressure \( p \) and density \( \rho \) are given by

\[
p = p_0 + \rho_0 g (z + \zeta) + p'(x, y, z, t), \tag{A.2}
\]

\[
\rho = \rho_0 + \rho'(x, y, z, t). \tag{A.3}
\]

In the equations (A.1) \( u, v \) and \( w \) are the velocities in \( x, y \) and \( z \)-direction respectively, \( \zeta \) is the sea surface elevation. As parameters, we have the molecular diffusion \( \nu \), the reference pressure \( p_0 \) and the reference density \( \rho_0 \). The tensor \( \tau_{ij} \) denotes the normal and shear stresses from direction \( i \) onto direction \( j \), but is also assumed to be symmetric. From these equations we get the well known Boussinesq equations, if assume that \( \tau \) can be written as

\[
\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \tag{A.4}
\]

since then we have

\[
\frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} = \mu \left( \frac{\partial}{\partial x} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \right), \tag{A.5}
\]

\[
= \mu \left( 2 \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + \frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 w}{\partial x \partial z} \right), \tag{A.6}
\]

\[
= \mu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) \tag{A.7}
\]

\[
= \mu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right), \tag{A.8}
\]

where we used the continuity equation (A.1a) for the last step. The kinematic viscosity \( \nu = \frac{\mu}{\rho_0} \), is the dynamic viscosity divided by the density. Similar expressions hold for the \( y \) and \( z \)-direction.

Using this notation the system (A.1) reduces to

\[
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0, \tag{A.9a}
\]

\[
\frac{Du}{Dt} = -\frac{1}{\rho_0} \frac{\partial p}{\partial x} + \nu \nabla^2 u, \tag{A.9b}
\]

\[
\frac{Dv}{Dt} = -\frac{1}{\rho_0} \frac{\partial p}{\partial y} + \nu \nabla^2 v, \tag{A.9c}
\]

\[
\frac{Dw}{Dt} = -\frac{1}{\rho_0} \frac{\partial p'}{\partial z} - \frac{\rho' g}{\rho_0} + \nu \nabla^2 w. \tag{A.9d}
\]
Although the equations are solved more often, we will not use them, since the original system works better for the boundary conditions at the bottom and the top.

We have a rectangular domain between $x = 0$, $x = L$, $y = 0$ and $y = B$, see Figure 26.

![Figure 26: Situation of the domain.](image)

As boundary conditions at $y = 0$ and $y = B$, we have $u = 0$, $v = 0$ (a no-slip condition). We have $(1 - h + \zeta)u = 0$ and $v = 0$ at $x = L$. At $x = 0$, we prescribe the sea level $\zeta(0, y, t) = \zeta_0(y, t)$. For the surface and bottom, we have the kinematic boundary conditions (A.13) and (A.14) and the dynamic conditions for the shear stress (A.31) and (A.33). These last mentioned boundary conditions will be discussed later on. These boundary conditions are used more often, see for instance Section 4.6 in [12]. We assume we have initial values for the unknowns $u$, $v$, $w$ and $\zeta$.

To get the depth-averaged equations, we integrate equation (A.1a)-(A.1d) over the depth. Later on we divide the momentum equation by the depth. First integrate the continuity equation (A.1a) over the depth:

$$\int_{h-H}^{\zeta} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \, dz = 0. \quad (A.10)$$

We apply Leibniz’ integration rule to take the derivatives outside the integration, for example

$$\frac{\partial}{\partial x} \int_{h-H}^{\zeta} u \, dz = \int_{h-H}^{\zeta} \frac{\partial u}{\partial x} \, dz + u|_{z=\zeta} \frac{\partial \zeta}{\partial x} - u|_{z=h-H} \frac{\partial h}{\partial x}. \quad (A.11)$$

Hence the continuity equation becomes

$$\frac{\partial}{\partial x} \int_{h-H}^{\zeta} u \, dz + \frac{\partial}{\partial y} \int_{h-H}^{\zeta} v \, dz + \overline{w|_{z=\zeta} - w|_{z=h-H} - u|_{z=\zeta} \frac{\partial \zeta}{\partial x}} + u|_{z=h-H} \frac{\partial h}{\partial x} - v|_{z=\zeta} \frac{\partial \zeta}{\partial y} + v|_{z=h-H} \frac{\partial h}{\partial y} = 0. \quad (A.12)$$

At the bottom we require that a fluid element at the boundary must stay at the boundary (the so-called kinematic boundary condition). Then the vertical component of the velocity has to equal the change of the bottom,

$$w|_{z=h-H} = \frac{D}{Dt} (h - H) = \frac{\partial h}{\partial t} + u \frac{\partial h}{\partial x} + v \frac{\partial h}{\partial y}. \quad (A.13)$$
and for the sea surface we have
\[ w|_{z=\xi} = \frac{D}{Dt}\xi \]
\[ = \frac{\partial \xi}{\partial t} + u \frac{\partial \xi}{\partial x} + v \frac{\partial \xi}{\partial y}. \]  \hfill (A.14)

Substituting these boundary conditions into equation (A.12) we obtain
\[ \frac{\partial}{\partial x} \int_{h-H}^{\xi} u\,dz + \frac{\partial}{\partial y} \int_{h-H}^{\xi} v\,dz + \frac{\partial \xi}{\partial t} - \frac{\partial h}{\partial t} = 0. \]  \hfill (A.15)

Define depth-averaged velocities \( \bar{u} \) and \( \bar{v} \) as
\[ \bar{u} = \frac{1}{H-h+\xi} \int_{h-H}^{\xi} u(x,y,z,t)\,dz, \]  \hfill (A.16)
\[ \bar{v} = \frac{1}{H-h+\xi} \int_{h-H}^{\xi} v(x,y,z,t)\,dz. \]  \hfill (A.17)

By using these in equation (A.15), the depth-averaged continuity equation is given by
\[ \frac{\partial \xi}{\partial t} - \frac{\partial h}{\partial t} + \frac{\partial}{\partial x} ((H-h+\xi)\bar{u}) + \frac{\partial}{\partial y} ((H-h+\xi)\bar{v}) = 0. \]  \hfill (A.18)

Note that if the bottom is assumed to be fixed or changes very slowly compared to \( \xi \), the term \( \frac{\partial h}{\partial t} \) may be ignored.

**Momentum equations**

We assume the system is in hydrostatic equilibrium and has constant density \( \rho_0 \). From this it follows that \( p' = 0 \) and
\[ p = p_0 + \rho_0(-z + \xi). \]  \hfill (A.19)

Integrating the momentum equation (A.1b) in the \( x \)-direction over the depth and using the hydrostatic equilibrium (A.19) gives
\[ \int_{h-H}^{\xi} \frac{Du}{Dt} \,dz = \int_{h-H}^{\xi} -g \frac{\partial \xi}{\partial x} + \frac{1}{\rho_0} \left( \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} \right) \,dz. \]  \hfill (A.20)

