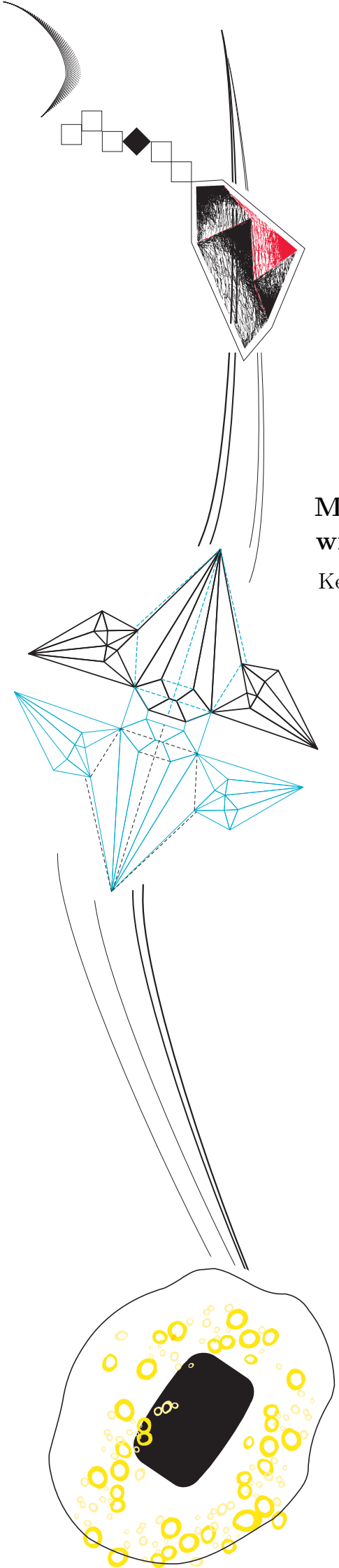


Modelling of water wave propagation with FUNWAVE-TVD

Kevin Overmars (s1358960)

UNIVERSITY OF TWENTE.

Preface

As part of the master program Mechanical Engineering every student has to do an internship outside of the University of Twente. This can be done either within the Netherlands or abroad. I preferred to go abroad to experience working and living in another country. I was attracted by Scandinavia due to the culture, nature and high proficiency of English. To explore the possibilities I spoke to professor Venner and professor Hoeijmakers, who introduced me to mister Johansen at SINTEF. I really want to thank them for the great opportunity they gave me.

My work was within the Flow Technology group of the SINTEF Industry institute. It is one of the six research institutes of SINTEF. The group consists of 15-20 people and is located in Trondheim, where SINTEF is also headquartered. They are working on all kinds of flow problems and are specialised in numerically computing multiphase flows. I have felt very welcome during my time in Trondheim and would like to thank the Flow Technology group for the great time I have had.

My work was within the SprayIce project in which the formation of ice depositions on ships and sea structures is modelled. The project covers the full range from sea waves to the actual freezing on the surface, in which the Flow Technology group is mainly working on the spray formation. My part of the work was focused on the modelling of water waves which could be used as input to other simulations. I would like to thank Stein Tore Johansen and Sverre Gullikstad Johnson for the support and help they gave me during my internship.

Summary

Marine icing is a severe hazard for ships and structures in arctic seas. In order to predict ice depositions SINTEF is working on the SprayIce project in which the formation of sprays is modelled. In this work the possibilities of FUNWAVE-TVD for modelling large-scale wave fields, which could serve as input to the spray formation simulations, are explored.

FUNWAVE-TVD numerically solves Boussinesq-type equations for the propagation of water waves. The vertical dimension is eliminated in Boussinesq-type equations by expanding the horizontal velocity around a reference level and integrating over the depth. In FUNWAVE-TVD a moving reference is incorporated. For the event of wave breaking and the dissipation of energy by wave breaking two schemes are included. The equations are rewritten to conservative form to make them suitable for the numerical schemes. The spatial scheme is a hybrid finite volume-finite difference scheme. The flux and first order derivatives are treated with a high order MUSCL-TVD scheme of which fourth and second order accurate schemes are available. Higher derivatives terms are treated with a central difference scheme. The time scheme is a third order Strong-Stability Preserving Runge-Kutta scheme with adaptive time stepping.

The model is validated by simulating solitary wave collisions which are compared to experimental data obtained from literature. A head-on and an overtaking collision are simulated in which waves move in opposite direction and the same direction respectively. In both cases the results are very similar to the experimental data, but minor errors do exist. In the head-on case the simulated collision occurs faster than the experimental results and in the overtaking case the wave propagation is faster than expected.

The large scale simulations show that FUNWAVE is capable of modelling these large wavefields. The nesting possibilities were also explored and demonstrate the capabilities of the code to transfer the results from a coarse grid to a finer grid. In these simulations the code is run on a computing cluster where up to 25 processors were used.

Some instability problems were observed in the large scale simulations on the transition from water to land. This is attributed to the high gradients which can occur in this region and a numerical feature, the minimum water depth.

Contents

Chapter 1

Introduction

The SprayIce project is a SINTEF research project which focuses on marine icing. This is a process in which waves hit vessels or platforms and create splashing droplets. When the (wind) conditions are right they are getting supercooled and freeze to the surface. The ice depositions on the surface can result in unsafe conditions. This ranges from slippery decks up to instabilities arising from the extra mass added by the deposition.

In order to make predictions of the ice depositions under various weather conditions the full process is modelled in the project. To do so the project is split up in various parts e.g. wave propagation, wave impacts, droplet-wind interactions. These parts are coupled in one direction. This report focuses on the first part: wave propagation. As the length scale of the droplets formation is much smaller than the scale for the propagating waves it would be expensive to solve them on the same grid. The total area in which it is desired to model the wave propagation can become quite large, in the order of kilometres.

As it is computationally expensive to simulate the waves it is desired to use a simplified model. To this end the model is reduced from three to two dimensions, where the depth becomes a parameter of the horizontal cells. This derivation and other model equations are discussed in chapter 2. The obtained equations form the basis of the open source program FUNWAVE-TVD, which is used in this research to simulate wave propagation. The numerical solvers used in this program are discussed in chapter 3.

Validation of the model is discussed in chapter 4. This will be followed by the end goal of this study: performing large scale simulations, which includes model nesting. In the nested model a simulation on a courser grid is coupled to a simulation on a finer grid.

Chapter 2

Model equations

FUNWAVE numerically solves Boussinesq-type model equations. Boussinesq-type models are widely used in the propagation of water waves in coastal areas. In Boussinesq-type models the vertical coordinate is eliminated from the model equation. This is done by using a Taylor expansion around a reference height and integrating the equations over the depth. Classical Boussinesq models are valid for weakly nonlinear and weakly dispersive waves, which is characterised by $\delta << 1$ and $\mu << 1$ respectively. These parameters represent ratios of wave height over still water height and wave height over wave length. The validity of the models has been extended and other effects like wave-breaking, bottom friction etc. have been included. This report will discuss the basics about the model used by FUNWAVE and the reader is referred to other sources for the full derivations.

2.1 Derivation

The governing equations of FUNWAVE are based on the model equations derived by Chen [1] and are extended to incorporate a moving reference level proposed by Kennedy et al. [2]. The basics of the derivation are described in this section and the reader is referred to [1] for a full derivation. The model is fully non-linear, which means no terms are truncated based on the non-linearity parameter δ. In the derivations terms beyond $O(\mu^2)$ are truncated. The equations here will be in a dimensional form.

2.1.1 Chen's equations

A schematic overview of the flow of waves over a variable sea bottom can be seen in figure 2.1. It is assumed that the flow over the sea bottom is incompressible and inviscid. The continuity equation and incompressible Euler equations thus form the basis of the model.

$$\nabla_3 \cdot \mathbf{u} = 0 \tag{2.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{1}{2}\nabla_3(\mathbf{u}\cdot\mathbf{u}) + (\nabla_3 \times \mathbf{u}) \times \mathbf{u} + \frac{1}{\rho}\nabla_3 p + g\nabla_3 z = 0 \tag{2.2}$$

where ∇_3 is the three dimensional gradient operator, $\mathbf{u}$ is the three dimensional velocity vector (u, v, w), t is the time, p is the pressure, ρ is the fluid density and g is the gravitational acceleration.
Next to this three boundary equations are formulated, a kinematic and a dynamic one at the free surface, $z = \eta$, and a kinematic one at the bottom, $z = -h$,

$$w = \frac{\partial \eta}{\partial t} + \hat{\mathbf{u}} \cdot \nabla\eta, \quad z = \eta \tag{2.3}$$

$$p = 0, \qquad z = \eta \tag{2.4}$$

$$w + \hat{\mathbf{u}} \cdot \nabla h = 0, \qquad z = -h \tag{2.5}$$

where ∇ is the two dimensional gradient operator and $\hat{\mathbf{u}}$ is the two dimensional velocity vector (u, v). By integrating the continuity equation over the depth and using the two kinematic boundary conditions a depth-integrated volume conservation equation can be derived:

$$\frac{\partial \eta}{\partial t} + \nabla \cdot \mathbf{M} = 0, \quad \mathbf{M} = \int_{-h}^{\eta} \hat{\mathbf{u}} dz. \tag{2.6}$$

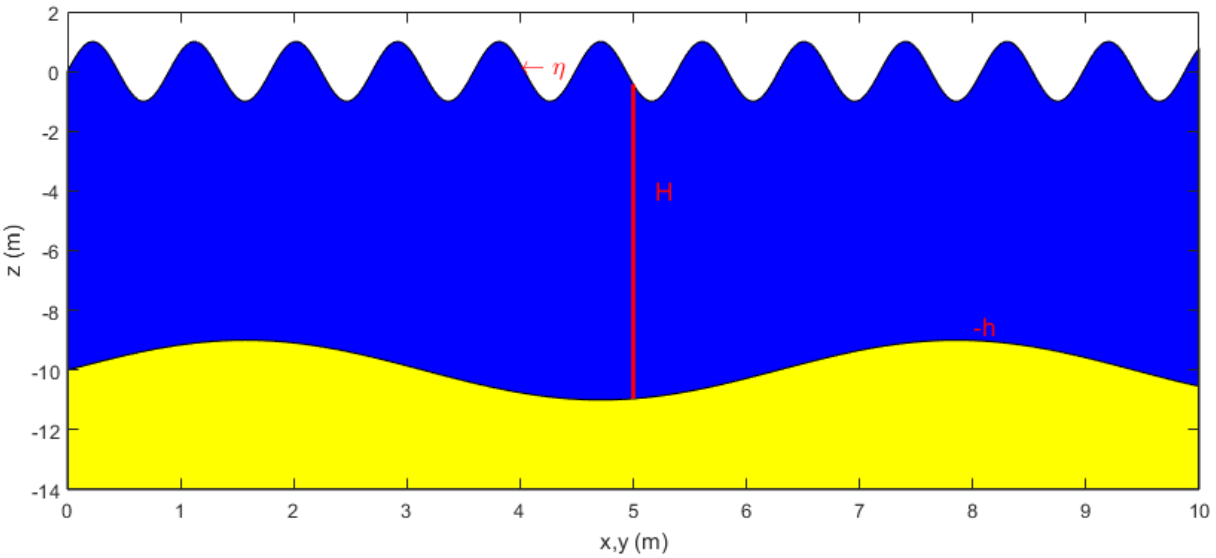


Figure 2.1: Schematic overview of water waves over a varying seabed height

At the reference level z_α, defined in sector 2.2, the horizontal velocity is approximated by an expansion, which is truncated after the $O(\mu^2)$ terms. This gives

$$\hat{\mathbf{u}} = \mathbf{u}_\alpha + \mathbf{u}_2(z) \tag{2.7}$$

where $\mathbf{u}_\alpha$ is the horizontal velocity field at the reference level and $\mathbf{u}_2$ contains the depth dependent terms remaining after the truncation. These are

$$\mathbf{u}_2(z) = (z_\alpha - z)\nabla A + \frac{1}{2}(z_\alpha^2 - z^2)\nabla B \tag{2.8}$$

with

$$\begin{aligned} A &= \nabla \cdot (h\mathbf{u}_\alpha), \\ B &= \nabla \cdot (\mathbf{u}_\alpha). \end{aligned} \tag{2.9}$$