We rewrite the left-hand side of (A.20) as follows, using \( u \frac{\partial u}{\partial x} = \frac{1}{2} \frac{\partial (u^2)}{\partial x} \) and integration by parts on the \( w \)-term,
\[ \int_{h-H}^{\xi} \frac{Du}{Dt} \,dz = \int_{h-H}^{\xi} \left( \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) \,dz \]
\[ = \frac{\partial}{\partial t} \int_{h-H}^{\xi} u \,dz - u|_{z=\xi} \frac{\partial \xi}{\partial t} + u|_{z=h-H} \frac{\partial h}{\partial t} \]
\[ + \frac{1}{2} \frac{\partial}{\partial x} \int_{h-H}^{\xi} u^2 \,dz - \frac{1}{2} u^2|_{z=\xi} \frac{\partial \xi}{\partial x} + \frac{1}{2} u^2|_{z=h-H} \frac{\partial h}{\partial x} \]
\[ + \int_{h-H}^{\xi} v \frac{\partial u}{\partial y} \,dz + [wu]|_{z=h-H} - \int_{h-H}^{\xi} u \frac{\partial w}{\partial z} \,dz. \]  \hfill (A.21)
Next we use the continuity equation,

\[- \int_{h-H}^{\zeta} \frac{\partial w}{\partial z} \, dz = \int_{h-H}^{\zeta} \frac{\partial u}{\partial x} \, dz + \int_{h-H}^{\zeta} \frac{\partial v}{\partial y} \, dz. \]  

(A.22)

Substitution of equation (A.22) into equation (A.21) and then applying Leibniz’ integration rule results in

\[\int_{h-H}^{\zeta} \frac{Du}{Dt} \, dz = \frac{\partial}{\partial t} \int_{h-H}^{\zeta} u \, dz - u|_{z=\zeta} \frac{\partial \xi}{\partial t} + u|_{z=h-H} \frac{\partial h}{\partial t} + \frac{\partial}{\partial x} \int_{h-H}^{\zeta} u^2 \, dz - u^2|_{z=\zeta} \frac{\partial \xi}{\partial x} + u^2|_{z=h-H} \frac{\partial h}{\partial x} + \int_{h-H}^{\zeta} \frac{\partial uv}{\partial y} \, dz + [uw]|_{z=h-H} \]  

(A.23)

Subsequently we apply the boundary conditions (A.13) and (A.14), both multiplied by \( u \), and the boundary terms cancel, to arrive at

\[\int_{h-H}^{\zeta} \frac{Du}{Dt} \, dz = \frac{\partial}{\partial t} \int_{h-H}^{\zeta} u \, dz + \frac{\partial}{\partial x} \int_{h-H}^{\zeta} u^2 \, dz + \frac{\partial}{\partial y} \int_{h-H}^{\zeta} uv \, dz. \]  

(A.24)

Substitution of equation (A.24) into (A.20) gives, noting that \( \frac{\partial \xi}{\partial x} \) does not depend on \( z \),

\[\frac{\partial}{\partial t} \int_{h-H}^{\zeta} u \, dz + \frac{\partial}{\partial x} \int_{h-H}^{\zeta} u^2 \, dz + \frac{\partial}{\partial y} \int_{h-H}^{\zeta} uv \, dz = -g(H - \xi + \zeta) \frac{\partial \xi}{\partial x} + \frac{1}{\rho_0} \int_{h-H}^{\zeta} \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} \, dz. \]  

(A.25)

Writing out the right-hand side using Leibniz’ rule gives

\[\frac{\partial}{\partial t} \int_{h-H}^{\zeta} u \, dz + \frac{\partial}{\partial x} \int_{h-H}^{\zeta} u^2 \, dz + \frac{\partial}{\partial y} \int_{h-H}^{\zeta} uv \, dz = -g(H - \xi + \zeta) \frac{\partial \xi}{\partial x} + \frac{1}{\rho_0} \int_{h-H}^{\zeta} \tau_{xx} \, dz + \frac{1}{\rho_0} \frac{\partial}{\partial y} \int_{h-H}^{\zeta} \tau_{xy} \, dz \]

(A.26)

\[+ \frac{1}{\rho_0} \left( - \tau_{xx}|_{z=\xi} \frac{\partial \xi}{\partial x} + \tau_{xx}|_{z=h-H} \frac{\partial h}{\partial x} - \tau_{xy}|_{z=\xi} \frac{\partial \xi}{\partial y} + \tau_{xy}|_{z=h-H} \frac{\partial h}{\partial y} + [\tau_{xx}]|_{z=h-H} \right) \]

We have a parametrisation of the dynamic boundary condition at the bottom

\[\tau_b = \rho_0 c_d |\bar{u}_b| \bar{u}_b, \]  

(A.27)
where \( \vec{u}_b \) is the tangential velocity at the bottom and \( \vec{\tau}_b \) is defined as

\[
\vec{\tau}_b = \mathbf{n} \cdot \mathbf{\tau},
\]

where \( \mathbf{n} \) is the normal vector to the bottom.

\[
\mathbf{n} = \begin{pmatrix} -\frac{\partial h}{\partial x} \\ -\frac{\partial h}{\partial y} \\ 1 \end{pmatrix}
\]

Substitution of \( \mathbf{\tau} = \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{pmatrix} \) into equation (A.28) and (A.29) gives the following final boundary condition at the bottom

\[
\begin{pmatrix} -\frac{\partial h}{\partial x} \\ -\frac{\partial h}{\partial y} \\ -\frac{\partial h}{\partial z} \end{pmatrix} = \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{pmatrix} \begin{pmatrix} u_b \\ v_b \\ w_b \end{pmatrix} \quad \text{at} \quad z = h - H.
\]

At the surface \( z = \zeta \) we prescribe we have no shear forces (wind or induced by other sources), so we have

\[
\vec{\tau}_s = 0,
\]

or equivalently

\[
\begin{pmatrix} -\frac{\partial \zeta}{\partial x} \tau_{xx} - \frac{\partial \zeta}{\partial y} \tau_{xy} + \tau_{xx} \\ -\frac{\partial \zeta}{\partial x} \tau_{yx} - \frac{\partial \zeta}{\partial y} \tau_{yy} + \tau_{yy} \\ -\frac{\partial \zeta}{\partial x} \tau_{zx} - \frac{\partial \zeta}{\partial y} \tau_{yz} + \tau_{zz} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{at} \quad z = \zeta.
\]

Substitution of the two boundary conditions (A.31) and (A.33) into equation (A.26) gives

\[
\frac{\partial}{\partial t} \int_{h-H}^{\zeta} u \, dz + \frac{\partial}{\partial x} \int_{h-H}^{\zeta} u^2 \, dz + \frac{\partial}{\partial y} \int_{h-H}^{\zeta} uv \, dz = -g (H - h + \zeta) \frac{\partial \zeta}{\partial x} + \frac{1}{\rho_0} \frac{\partial}{\partial x} \int_{h-H}^{\zeta} \tau_{xx} \, dz + \frac{1}{\rho_0} \frac{\partial}{\partial y} \int_{h-H}^{\zeta} \tau_{xy} \, dz + 0 - cd |\vec{u}_b| u_b
\]

We write for the depth-averaged quantities, for example \( u \),

\[
u = \bar{u} + u',
\]

where

\[
\bar{u} = \frac{1}{H - h + \zeta} \int_{h-H}^{\zeta} u \, dz,
\]

\[
0 = \int_{h-H}^{\zeta} u' \, dz.
\]

Then we get

\[
\int_{h-H}^{\zeta} uv \, dz = \int_{h-H}^{\zeta} \bar{u} \bar{v} + \bar{u} u' + u' \bar{v} + u' u' \, dz
\]

\[
= (H - h + \zeta) \bar{u} \bar{v} + 0 + 0 + \int_{h-H}^{\zeta} u' u' \, dz,
\]

53
Now we make the closure assumption that

\[ \int_{h-H}^{c} u^2 \, dz = (H - h + \zeta)\bar{\pi}^2 + \int_{h-H}^{c} u' \, dz. \quad (A.36) \]