Evaluating the integral in the volume conservation equation results in

$$\mathbf{M} = H(\mathbf{u}_\alpha + \bar{\mathbf{u}}_2) \tag{2.10}$$

where $H = h + \eta$ and $\bar{\mathbf{u}}_2$ is the depth averaged $\mathbf{u}_2$, which is given by

$$\bar{\mathbf{u}}_2 = \frac{1}{H}\int_{-h}^{\eta} \mathbf{u}_2(z)dz = \left(z_\alpha + \frac{1}{2}(h-\eta)\right)\nabla A + \left(\frac{z_\alpha^2}{2} - (h^2 - h\eta + \eta^2)\right)\nabla B. \tag{2.11}$$

Similar procedures can be followed to obtain a depth averaged horizontal momentum equation

$$\frac{\partial \mathbf{u}_\alpha}{\partial t} + (\mathbf{u}_\alpha \cdot \nabla)\mathbf{u}_\alpha + g\nabla\eta + \mathbf{V}_1 + \mathbf{V}_2 + \mathbf{V}_3 + \mathbf{R} = 0 \tag{2.12}$$

where g is the gravitational acceleration and $\mathbf{R}$ includes dissipative and diffusive terms like bottom friction. In $\mathbf{V}_1$ and $\mathbf{V}_2$ dispersive Boussinesq terms are captured, given by

$$\mathbf{V}_1 = \left\{ z_\alpha \nabla A + \frac{z_\alpha^2}{2}\nabla B \right\}_{,t} - \nabla\left[\eta A_{,t} + \frac{\eta^2}{2}B_{,t}\right], \tag{2.13}$$

$$\mathbf{V}_2 = \nabla\left\{ (z_\alpha - \eta)(\mathbf{u}_\alpha \cdot \nabla)A + \frac{1}{2}(z_\alpha^2 - \eta^2)(\mathbf{u}_\alpha \cdot \nabla)B + \frac{1}{2}[A + \eta B]^2 \right\}. \tag{2.14}$$

$\mathbf{V}_3$ includes the $O(\mu^2)$ contribution of the vertical vorticity and is given by

$$\mathbf{V}_3 = \omega_0 \mathbf{i}_z \times \bar{\mathbf{u}}_2 + \omega_2 \mathbf{i}_z \times \mathbf{u}_\alpha \tag{2.15}$$

where

$$\omega_0 = (\nabla \times \mathbf{u}_\alpha) \cdot \mathbf{i}_z = \frac{\partial v_\alpha}{\partial x} - \frac{\partial u_\alpha}{\partial y}, \tag{2.16}$$

$$\omega_2 = (\nabla \times \mathbf{u}_2) \cdot \mathbf{i}_z = \frac{\partial z_\alpha}{\partial x}\left(\frac{\partial A}{\partial y} + z_\alpha \frac{\partial B}{\partial y}\right) - \frac{\partial z_\alpha}{\partial y}\left(\frac{\partial A}{\partial x} + z_\alpha \frac{\partial B}{\partial x}\right). \tag{2.17}$$

2.1.2 Conservative form

The numerical scheme requires the equations to be in a conservative form and equation 2.12 is thus rewritten to conservative form, where $\mathbf{M}$ is used as a conserved variable. This results in

$$\frac{\partial \mathbf{M}}{\partial t} + \nabla \cdot \left(\frac{\mathbf{M}\mathbf{M}}{H}\right) + gH\nabla\eta = H(\bar{\mathbf{u}}_{2,t} + \mathbf{u}_\alpha \cdot \nabla \bar{\mathbf{u}}_2 + \mathbf{u}_2 \cdot \nabla \bar{\mathbf{u}}_\alpha - \mathbf{V}_1 - \mathbf{V}_2 - \mathbf{V}_3 - \mathbf{R}). \tag{2.18}$$

Next to this the surface gradient term is split as

$$gH\nabla\eta = \nabla\left[\frac{1}{2}g(\eta^2 + 2h\eta)\right] - g\eta\nabla h \tag{2.19}$$

and the time derivative terms in $\mathbf{V}_1$ are separated into

$$\mathbf{V}_1 = \mathbf{V}'_{1,t} + \mathbf{V}''_1 \tag{2.20}$$

$$= \left\{\frac{z_\alpha^2}{2}\nabla B + z_\alpha \nabla A - \nabla\left[\frac{\eta^2}{2}B + \eta A\right]\right\}_{,t} + \nabla[\eta_{,t}(A + \eta B)]. \tag{2.21}$$

This is substituted into 2.18 to obtain

$$\begin{aligned} &\frac{\partial \mathbf{M}}{\partial t} + \nabla \cdot \left(\frac{\mathbf{M}\mathbf{M}}{H}\right) + \nabla\left[\frac{1}{2}g(\eta^2 + 2h\eta)\right] = \\ &\quad H(\bar{\mathbf{u}}_{2,t} + \mathbf{u}_\alpha \cdot \nabla \bar{\mathbf{u}}_2 + \mathbf{u}_2 \cdot \nabla \bar{\mathbf{u}}_\alpha - \mathbf{V}'_{1,t} - \mathbf{V}''_1 - \mathbf{V}_2 - \mathbf{V}_3 - \mathbf{R}) + g\eta\nabla h \end{aligned} \tag{2.22}$$

The time derivative terms $\bar{\mathbf{u}}_{2,t}$ and $\mathbf{V}'_{1,t}$ on the right hand side result in difficulties when applying the adaptive time stepping scheme. To resolve these issues equation 2.22 is rewritten into

$$\begin{aligned} &\frac{\partial \mathbf{V}}{\partial t} + \nabla \cdot \left[\frac{\mathbf{M}\mathbf{M}}{H}\right] + \nabla\left[\frac{1}{2}g(\eta^2 + h\eta)\right] = \\ &\quad \frac{\partial \eta}{\partial t}(\mathbf{V}'_1 - \bar{\mathbf{u}}_2) + H(\mathbf{u}_\alpha \cdot \nabla \bar{\mathbf{u}}_2 + \bar{\mathbf{u}}_2 \cdot \nabla \mathbf{u}_\alpha - \mathbf{V}''_1 - \mathbf{V}_2 - \mathbf{V}_3 - \mathbf{R}) + g\eta\nabla h \end{aligned} \tag{2.23}$$

where $\mathbf{V}$ is given by

$$\mathbf{V} = H(\mathbf{u}_\alpha + \mathbf{V}'_1) \tag{2.24}$$

The time derivative of the surface elevation on the right hand side $\frac{\partial \eta}{\partial t}$ can be calculated explicitly by making use of the depth integrated volume equation, which is stated here again for completeness.

$$\frac{\partial \eta}{\partial t} + \nabla \cdot \mathbf{M} = 0 \tag{2.25}$$

Equations 2.25 and 2.23 are the equations which will be solved numerically. The numerical schemes that are used will be discussed in the next chapter.

2.2 Moving reference level

The reference level z_α is usually chosen such that it optimises the dispersion relation of the linearised model relative to the full linear dispersion. z_α is determined by the relation

$$\alpha = \left(\frac{z_\alpha}{h}\right)^2 + \frac{z_\alpha}{h} \tag{2.26}$$

where the choice of $\alpha = -0.39$ minimizes the maximum error in wave phase speed. This choice corresponds to $z_\alpha = -0.53h$. Kennedy et al. [2] introduced a moving reference level which allows for more flexibility in optimizing the non-linear behaviour. The "datum invariant" form of Kennedy et al. is adopted here:

$$z_\alpha = \zeta h + \beta\eta \tag{2.27}$$

where $\zeta = -0.53$ and $\beta = 1 + \zeta = 0.47$. The reference level is thus placed at 53% of the total local water depth.

2.3 Wave breaking

Two wave breaking schemes are included in FUNWAVE. The first one is using the capabilities of shock-capturing of the total variation diminishing (TVD) scheme. It uses the approach of Tonelli and Petti [3] who applied the TVD scheme to the non-linear shallow water equations (NSWE) to model hydraulic jumps. These equations are similar to the model equations, but without any of the dispersion terms arising from $\bar{\mathbf{u}}_2$. They are

$$\frac{\partial\eta}{\partial t} + \nabla\cdot(H\mathbf{u}_\alpha) = 0 \tag{2.28}$$

$$\frac{\partial H\mathbf{u}_\alpha}{\partial t} + \nabla\cdot[H\mathbf{u}_\alpha\mathbf{u}_\alpha] + \nabla\left[\frac{1}{2}g(\eta^2 + 2h\eta)\right] = -H\mathbf{R} + g\eta\nabla h \tag{2.29}$$

In FUNWAVE this is incorporated by switching from the Boussinesq equations to the NSWE if the cell Froude number exceeds a certain threshold. The ratio of the surface elevation to the water depth is used as a criterion to switch. The other one, which was originally implemented in previous versions of FUNWAVE, uses an artificial eddy-viscosity scheme in order to model the dissipating energy.

Chapter 3

Numerical solver

The model equations introduced in the previous chapter will be numerically solved. This consists out of two parts: a spatial and a time part. The spatial part is solved with a MUSCL-TVD scheme for which a fourth and second order version can be chosen. The time integration is done with a third order Runge-Kutta scheme.

3.1 Compact form of governing equations

In the previous chapter quite some variables were introduced. To prevent confusion the following notation is adopted here:

$$\begin{aligned}
\mathbf{u}_\alpha &= (u, v), \\
\bar{\mathbf{u}}_2 &= (U_4, V_4), \\
\mathbf{M} &= (P, Q) = H[u + U_4, v + V_4], \\
\mathbf{V}_1' &= (U_1', V_1'), \\
\mathbf{V}_1'' &= (U_1'', V_1''), \\
\mathbf{V}_2 &= (U_2, V_2), \\
\mathbf{V}_3 &= (U_3, V_3), \\
\mathbf{V} &= (U, V) = H[u + U_1', v + V_1'].
\end{aligned}$$

The general conservative form can be written as

$$\frac{\partial \mathbf{\Psi}}{\partial t} + \nabla \cdot \mathbf{\Theta}(\mathbf{\Psi}) = \mathbf{S} \tag{3.1}$$

where $\mathbf{\Psi}$ and $\mathbf{\Theta}(\mathbf{\Psi})$ are the vector of conserved variables and flux vector function, which are given by

$$\mathbf{\Psi} = \begin{pmatrix} \eta \\ U \\ V \end{pmatrix}, \quad \mathbf{\Theta}(\mathbf{\Psi}) = \begin{pmatrix} P\mathbf{i} + Q\mathbf{j} \\ [\frac{P^2}{H} + \frac{1}{2}g(\eta^2 + 2\eta h)]\mathbf{i} + \frac{PQ}{H}\mathbf{j} \\ \frac{PQ}{H}\mathbf{i} + [\frac{Q^2}{H} + \frac{1}{2}(\eta^2 + 2\eta h)]\mathbf{j} \end{pmatrix}. \tag{3.2}$$

The source function $\mathbf{S}$ on the right-hand side is

$$\mathbf{S} = \begin{pmatrix} 0 \\ g\eta h_{,x} + \psi_x - HR_x \\ g\eta h_{,y} + \psi_y - HR_y \end{pmatrix} \tag{3.3}$$

where

$$\psi_x = \eta_{,t}(U_1' - U_4) + H(uU_{4,x} + vU_{4,y} + U_4 u_{,x} + V_4 u_{,y} - U_1" - U_2 - U_3) \tag{3.4}$$

$$\psi_y = \eta_{,t}(V_1' - V_4) + H(uV_{4,x} + vV_{4,y} + U_4 v_{,x} + V_4 v_{,y} - V_1" - V_2 - V_3) \tag{3.5}$$

3.2 Spatial discretization

The spatial discretization is done with a hybrid finite volume-finite difference scheme. A high-order MUSCL-TVD scheme is used for the flux and first-order derivative terms. A central difference scheme is used at the cell centroid for the higher derivative terms.