Equation (A.34) now becomes

\[
\frac{\partial}{\partial t}(H - h + \zeta)\bar{\pi} + \frac{\partial}{\partial x}(H - h + \zeta)\bar{\pi}^2 + \frac{\partial}{\partial x} \int_{h-H}^{c} u' \, dz \\
+ \frac{\partial}{\partial y}(H - h + \zeta)\bar{\pi} + \frac{\partial}{\partial y} \int_{h-H}^{c} u' \, dz \\
= -g(H - h + \zeta)\frac{\partial c}{\partial x} + \frac{1}{\rho_0} \frac{\partial}{\partial x} \int_{h-H}^{c} \tau^{xx} \, dz + \frac{1}{\rho_0} \frac{\partial}{\partial y} \int_{h-H}^{c} \tau^{xy} \, dz \\
+ 0 - c_d |\bar{u}_b| u_b
\]

Writing out the derivatives, the left-hand side of (A.37) becomes

\[
\frac{\partial}{\partial t}(H - h + \zeta)\bar{\pi} + \frac{\partial}{\partial x}(H - h + \zeta)\bar{\pi}^2 + \frac{\partial}{\partial x} \int_{h-H}^{c} u' \, dz \\
+ \frac{\partial}{\partial y}(H - h + \zeta)\bar{\pi} + \frac{\partial}{\partial y} \int_{h-H}^{c} u' \, dz \\
= (H - h + \zeta)\frac{\partial \bar{\pi}}{\partial t} + (H - h + \zeta)\frac{\partial \bar{\pi}}{\partial x} + (H - h + \zeta)\frac{\partial \bar{\pi}}{\partial y} + \frac{\partial}{\partial x} \int_{h-H}^{c} u' \, dz \\
+ \frac{\partial}{\partial y} \int_{h-H}^{c} u' \, dz \\
+ \frac{\partial}{\partial x} (H - h + \zeta)\frac{\partial \bar{\pi}}{\partial x} + \frac{\partial}{\partial y} (H - h + \zeta)\frac{\partial \bar{\pi}}{\partial y} \\
= (H - h + \zeta) \left( \frac{\partial \bar{\pi}}{\partial t} + \bar{\pi} \frac{\partial \bar{\pi}}{\partial x} + \bar{\pi} \frac{\partial \bar{\pi}}{\partial y} \right) + \frac{\partial}{\partial x} \int_{h-H}^{c} u' \, dz + \frac{\partial}{\partial y} \int_{h-H}^{c} u' \, dz \\
+ \pi \left( \frac{\partial}{\partial t} (H - h + \zeta) + \frac{\partial}{\partial x} (H - h + \zeta)\pi + \frac{\partial}{\partial y} (H - h + \zeta)\pi \right).
\]

Now we apply the depth integrated continuity equation (A.18) to get rid of a couple of terms. The left-hand side becomes

\[ (H - h + \zeta) \left( \frac{\partial \bar{\pi}}{\partial t} + \bar{\pi} \frac{\partial \bar{\pi}}{\partial x} + \bar{\pi} \frac{\partial \bar{\pi}}{\partial y} \right) + \frac{\partial}{\partial x} \int_{h-H}^{c} u' \, dz + \frac{\partial}{\partial y} \int_{h-H}^{c} u' \, dz. \quad (A.40) \]

Using equation (A.39) and (A.40) in equation (A.37) we arrive at

\[
(H - h + \zeta) \left( \frac{\partial \bar{\pi}}{\partial t} + \bar{\pi} \frac{\partial \bar{\pi}}{\partial x} + \bar{\pi} \frac{\partial \bar{\pi}}{\partial y} \right) + \frac{\partial}{\partial x} \int_{h-H}^{c} u' \, dz + \frac{\partial}{\partial y} \int_{h-H}^{c} u' \, dz \\
= -g(H - h + \zeta)\frac{\partial c}{\partial x} + \frac{1}{\rho_0} \frac{\partial}{\partial x} \int_{h-H}^{c} \tau^{xx} \, dz + \frac{1}{\rho_0} \frac{\partial}{\partial y} \int_{h-H}^{c} \tau^{xy} \, dz \\
+ 0 - c_d |\bar{u}_b| u_b.
\]

Now we make the closure assumption that

\[
\frac{1}{\rho_0} \int_{h-H}^{c} \tau^{xx} \, dz - \int_{h-H}^{c} u' \, dz = 2K_h \frac{\partial}{\partial x} ((H - h + \zeta)\pi) \\
\frac{1}{\rho_0} \int_{h-H}^{c} \tau^{xy} \, dz - \int_{h-H}^{c} u' \, dz = K_h \left( \frac{\partial}{\partial x} ((H - h + \zeta)\pi) + \frac{\partial}{\partial y} ((H - h + \zeta)\pi) \right),
\]

54
where \( K_h \) is a diffusion coefficient. By this we assume that the derivations of the variations of the velocities and the derivatives of the shear stresses behave as a diffusion of the mean velocity. This assumption is common practice, see for instance p. 24, in [8].

Using this assumption in equation (A.41) we have the equation

\[
(H - h + \zeta) \left( \frac{\partial \bar{\pi}}{\partial t} + \bar{\pi} \frac{\partial \bar{\pi}}{\partial x} + \bar{\pi} \frac{\partial \bar{\pi}}{\partial y} \right)
= -g(H - h + \zeta) \frac{\partial \zeta}{\partial x} + 2K_h \frac{\partial^2}{\partial x^2}((H - h + \zeta)\bar{\pi})
- c_d |\bar{u}_b| u_b + K_h \frac{\partial}{\partial y} \left( \frac{\partial}{\partial x}((H - h + \zeta)\bar{\pi}) + \frac{\partial}{\partial y}((H - h + \zeta)\bar{\pi}) \right). \tag{A.44}
\]

So as the final equation we have

\[
(H - h + \zeta) \left( \frac{\partial \bar{\pi}}{\partial t} + \bar{\pi} \frac{\partial \bar{\pi}}{\partial x} + \bar{\pi} \frac{\partial \bar{\pi}}{\partial y} \right)
= -g(H - h + \zeta) \frac{\partial \zeta}{\partial x} - c_d |\bar{u}_b| u_b \tag{A.45}
+ 2K_h \frac{\partial^2}{\partial x^2}((H - h + \zeta)\bar{\pi}) + K_h \frac{\partial^2}{\partial y^2}((H - h + \zeta)\bar{\pi}) + K_h \frac{\partial^2}{\partial x \partial y}((H - h + \zeta)\bar{\pi}).
\]

We can divide this equation by the depth \( H - h + \zeta \) and obtain the depth-averaged momentum equation in the \( x \)-direction,

\[
\frac{\partial \bar{\pi}}{\partial t} + \bar{\pi} \frac{\partial \bar{\pi}}{\partial x} + \bar{\pi} \frac{\partial \bar{\pi}}{\partial y} + g \frac{\partial \zeta}{\partial x} + \frac{c_d}{H - h + \zeta} |\bar{u}_b| u_b \tag{A.46}
= \frac{K_h}{H - h + \zeta} \left( \frac{\partial^2}{\partial x^2}(2(H - h + \zeta)\bar{\pi}) + \frac{\partial^2}{\partial y^2}((H - h + \zeta)\bar{\pi}) + \frac{\partial^2}{\partial x \partial y}((H - h + \zeta)\bar{\pi}) \right).
\]

For the momentum equation in the \( y \)-direction we can do a similar integration and manipulation to obtain a similar depth integrated momentum equation.

### A.1 Bottom stress

As final part we have to link the bottom stress to only depth-averaged quantities. Until now we still have the velocity at the bottom. We assume that we can rewrite the friction as

\[
\frac{c_d}{H - h + \zeta} |\bar{u}_b| u_b = \frac{\tilde{c}_d}{H - h + \zeta} \bar{\pi} |\bar{u}| \bar{u}. \tag{A.47}
\]

So we change the friction parameter and take the depth-averaged velocities instead of the velocities at the bottom.
Calculations for case 2

In this section we describe the discretisation for the boundary cell for case 2 for the problem with a moving boundary. The discretisation for case 1 was treated in Sections 4.3.1 and 4.3.2.

Fixed cell averages in space

For the second case we use analogous principles as for the first case, but get a slightly different discretisation. Define

\[ \zeta_{i+1} = \frac{1}{a \Delta x} \int_{(i-\frac{1}{2}) \Delta x}^{(i+\frac{1}{2}) \Delta x} \zeta(x, t) \, dx, \]  
\[ u_{i+1} = \frac{1}{a \Delta x} \int_{(i-\frac{1}{2}) \Delta x}^{(i+\frac{1}{2}) \Delta x} u(x, t) \, dx. \]  

Now integrate the continuity equation (4.35a) over cell \( C_{i+1} \),

\[ \int_{C_{i+1}} \partial \zeta \partial t + \partial \partial x ((1 - h + \varepsilon \zeta) u) \, dx = 0, \]  
\[ \int_{C_{i+1}} \zeta^j_{i+1} - \zeta^j_{i+1} \Delta t \partial x + \int_{C_{i+1}} \partial \partial x ((1 - h + \varepsilon \zeta) u) \, dx = 0, \]  
\[ a \frac{\Delta x}{\Delta t} (\zeta^j_{i+1} - \zeta^j_{i+1}) + [(1 - h + \varepsilon \zeta) u]_{C_{i+1}} = 0. \]  

At the moving boundary at \( x = \Gamma \) we have the boundary condition (4.37) and hence we obtain for the continuity equation

\[ a \frac{\Delta x}{\Delta t} (\zeta^j_{i+1} - \zeta^j_{i+1}) = (1 - h + \varepsilon \zeta) u \big|_{x = x_i + \frac{1}{2}} \]
\[ = \frac{1}{2} ((1 - h_i + \varepsilon \zeta_i) u_i + (1 - h_{i+1} + \varepsilon \zeta_{i+1}) u_{i+1}). \]  

For the momentum equation we integrate over cell \( C_{i+1} \) and obtain

\[ a \frac{\Delta x}{\Delta t} (u^j_{i+1} - u^j_{i+1}) + \frac{1}{2} [\zeta]_{C_{i+1}} + \int_{C_{i+1}} \frac{r}{1 - h + \varepsilon \zeta} u \, dx = 0. \]  

Rewrite again, using the midpoint rule, gives

\[ a \frac{\Delta x}{\Delta t} (u^j_{i+1} - u^j_{i+1}) = -\frac{1}{2} \left( \zeta \big|_{x=\Gamma} - \zeta \big|_{x = x_i + \frac{3}{2}} \right) - \frac{r}{1 - h_i + \varepsilon \zeta_{i+1} + u_{i+1}.} \]

Using that \( 1 - h + \varepsilon \zeta = 0 \) at \( x = \Gamma = x_i + \frac{3}{2} + a \Delta x \) and linear interpolation of the bottom we have

\[ \zeta \big|_{x=\Gamma} = \frac{h - 1}{\varepsilon} = \frac{1}{\varepsilon} \left( (h_{i+2} - h_{i+1}) \frac{(a - \frac{1}{2}) \Delta x}{\Delta x} + h_{i+1} - 1 \right) \]
\[ = \frac{1}{\varepsilon} \left( \left( \frac{3}{2} - a \right) h_{i+1} + \left( a - \frac{1}{2} \right) h_{i+2} - 1 \right). \]  

With linear interpolation we get

\[ \zeta \big|_{x = x_i + \frac{3}{2}} = \frac{1}{2} (\zeta_i + \zeta_{i+1}). \]
Now by substitution of equation (B.11) and (B.12) into equation (B.9) we arrive at the discretisation for the momentum equation given by

$$a \frac{\Delta x}{\Delta t} \left( u^{i+1}_{j} - u^{i}_{j} \right) = -\frac{1}{\varepsilon \lambda^2} \left( \frac{3}{2} - a \right) h_{i+1} + \left( a - \frac{1}{2} \right) h_{i+2} - 1 \right) + \frac{1}{\lambda^2} (\zeta_i + \zeta_{i+1}) - \frac{r a \Delta x}{1 - h_{i+1} + \varepsilon \zeta_{i+1}} u_{i+1}. \quad (B.13)$$

**Moving cell averages**

We will describe the differences from the previous derivation when moving the cell averages to the center of the actual boundary cell. For the second case we found for the continuity equation

$$a \frac{\Delta x}{\Delta t} \left( \zeta^{i+1}_{j} - \zeta^{i}_{j} \right) = (1 - h + \varepsilon \zeta) u_{x=x_{i+\frac{1}{2}}}. \quad (B.14)$$

For the value at $x = x_{i+\frac{1}{2}}$ we need to interpolate, but the value of cell $C_{i+1}$ is not at $x_{i+1}$ but in the middle of the cell. The situation is sketched in Figure 27. The linear interpolation formula is given by

$$f_{\text{lin}}(x) = (f_{i+1} - f_i) \frac{1}{(\frac{1}{2} + \frac{1}{2}) a \Delta x} (x - x_i) + f_i, \quad (B.15)$$

$$f_{\text{lin}}(x_{i+\frac{1}{2}}) = (f_{i+1} - f_i) \frac{1}{1 + a} + f_i$$

$$= \frac{1}{1 + a} (a f_i + f_{i+1}). \quad (B.16)$$

One can check that for $a = 1$ we have the usual average value. Using this formula, we get the discretisation for the continuity equation

$$a \frac{\Delta x}{\Delta t} \left( \zeta^{i+1}_{i+1} - \zeta^{i}_{i+1} \right) = \frac{1}{1 + a} \left( a(1 - h + \varepsilon \zeta) u_i + (1 - h_{i+1} + \varepsilon \zeta_{i+1}) u_{i+1} \right). \quad (B.17)$$

For the momentum equation we found equation (B.9)

$$a \frac{\Delta x}{\Delta t} \left( u^{i+1}_{j} - u^{i}_{j} \right) = -\frac{1}{\lambda^2} \left( \zeta|_{x=x_{i+\frac{1}{2}}} - \zeta|_{x=x_{i+1}} \right) - \frac{r a \Delta x}{1 - h_{i+1} + \varepsilon \zeta_{i+1}} u_{i+1}, \quad (B.18)$$

and equation (B.11), which remains unchanged. Using the formula for the interpolation at the boundary cell (B.16) we get

$$\zeta|_{x=x_{i+\frac{1}{2}}} = \frac{1}{1 + a} \left( a \zeta_i + \zeta_{i+1} \right). \quad (B.19)$$