Fourth order scheme

The fourth order construction at the cell interfaces is given by

$$\phi^L_{i+1/2} = \phi_i + \frac{1}{6}[\chi(r)\Delta^*\phi_{i-1/2} + 2\chi(1/r)\Delta^*\phi_{i+1/2}], \tag{3.6}$$

$$\phi^R_{i-1/2} = \phi_i - \frac{1}{6}[2\chi(r)\Delta^*\phi_{i-1/2} + \chi(1/r)\Delta^*\phi_{i+1/2}] \tag{3.7}$$

where $\phi^L_{i+1/2}$ is the constructed value at the left-hand side of the interface $i+\frac{1}{2}$', $\phi^R_{i-1/2}$ is the constructed value at the right-hand side of $i-\frac{1}{2}$ and $\chi(r)$ is the limiter function which is defined later on. The values of $\Delta^*\phi$ are evaluated by

$$\Delta^*\phi_{i+1/2} = \Delta\phi_{i+1/2} + \frac{1}{6}\Delta^3\bar{\phi}_{i+1/2} \tag{3.8}$$

$$\Delta\phi_{i+1/2} = \phi_{i+1} - \phi_i, \tag{3.9}$$

$$\Delta^3\bar{\phi}_{i+1/2} = \Delta\bar{\phi}_{i+3/2} - 2\Delta\bar{\phi}_{i+1/2} + \Delta^3\bar{\phi}_{i-1/2}, \tag{3.10}$$

$$\Delta\bar{\phi}_{i-1/2} = \text{minmod}(\Delta\phi_{i-1/2}, \Delta\phi_{i+1/2}, \Delta\phi_{i+3/2}), \tag{3.11}$$

$$\Delta\bar{\phi}_{i+1/2} = \text{minmod}(\Delta\phi_{i+1/2}, \Delta\phi_{i+3/2}, \Delta\phi_{i-1/2}), \tag{3.12}$$

$$\Delta\bar{\phi}_{i+3/2} = \text{minmod}(\Delta\phi_{i+3/2}, \Delta\phi_{i-1/2}, \Delta\phi_{i+1/2}). \tag{3.13}$$

Here minmod represent the Minmod limiter which is given by

$$\text{minmod}(a,b,c) = \text{sign}(a)\max(0, \min[|a|, 2\,\text{sign}(a)b, 2\,\text{sign}(a)c]). \tag{3.14}$$

The limiter function χ in equations 3.6 and 3.7 is the Van-Leer limiter which is

$$\chi(r) = \frac{r+|r|}{1+r} \tag{3.15}$$

where r is

$$r = \frac{\Delta^*\phi_{i+1/2}}{\Delta^*\phi_{i-1/2}}. \tag{3.16}$$

Second order scheme

If the fourth order scheme does not converge there is the option to use a second order MUSCL-TVD scheme with Van-Leer limiter. In this scheme the reconstruction at the interface is

$$\phi^L_{i+1/2} = \phi_i + \frac{1}{2}\Delta x\sigma\phi_i, \quad \phi^R_{i-1/2} = \phi_i - \frac{1}{2}\Delta x\sigma\phi_i \tag{3.17}$$

where

$$\sigma\phi_i = \Upsilon\left(\frac{\phi_{i+1}-\phi_i}{\Delta x}, \frac{\phi_i - \phi_{i-1}}{\Delta x}\right). \tag{3.18}$$

The Van-Leer limiter $\Upsilon(a,b)$ is here given by

$$\Upsilon(a,b) = \frac{a|b| + |a|b}{|a|+|b|}. \tag{3.19}$$

Flux at interface

For the computation of the numerical fluxes at the interfaces a Harten-Lax-van Leer (HLL) approximate Riemann solver is used.

$$\mathbf{\Theta}(\mathbf{\Psi}^L, \mathbf{\Psi}^R) = \begin{cases} \mathbf{\Theta}(\mathbf{\Psi}^L) & \text{if } s_L \geq 0 \\ \mathbf{\Theta}^*(\mathbf{\Psi}^L, \mathbf{\Psi}^R) & \text{if } s_L < 0 < s_R \\ \mathbf{\Theta}(\mathbf{\Psi}^R) & \text{if } s_R \leq 0 \end{cases}, \tag{3.20}$$

where

$$\mathbf{\Theta}^*(\mathbf{\Psi}^L, \mathbf{\Psi}^R) = \frac{s_R\mathbf{\Theta}(\mathbf{\Psi}^L) - s_L\mathbf{\Theta}(\mathbf{\Psi}^R) + s_L s_R(\mathbf{\Psi}^R - \mathbf{\Psi}^L)}{s_R - s_L}. \tag{3.21}$$

The left and right wave speeds are

$$s_L = \min(\mathbf{V}^L \cdot \mathbf{n} - \sqrt{g(h+\eta)^L}, u_s - \sqrt{\psi_s}), \tag{3.22}$$

$$s_R = \max(\mathbf{V}^R \cdot \mathbf{n} - \sqrt{g(h+\eta)^R}, u_s + \sqrt{\psi_s}) \tag{3.23}$$

with $\mathbf{n}$ being the normalised outward pointing cell side vector.