![Figure 27: Situation at the interface, case 2, for moving averages.](image-url)
Now by substitution of equation (B.11) and (B.19) into equation (B.18) we arrive at the discretisation for the momentum equation given by

$$a \frac{\Delta x}{\Delta t} \left( u_{i+1}^j - u_{i+1}^{j-1} \right) = -\frac{1}{\varepsilon \lambda^2} \left( \left( \frac{3}{2} - a \right) h_{i+1} + \left( a - \frac{1}{2} \right) h_{i+2} - 1 \right) + \frac{1}{\lambda^2 \left( 1 + a \right)} (a \zeta_i + \zeta_{i+1}) - \frac{r}{1 - h_{i+1} + \varepsilon \zeta_{i+1}} u_{i+1}. \quad (B.20)$$
C MATLAB-code

clc
verder=0;
if verder
    IC = [BZ(n);Y(:,end)];
    Y1=Y;
    phi01 =phi0;
    phi0n1 =phi0n;
    %ICphi = phi(:,end);
    ICphi = phi;
    BZ1=BZ;
    T1=T;
else
    clear all
    verder=0;
end
tic

g_dim=9.81; %gravitatie constante
H_dim = 10; %hoogte van referentie vlak
Amp=1; %Amplitude van oppervlak
cd_dim = .25; %wrijvings coefficient, aangepast voor linearisatie
    %0.0025 standaard, goed LS .25
hmin = 0.001; %extra minimale waterdiepte voor wrijving
omega = 0.00014; %M2 frequentie / frequentie randvoorwaarde
N = 400; %aantal gridpunten, nu moet N>=4 max LS 200-400
L_dim = 1000; %Statische lengte bekken

%dimensieloze parameters:
L = 1;
H = 1;
g = g_dim*H_dim/omega^2/L_dim^2;
cd = cd_dim/H_dim/omega;
epsilon = Amp/H_dim;

dx= L/(N-1); %stapgrootte in x-richting
theta =1; %theta uit theta methode, 0=expliciet, 1=impliciet (volledig)
dt = .000125; %tijdstap grootte %.0005
%dt=1/7000*pi;
T=.3;
%T = 2*pi+2*dt; %eindtijd
%T=3*pi;
n = floor(T/dt); %aantal tijdstappen

grid = (0:N-1)*dx;
%h = 1*ones(size(grid)); %bodem hoogte tov referentiediepte
\%h = 0.5+3*grid/L;
\%h = 1 + [2*grid(1:10)/L, -4*grid(10)/L + 6*grid(11:N)/L ];
\%h = 29.25*(grid/L).^3 - 49.5*(grid/L).^2 + 22.25*grid/L;
\%h = 3*(5.52*(grid/L/.95).^4 + 24.97*(grid/L/.95).^3 - 49.5*(grid/L/.95).^2 + ...
22.52*grid/L/.95-.2); \%geschikt voor LS

h=3/2*H_dim/L*grid; \%lineaire bodem LS
h=h/H_dim;

Mstep=1;
Y = zeros(2*N-1,ceil(n/Mstep));
BZ = 1*sin((0:n-1)*dt); \%altijd geschaalde amplitude 1 en periode 1 (2*pi)
if verder
    BZ = 1*sin((0:n-1)*dt+T1);
end

\%Initial condition
\%Y(1:2:2*N-2,1)= -1./(H-h(1:N-1).*0.00014.*(grid(1:N-1) - L); \%Initial U
\%Y(1:2:2*N-2,1)= -1./(H-h(1:N-1)).*(grid(1:N-1) - L)*H_dim/1/L_dim; \%Initial U

ICLS=3;

if ICLS==1
    if N==50
        load ICn50LevelSet.mat
        Y(1:99,1) = ICn50LS;
    elseif N==100
        load ICn100LevelSet.mat
        Y(1:199,1) = ICn100LS;
    end
else ICLS==2
    Y(1:2:2*floor(2/3*N),1) = .6599; \%meanIcU
    Y(2:2:2*floor(2/3*N)-1,1) = .0165*log(2/3 - (1:floor(2/3*N-1))*dx)+.0065;
    \%ICuLSinterpoleren
else
    if N==50
        load ICn50.mat
        Y(1:98,1) = ICn50;
    elseif N==1001
        load ICn100.mat
        Y(1:198,1) = ICn100;
    end
end

if ICLS==2
    if N==200
        load ICu200.mat
        \%load ICu2002.mat
        \%Y(1:2:end,1) = 0.7;
Y(1:2:2*133-1,1) = ICu200;
load ICz200.mat
Y(2:2:2*133-1,1) = ICz200;
end

if N==400
    load ICfitN100LS.mat
    Y(1:2:2*266-1,1) = MeanICu;
    disp('hier')
    %Y(2:2:2*266-1,1) = polyval(pfz8,0:.25:66); %klopt niet helemaal
    Y(2:2:end,1) = 0;
end

indx = find(1-h<0,1,'first')-1;
s0 = grid(indx) - dx*(1-h(indx))/(-h(indx +1)+h(indx));
%--phi=zeros(N,n);
phi = zeros(N,1); %nodig voor goede vorm
%--phi(:,1)=grid-s0;
phi(:,1) = grid-s0;
if verder
    Y(:,1) = IC(2:2*N);
    %--phi(:,1) = ICphi;
    phi = ICphi;
end

d0 = zeros(2*N,1);
dm3 = repmat([g/2 0]',N,1);
dp1 = -dm3;
dm1 = repmat([0 H/2]',N,1);
dm1(2:2:2*N-2) = dm1(2:2:2*N-2) - h(1:N-1)'/2;
    %fout tot 26 april, h(2:N), ipv h(1:N-1)
dp3 = repmat([0 -H/2 ]',N,1);
dp3(4:2:2*N) = dp3(4:2:2*N) + h(2:N)'/2;
dm1(1) = g;
dp1(3) = -g;
dm1(2*N-2) = 2*dm1(2*N-2);
dp1(2*N)=h(N)-H;

A=spdiags([dm3 dm1 d0 dp1 dp3], [-3 -1 0 1 3], 2*N, 2*N);

ff=zeros(2*N-2,2);
ff(:,1) = A(2:2*N-1,1);
ff(:,2) = A(2:2*N-1,2*N);

A = A(2:2*N-1,2:2*N-1);
toc
S=Y(:,1);
phi0=zeros(1,n);
phi0n=zeros(1,n);
phi0Oud=zeros(1,n);%tijdelijk
phi0Nieuw=zeros(1,n);%tijdelijk
Locn = zeros(1,n); %midden van cel aan kust
aAlle = zeros(n,1);
bAlle = zeros(n,1);
TijdUrnd = zeros(n,3);
TijdzuRand2 = zeros(4,n);
TijdVolg =zeros(n,4);
ContrUvoort = zeros(n,3);
uoprand=0;
if theta ==0
  u = zeros(N,1); %voortgezette snelheid front
%  u = zeros(N,n); %voortgezette snelheid front
else
  u = zeros(N,n); %voortgezette snelheid front
end
Tnmax = 0;
Tnmin = 10000;
if theta == 0
  for i=1:n-1
    indx = find(phi(:,i)>0,1,'first')-1;
    phi0(i) = grid(indx) -dx*phi(indx,i)/(phi(indx +1,i)-phi(indx,i));
%    indx = find(phi>0,1,'first')-1;
%    phi0(i) = grid(indx) -dx*phi(indx)/(phi(indx +1)-phi(indx));
    cel=floor((phi0(i) +dx/2)/dx)+1; %cel waarin s ligt, oke
    a=(phi0(i) - (cel-3/2)*dx)/dx;
    aAlle(i) =a;
    phi0n(i) = cel-1+(a>=.5); %laatste cel met waarde, oke
    Locn(i) = (a<.5)*(grid(cel-1)+a*dx/2) + (a>=.5)*(grid(cel)-dx/2+a*dx/2);
    St=S;
    S=zeros(2*N-1,1);
    lst=length(St);
    S(1:lst,1)=St; %maakt S weer 2*N-1 lang
    %S(lst+1:1:lst+2) = S(lst-1:lst);
    %S(lst+1) = S(lst-1) + 1*(S(lst-1)-S(lst-3)); %snelheid nieuwe punt
    %S(lst+2) = S(lst) + 1*(S(lst)-S(lst-2)); %zeta nieuwe punt
    S(lst+1) = S(lst-1) + 1/(.5+a)*((-1+(3/2-a)*h(phi0n(i)) +... 
        (a-1/2)*h(phi0n(i)+1))/epsilon - S(lst-1));
    S(lst+2) = S(lst) +(uoprand - S(lst))/(.5+a);
    At=A(1:2*phi0n(i)-1,1:2*phi0n(i)-1);
    bron = zeros(2*phi0n(i)-1,1);
  end
  
end
bron([1; 3]) = [g; g/2]*BZ(i);
bron(2) = 1/2*epsilon*S(1)*BZ(i); %fout tot 28 april, 1/2 miste