3.3 Time discretization

The time integration scheme is a third-order Strong-Stability Preserving (SSP) Runge-Kutta scheme. The scheme is given by

$$\begin{aligned} \mathbf{\Psi}^{(1)} &= \mathbf{\Psi}^n + \Delta t\left(-\nabla \cdot \mathbf{\Theta}(\mathbf{\Psi}^n) + \mathbf{S}^n\right) \\ \mathbf{\Psi}^{(2)} &= \frac{3}{4}\mathbf{\Psi}^n + \frac{1}{4}[\mathbf{\Psi}^{(1)} + \Delta t\left(-\nabla \cdot \mathbf{\Theta}(\mathbf{\Psi}^{(1)}) + \mathbf{S}^{(1)}\right)] \\ \mathbf{\Psi}^{n+1} &= \frac{1}{3}\mathbf{\Psi}^n + \frac{2}{3}[\mathbf{\Psi}^{(2)} + \Delta t\left(-\nabla \cdot \mathbf{\Theta}(\mathbf{\Psi}^{(2)}) + \mathbf{S}^{(2)}\right)] \end{aligned} \tag{3.24}$$

where $\mathbf{\Psi}^{n+1}$ is the variable vector at the next time step and $\mathbf{\Psi}^{(1)}$ and $\mathbf{\Psi}^{(2)}$ are intermediate values. From $\mathbf{\Psi}$ the velocity (u, v) can be solved by a system of tridiagonal matrix equations formed by equation 2.23.

An adaptive time step is used which is based on the Courant-Friedrichs-Lewy (CFL) criterion

$$\Delta t = C \min\left(\min(\frac{\Delta x}{|u_{i,j}| + \sqrt{g(h_{i,j}+\eta_{i,j}}}), \min(\frac{\Delta y}{|v_{i,j}| + \sqrt{g(h_{i,j}+\eta_{i,j}}})\right) \tag{3.25}$$

where C is the Courant number.

Chapter 4

Model tests

In order to validate the FUNWAVE model several tests are performed. In these tests results of numerical simulations, performed with FUNWAVE, will be compared with experimental data.

4.1 1D solitary wave propagation

To start simple solitary wave inputs are generated and simulated. Solitary waves are a type of non-linear shallow water waves, which were first observed by John Scott Russell who called it the wave of translation. They are characterised by very long wave lengths and their ability to maintain their shape over a very long distance. The first theoretical work on these types of waves was done by Joseph Boussinesq. Later Korteweg and de Vries developed a simplified equation which is only capable of describing wave propagation in one way. The main benefit is that is has exact solutions. These solutions are called cnoidal wave solutions, which are formed by Jacobi elliptic functions and are valid for wavelengths larger than 10 times the water depth. In the solitary wave or infinitely wavelength limit the solution for the free surface of such a wave reduces to

$$\eta(x,t) = A\,\mathrm{sech}^2\left(\frac{x-ct}{\Delta}\right) \tag{4.1}$$

where A is the amplitude,

$$\Delta = h\sqrt{\frac{4h}{3A}} \tag{4.2}$$

and

$$c = \sqrt{g(h+A)}. \tag{4.3}$$

To use this solution as an initial condition in FUNWAVE an initial velocity field is needed. This can be described by

$$u(x,t) = \frac{c}{h}\eta \tag{4.4}$$

In Lei [4] it was found this velocity initialisation produced tail waves, which were reduced by reducing the initial velocity with a reduction factor. This will also be taken into account here as

$$u_{red} = f_{red} \cdot \frac{c}{h}\eta \tag{4.5}$$

with f_{red} an arbitrary number between 0 and 1. In the following cases $f_{red} = 0.85$ is used. The last initialisation option is a build-in initialisation in FUNWAVE.

Simulations are done for waves with still water depth $h = 0.05m$ and wave amplitude $A = 0.01$. The domain is 10 meters long and $\Delta x = 0.01m$. The adaptive time step is based on $C = 0.5$. The results from the different initialisations is shown in figure 4.1. It shows all velocity initialisations result in tail waves. The reduced velocity initialisation shows the least amount of tail waves, while the wave amplitude is best matched by the initialisation of FUNWAVE. Furthermore the velocity initialisation according to equation 4.4 results in a too large amplitude. This also results in the wave being ahead of the exact one, which is calculated with equation 4.1, as the wave speed is a bit higher.

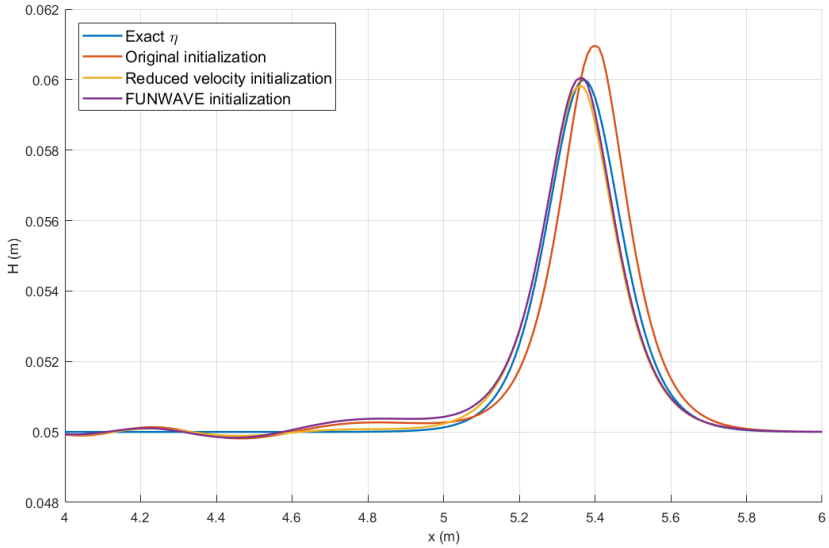


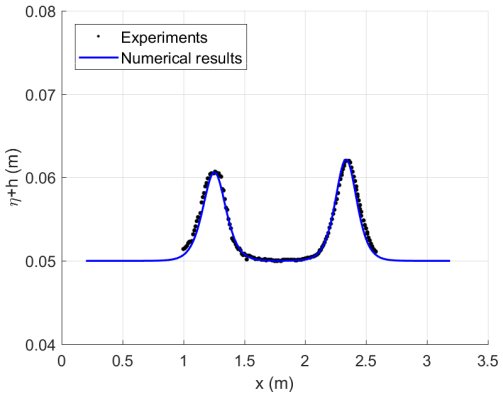
Figure 4.1: Wave profile after five seconds for different initialisations with $h = 0.05m$ and $A = 0.01m$