% Anlt = spdiags(d0,0,2*N-2,2*N-2);
d0 = zeros(2*N-1,1);
d0(1:2:2*N-1) = -dx*cq/(1-h'+ epsilon*[BZ(i); S(2:2*N-2)]+hmin);
dm2t = zeros(2*N-1,1);
dm2t(2:2:2*N-3) = 1/2*epsilon*S(3:2:2*N-3); %fout tot 28 april, 1/2 miste
dp2t = zeros(2*N-1,1);
dp2t(4:2:2*N-1) = -1/2*epsilon*S(5:2:2*N-1);
Anlt = spdiags([dm2t,d0,dp2t],[-2 0 2],2*N-1,2*N-1);
Anlt = Anlt(1:2*phi0n(i)-1,1:2*phi0n(i)-1); %gooit laatste deel weg

a=a+1*(a<.5);
bAlle(i) = a;
At(2*phi0n(i) -2,2*phi0n(i) -3) = (1-h(phi0n(i)-1))/2/a;
At(2*phi0n(i) -2,2*phi0n(i) -1) = (1-h(phi0n(i)))/2/a;
Anlt(2*phi0n(i) -2,2*phi0n(i) -4) = epsilon*S(2*phi0n(i)-3)/2/a;
Anlt(2*phi0n(i) -2,2*phi0n(i) -2) = epsilon*S(2*phi0n(i)-1)/2/a;
At(2*phi0n(i) -1,2*phi0n(i)-4) = g/2/a;
At(2*phi0n(i) -1,2*phi0n(i)-2) = g/2/a;
bron(2*phi0n(i)-1) = -g/epsilon/a*(-1+(3/2-a)*h(phi0n(i)) +...
(-1/2*a)*h(phi0n(i)+1));
S = S(1:2*phi0n(i)-1);
S = S + dt/dx*(At+Anlt)*S + bron);

if i-Mstep*floor(i/Mstep)==(1-Mstep==1)
 Y(1:2*phi0n(i)-1,floor(i/Mstep)+1)=S;
end

TijdUrnd(i+1,1) = S(2*phi0n(i)-1); %snelheid laatste gridpunt
TijdUrnd(i+1,2) = (a+1/2)*S(2*phi0n(i)-1) - (a-1/2)*S(2*phi0n(i)-3);
%snelheid op de rand
TijdUrnd(i+1,3) = 2*S(2*phi0n(i)-1) - S(2*phi0n(i)-3);
%snelheid volgende punt

%uoprand = S(2*phi0n(i)-1); %eertse keus, gewoon vorige gridpunt
uoprand = TijdUrnd(i+1,2); %moet i+1 zijn

% u voortzetten
identical to code for other cases of theta...

% phi(i+1) bepalen
% dm31= [-max(0,u(2:N,i))];0];
% d30 = abs(u(:,i));
% dp31= [0;min(0,u(1:N-1,i))];
dm31= [-max(0,u(2:N));0];
d30 = abs(u(:));
dp31 = [0; min(0, u(1:N-1))];

B = spdiags([dm31, d30, dp31], [-1, 0, 1], N, N);

phi(:, i+1) = phi(:, i) - dt/dx * B * phi(:, i);

if u(N, i) < 0
    phi(N, i+1) = phi(N, i);
    phi(N) = phioud(N);
elseif u(1, i) > 0
    phi(1, i+1) = phi(1, i);
    phi(1) = phioud(1);
end

end

elseif theta == 1
    disp('theta is 1')
    for i = 1:n-1
        indx = find(phi(:, i) > 0, 1, 'first') - 1;
        phi0(i) = grid(indx) - dx * phi(indx, i) / (phi(indx + 1, i) - phi(indx, i));
        cel = floor((phi0(i) + dx/2) / dx) + 1; % cel waarin s ligt
        a = (phi0(i) - (cel - 3/2) * dx) / dx;
        aAlle(i) = a;
        phi0n(i) = cel - 1 + (a >= 0.5);
        Locn(i) = (a < 0.5) * (grid(cel - 1) + a * dx/2) + (a >= 0.5) * (grid(cel) - dx/2 + a * dx/2);
    end

St = S;
S = zeros(2*N-1, 1);
lst = length(St);
S(l1, lst) = St; % maakt S weer 2*N-1 lang
TijdzuRand2(1, i) = S(l1-1) + 1*(S(l1-1) - S(l1-3));
TijdzuRand2(2, i) = S(l1-1) + 1/(0.5*a)*((-1 + (3/2 - a)*h(php0n(i)) + ...
(a-1/2)*h(php0n(i)+1))/epsilon - S(l1-1));
TijdzuRand2(3, i) = S(l1) + 1*(S(l1) - S(l1-2));
TijdzuRand2(4, i) = S(l1) + (uprand - S(l1))/0.5*a);

S(lsta+1) = S(lsta-1) + 1*(S(lsta-1) - S(lsta-3));
S(lsta+2) = S(lsta) + 1*(S(lsta) - S(lsta-2));
S(lsta+1) = S(lsta-1) + 1/(0.5*a)*((-1 + (3/2 - a)*h(php0n(i)) + ...
(a-1/2)*h(php0n(i)+1))/epsilon - S(lsta-1));
\[ S(lst+2) = S(lst) + (uoprand - S(lst)) / (.5*a); \] % snelheid nieuwe punt

\[ A_t = A(1:2*\phi_0n(i)-1, 1:2*\phi_0n(i)-1); \]
\[ \text{bron} = \text{zeros}(2*\phi_0n(i)-1,1); \]
\[ \text{bron}(1:3) = [g; g/2]*BZ(i+1); \]
\[ \text{bron}(2) = 1/2*\epsilon*S(1)*BZ(i+1); \] % fout tot 28 april, 1/2 miste

\[ d_0 = \text{zeros}(2*N-1,1); \]
\[ d_0(1:2:2*N-1) = -dx*cd. / (1-h'+ \epsilon*[BZ(i); S(2:2:2*N-2)] + h\text{min}); \]
\[ d_{m2} = \text{zeros}(2*N-1,1); \]
\[ d_{m2}(2:2:2*N-3) = 1/2*\epsilon*S(3:2:2*N-3); \] % fout 28 april, 1/2 miste

\[ d_{p2} = \text{zeros}(2*N-1,1); \]
\[ d_{p2}(4:2:2*N-1) = -1/2*\epsilon*S(5:2:2*N-1); \]
\[ A_{nlt} = \text{spdiags}([d_{m2}d_0d_{p2}],[2 0 2],[2*N-1,2*N-1]); \]
\[ A_{nlt} = A_{nlt}(1:2*\phi_0n(i)-1, 1:2*\phi_0n(i)-1); \] % gooit laatste deel weg

\[ a = a + 1*(a<.5); \]
\[ b_{Alle}(i) = a; \]
\[ A_t(2*\phi_0n(i)-2,2*\phi_0n(i)-3) = (1-h(\phi_0n(i)-1))/2/a; \]
\[ A_t(2*\phi_0n(i)-2,2*\phi_0n(i)-1) = (1-h(\phi_0n(i)))/2/a; \]
\[ A_{nlt}(2*\phi_0n(i)-2,2*\phi_0n(i)-4) = \epsilon*S(2*\phi_0n(i)-3)/2/a; \]
\[ A_{nlt}(2*\phi_0n(i)-2,2*\phi_0n(i)-2) = \epsilon*S(2*\phi_0n(i)-1)/2/a; \]
\[ A_t(2*\phi_0n(i)-1,2*\phi_0n(i)-4) = g/2/a; \]
\[ A_t(2*\phi_0n(i)-1,2*\phi_0n(i)-2) = g/2/a; \]
\[ \text{bron}(2*\phi_0n(i)-1) = g/\epsilon/a*(h(\phi_0n(i))) + \ldots \]
\[ (1/2+a)*h(\phi_0n(i)+1)); \]

\[ S = S(1:2*\phi_0n(i)-1); \]
\[ S = (\text{eye}(2*\phi_0n(i)-1)-dt/dx*(A_t+A_{nlt})) \backslash (S + dt/dx*\text{bron}); \]
% minteken in inverse miste tot 11 mei
if i-Mstep*floor(i/Mstep)==(1-Mstep==1)

\[ Y(1:2*\phi_0n(i)-1,floor(i/Mstep)+1)=S; \]
end

\[ TijdUrnd(i+1,1) = S(2*\phi_0n(i)-1); \] % snelheid laatste gridpunt
\[ TijdUrnd(i+1,2) = (a+1/2)*S(2*\phi_0n(i)-1) - (a-1/2)*S(2*\phi_0n(i)-3); \]
\[ TijdUrnd(i+1,3) = 2*S(2*\phi_0n(i)-1) - S(2*\phi_0n(i)-3); \]
% uoprand = S(2*\phi_0n(i)-1); % eerste keus, gewoon vorige gridpunt
uoprand = TijdUrnd(i+1,2); % moet i+1 zijn
% u voortzetten
identical to code for other cases of theta...
% or when no extension is use just the only line:
\[ u(:,i) = uoprand; \]
% \phi(i+1) bepalen
\[ dm31 = [-\max(0,u(2:N,i));0]; \]
\[ d30 = \text{abs}(u(:,i)); \]
dp31 = [0; min(0, u(1:N-1, i))];

B = spdiags([dm31, d30, dp31], [-1, 0, 1], N, N);
phi(:, i+1) = phi(:, i) - dt/dx*B*phi(:, i);

if u(N, i) < 0  % randvoorwaarde bij inflow
    phi(N, i+1) = phi(N, i);
elseif u(1, i) > 0
    phi(1, i+1) = phi(1, i);
end

% phi reinitialiseren
identical to code for other cases of theta...
end
else
    disp('theta is vrij')
for i = 1:n-1
    indx = find(phi(:, i) > 0, 1, 'first') - 1;
    phi0(i) = grid(indx) - dx*phi(indx, i)/(phi(indx + 1, i) - phi(indx, i));
    cel = floor((phi0(i) + dx/2)/dx) + 1; % cel waarin s ligt
    a = (phi0(i) - (cel-3/2)*dx)/dx;
    aAlle(i) = a;
    phi0n(i) = cel - 1 + (a >= .5);
    Locn(i) = (a < .5)*(grid(cel - 1) + a*dx/2) + (a >= .5)*(grid(cel) - dx/2 + a*dx/2);
end
St = S;
S = zeros(2*N - 1, 1);
lst = length(St);
S(1:lst, 1) = St; % maakt S weer 2*N - 1 lang
TijdzuRand2(1, i) = S(lst - 1) + 1*(S(lst - 1) - S(lst - 3));
TijdzuRand2(2, i) = S(lst - 1) + 1/(.5*a)*((-1 + (3/2 - a)*h(phi0n(i)) +...)
                        (a - 1/2)*h(phi0n(i) + 1))/epsilon - S(lst - 1));
TijdzuRand2(3, i) = S(lst - 1) + 1*(S(lst) - S(lst - 2));
TijdzuRand2(4, i) = S(lst) + (uoprand - S(lst))/(.5*a);
S(lst + 1) = S(lst - 1) + 1/(.5*a)*((-1 + (3/2 - a)*h(phi0n(i)) +...)
                              (a - 1/2)*h(phi0n(i) + 1))/epsilon - S(lst - 1)); % zeta nieuwe punt
S(lst + 2) = S(lst) + (uoprand - S(lst))/(.5*a); % snelheid nieuwe punt

At = A(1:2*phi0n(i) - 1, 1:2*phi0n(i) - 1);
bronOd = zeros(2*phi0n(i) - 1, 1);
bronNw = zeros(2*phi0n(i) - 1, 1);

bronOd([1; 3]) = [g; g/2]*BZ(i);
bronOd(2) = 1/2*epsilon*S(1)*BZ(i);  % fout tot 28 april, 1/2 miste
bronNw([1; 3]) = [g; g/2]*BZ(i + 1);
bronNw(2) = 1/2*epsilon*S(1)*BZ(i + 1);  % fout tot 28 april, 1/2 miste

d0 = zeros(2*N - 1, 1);
d0(1:2:2*N - 1) = -dx*cd./(1 - h' + epsilon*[BZ(i); S(2:2:2*N - 2)] + hmin);
dm2t = zeros(2*N-1,1);
dm2t(2:2:2*N-3) = 1/2*epsilon*S(3:2:2*N-3); %fout 28 april, 1/2 miste
dp2t = zeros(2*N-1,1);
dp2t(4:2:2*N-1) = -1/2*epsilon*S(5:2:2*N-1);
Anlt = spdiags([dm2t,d0,dp2t],[-2 0 2],2*N-1,2*N-1);
Anlt = Anlt(1:2*phi0n(i)-1,1:2*phi0n(i)-1); %gooit laatste deel weg

a=a+1*(a<.5);
bAlle(i) = a;
At(2*phi0n(i)-2,2*phi0n(i)-3) = (1-h(phi0n(i)-1))/2/a;
At(2*phi0n(i)-2,2*phi0n(i)-1) = (1-h(phi0n(i)))/2/a;
Anlt(2*phi0n(i)-2,2*phi0n(i)-4) = epsilon*S(2*phi0n(i)-3)/2/a;
Anlt(2*phi0n(i)-2,2*phi0n(i)-2) = epsilon*S(2*phi0n(i)-1)/2/a;
At(2*phi0n(i)-1,2*phi0n(i)-4) = g/2/a;
At(2*phi0n(i)-1,2*phi0n(i)-2) = g/2/a;
bronOd(2*phi0n(i)-1) = -g/epsilon/a*(-1+(3/2-a)*h(phi0n(i)) +...(-1/2+a)*h(phi0n(i)+1));
bronNw(2*phi0n(i)-1) = -g/epsilon/a*(-1+(3/2-a)*h(phi0n(i)) +...(-1/2+a)*h(phi0n(i)+1));