4.2 1D solitary wave collisions

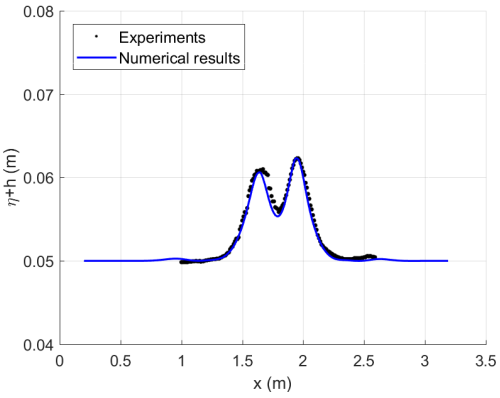
Experiments on solitary wave interactions were conducted by Craig et al. [5]. This was done in an aligned wave channel in the W.G. Prichard Fluid Mechanics Laboratory of Penn State University. The wave channel has a length of $13.165m$ and a width of $25.4cm$. The still water depth was $5.0cm$ with an accuracy of $0.25mm$. With a horizontal piston-like motion of a wave paddle highly repeatable solitary waves were generated. With a bottom-mounted pressure transducer and four non-contacting wave gauges on a carriage the water surface was measured. Two cases were considered: head-on and overtaking collisions. These will be discussed in the next sections. The experimental data shown is a combination of 40 repeated experiments.

4.2.1 Head-on collision

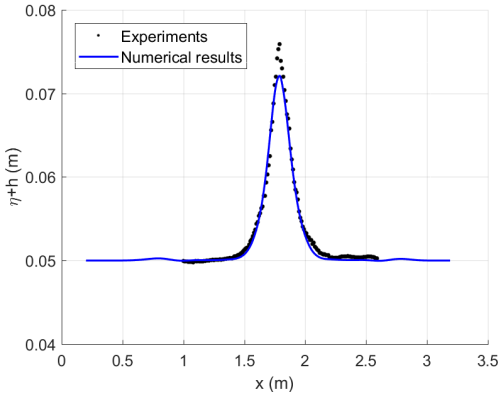
The head-on collision is formed by two solitary waves with different amplitudes travelling in opposite direction. The waves are initialised as the sum of two solitary waves with the free surface and velocity according to equations 4.1 and 4.5 with the reduction factor $f_{red} = 0.85$. The computational domain is $2.5m$ and the computational time is $1.5s$. The grid resolution is $\Delta x = 0.01m$ and the time stepping is based on $C = 0.5$. The results of the simulation are shown in figure 4.2. In the numerical simulation bottom friction was included after seeing the huge impact on the overtaking collision, which is discussed in the next section. No change of the results was observed in the head-on case, which can be explained by the shorter simulation time and related travelled distance. The numerical results are different from the experimental data. It appears the computed wave collision is ahead of the measured data. If we compare it with the graphs in Craig et al. [5] the results look more like the summation of two Korteweg-de Vries (KdV) solitons. The last frame shows that after the collapse the wave profiles are similar, thus the end result of numerical and experimental data is the same. To research the behaviour of the collision the wave height is extracted a bit earlier in time than the experimental time. This in time shifted wave height is plotted in figure 4.3. The time shifts range from 0 up to $35ms$, which is close to the time difference between the subsequent experimental data. From the shifted pictures it can be concluded that the collision behaviour is similar.