S = S(1:2*phi0n(i)-1);
	heta2=theta; %bron oud of middelen
S = (eye(2*phi0n(i)-1)-theta*dt/dx*(At+Anlt))...
(S + (1-theta)*dt/dx*(At+Anlt)*S +...
(1-theta2)*dt/dx*bronOd + theta2*dt/dx*bronNw);
%minteken in inverse miste tot 11 mei

if i-Mstep*floor(i/Mstep)==(1-Mstep==1)
Y(1:2*phi0n(i)-1,floor(i/Mstep)+1)=S;
end

TijdUrnd(i+1,1) = S(2*phi0n(i)-1); %snelheid laatste gridpunt
TijdUrnd(i+1,2) = (a+1/2)*S(2*phi0n(i)-1) - (a-1/2)*S(2*phi0n(i)-3); %snelheid op de rand
TijdUrnd(i+1,3) = 2*S(2*phi0n(i)-1) - S(2*phi0n(i)-3); %snelheid volgende punt
uoprand = TijdUrnd(i+1,2); %moet i+1 zijn

%u voortzetten
s = phi0(i);
aa=(s-(indx-1)*dx)/dx;

dm21 = [-max(0,sign(phi(2:N,i)))]|0];
d20 = ones(N,1);
wp21 = [0;min(0,sign(phi(1:N-1,i)))]|0];

dm21(indx-1) = 0; %Gridpunt i
dm21(indx) = 0; %Gridpunt i+1
dp21 indx+1 = 0; %Gridpunt i
dp21 indx+2 = 0; %Gridpunt i+1
d20 indx = 1/(1/2+aa);
d20 indx+1 = 1/(3/2-aa);

f2 = zeros(N,1);
f2 indx = uoperand/(1/2+aa);
f2 indx+1 = uoperand/(3/2-aa);

C2 = spdiags([dm21 d20 dp21], [-1 0 1], N, N);

dtau = 0.85*dx;
Tn = floor(max(s,1-s)/dtau + 5)+7+3; %+10
Tnmin = min(Tn,Tnmin);
Tnmax = max(Tn,Tnmax);
unum = zeros(N, Tn); %kan nog tot een vector
for k = 1: Tn-1
    unum(:,k+1) = unum(:,k) - dtau/dx*C2*unum(:,k) + dtau/dx*f2;
end
u(:, i) = unum(:, Tn);
ContrUvoort(i,1) = unum(floor(N/2), Tn) - unum(1, Tn);
ContrUvoort(i,2) = unum(floor(N/2), Tn) - unum(N, Tn);
ContrUvoort(i,3) = unum(N, Tn) - unum(1, Tn);

%phi(i+1) bepalen
dm31 = [-max(0, u(2:N,i)); 0];
d30 = abs(u(:,i));
dp31 = [0; min(0, u(1:N-1,i))];

B = spdiags([dm31 d30 dp31], [-1,0,1], N, N);

phi(:, i+1) = phi(:, i) - dt/dx*B*phi(:, i);

if u(N,i) < 0 %randvoorwaarde bij inflow
    phi(N,i+1) = phi(N,i);
elseif u(1,i) > 0
    phi(1,i+1) = phi(1,i);
end

%phi reinitialiseren
index0 = find(phi(:,i+1)>0, 1, 'first')-1;
phi00ud(i+1) = grid(index0) - dx*phi(index0,i)/(phi(index0+1,i)...-
    phi(index0,i));
P = phi(:, i+1);
Sign = P./sqrt(P.^2 + (2*dx)^2);
m = 4; %2 stappen genoeg
dt2 = dx/10; %dx/2 kan ook.
for j = 1:m

68
G0 = GodunovWeno(P, N, dx);
s1 = P + dt*Sign.*G0;
G1 = GodunovWeno(s1, N, dx);
s2 = s1 + dt*Sign.*G1;
s3 = 3/4*P + 1/4*s2;
G2 = GodunovWeno(s3, N, dx);
s4 = s3 + dt*Sign.*G2;
P = 1/3*P + 2/3*s4; % 3rd order TVD Runge Kutta (book Osher en Fedkiw)
end
phi(:, i+1) = P;
index0 = find(phi(:, i+1) > 0, 1, 'first') - 1;
phi0Nieuw(i+1) = grid(index0) - dx*phi(index0, i)/(phi(index0+1, i) - phi(index0, i));
end
end
i = n;
indx = find(phi(:, i) > 0, 1, 'first') - 1;
phi0(i) = grid(indx) - dx*phi(indx, i)/(phi(indx + 1, i) - phi(indx, i));

%--indx = find(phi > 0, 1, 'first') - 1;
%--phi0(i) = grid(indx) - dx*phi(indx)/(phi(indx + 1) - phi(indx));

a = (phi0(i) - (cel - 3/2)*dx)/dx;
aAlle(i) = a;
phi0n(i) = cel - 1 + (a >= .5);
Locn(i) = (a < .5)*(grid(cel - 1) + a*dx/2) + (a >= .5)*(grid(cel) - dx/2 + a*dx/2);
toc
if verder
    MeerdereSimulatiesGrafiekLevelSet
else
    GrafiekFilmNieuwKleineYLevelset1
    % GrafiekInstroomNKleineY
end

Godunov WENO scheme

function GodunovWeno = newfunc(phi, N, dx)
% computes the godunov flux using a fifth order weno scheme

Dm = [0; phi(2:N)-phi(1:N-1)]/dx;

t1 = [0; 0; Dm(1:N-2)];
t2 = [0; Dm(1:N-1)];
t3 = Dm;
t4 = [Dm(2:N); 0];
t5 = [Dm(3:N); 0; 0];

fx1 = t1/3 - 7/6*t2 + 11/6*t3;
fx2 = -t2/6 + 5/6*t3 + t4/3;
fx3 = v3/3 +5/6*v4 -v5/6;
S1=13/12*(v1-2*v2+v3).^2+(v1-4*v2+3*v3).^2/4;
S2=13/12*(v2-2*v3+v4).^2+(v2-v4).^2/4;
S3=13/12*(v3-2*v4+v5).^2+(3*v3-4*v4+v5).^2/4;
ep=10e-6;
a1=.1./((S1+ep).^2);
a2=.6./((S2+ep).^2);
a3=.3./((S3+ep).^2);
as=a1+a2+a3;
o1=a1./as;
o2=a2./as;
o3=a3./as;
a = o1.*fx1 + o2.*fx2 + o3.*fx3;

w1=[phi(4:N)-phi(3:N-1); 0; 0; 0]/dx;
w2=v5;
w3=v4;
w4=v3;
w5=v2;
v1=w1;
v2=w2;
v3=w3;
v4=w4;
v5=w5;
fx1 = v1/3 -7/6*v2 +11/6*v3;
fx2 = -v2/6 +5/6*v3 +v4/3;
fx3 = v3/3 +5/6*v4 -v5/6;
S1=13/12*(v1-2*v2+v3).^2+(v1-4*v2+3*v3).^2/4;
S2=13/12*(v2-2*v3+v4).^2+(v2-v4).^2/4;
S3=13/12*(v3-2*v4+v5).^2+(3*v3-4*v4+v5).^2/4;
ep=10e-6;
a1=.1./((S1+ep).^2);
a2=.6./((S2+ep).^2);
a3=.3./((S3+ep).^2);
as=a1+a2+a3;
o1=a1./as;
o2=a2./as;
o3=a3./as;
b = o1.*fx1 + o2.*fx2 + o3.*fx3;

GodunovGweno = 1-sqrt(max(max(a,0).^2,min(b,0).^2)).*(phi>0) -...
sqrt(max(min(a,0).^2,max(b,0).^2)).*(phi<0) - 1*(phi==0);