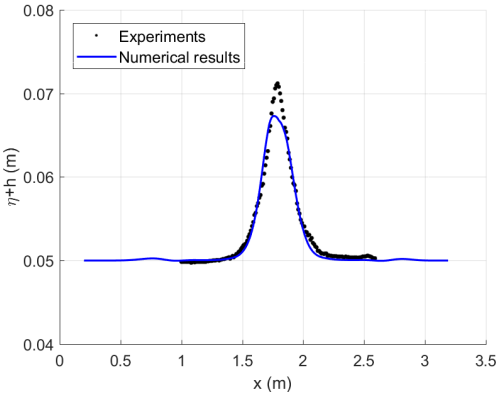
(a) $t = 0s$



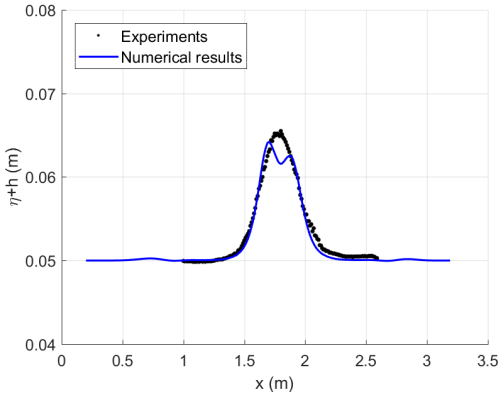
(b) $t = 0.5007s$



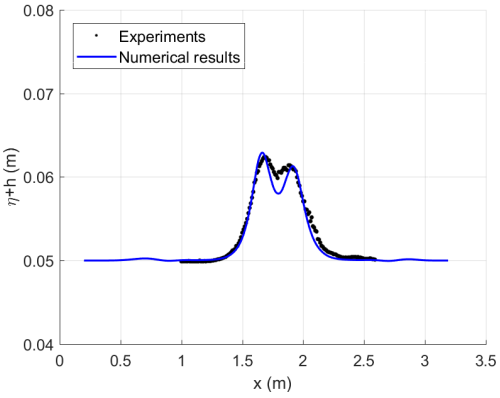
(c) $t = 0.7526s$



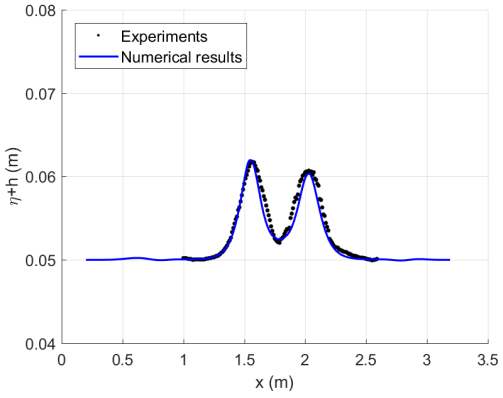
(d) $t = 0.8018s$



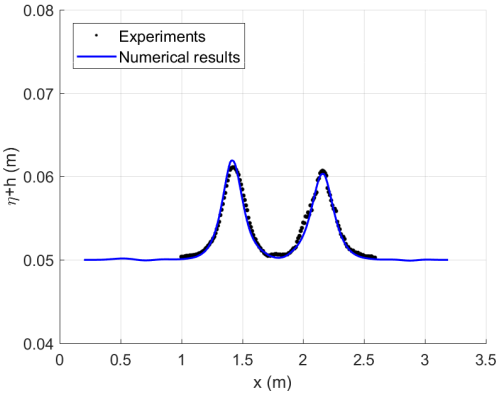
(e) $t = 0.8510s$



(f) $t = 0.8940s$

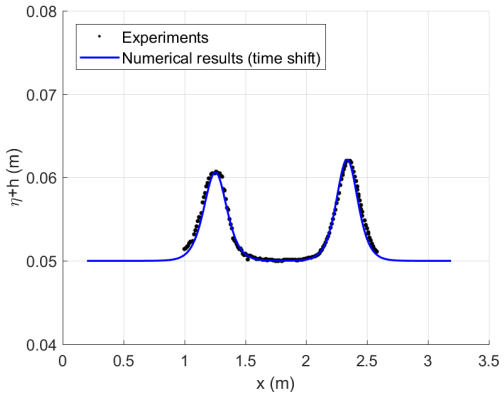


(g) $t = 1.0291s$

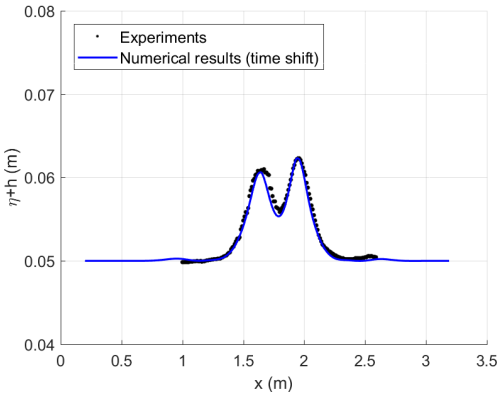


(h) $t = 1.2012s$

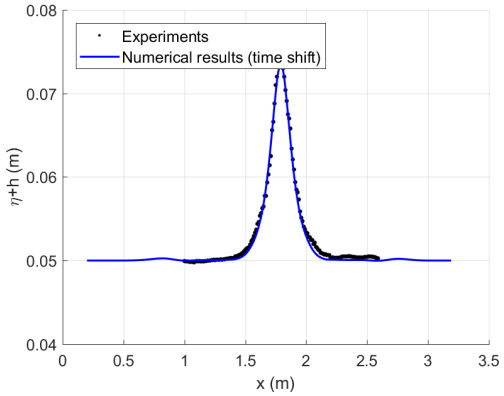
Figure 4.2: Head-on collision between two solitary waves with amplitudes $A_1 = 0.01217m$ and $A_2 = 0.01063m$



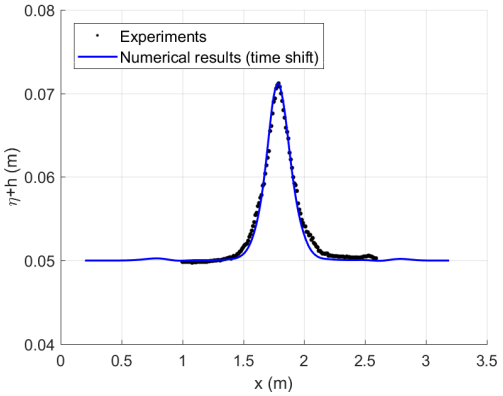
(a) $t = 0s$



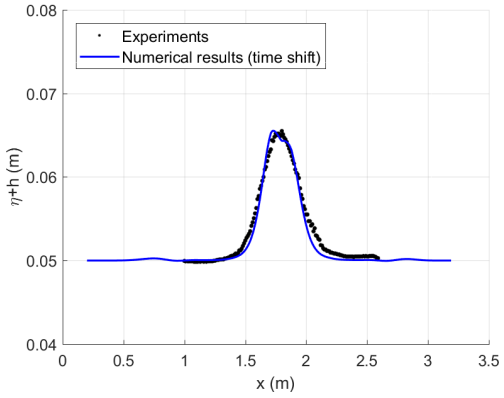
(b) $t = 0.5007s$



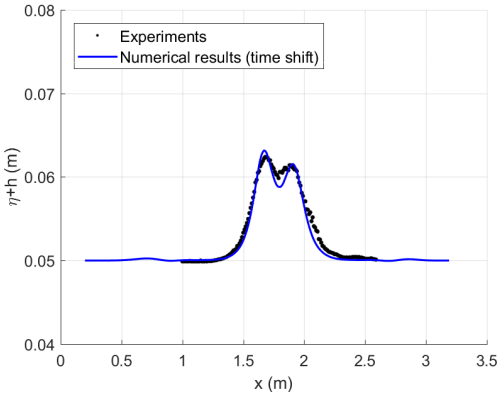
(c) $t = 0.7526s$



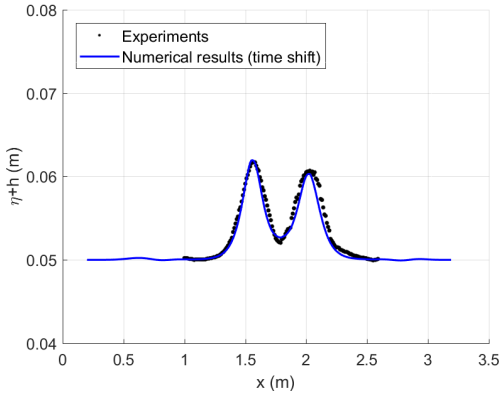
(d) $t = 0.8018s$



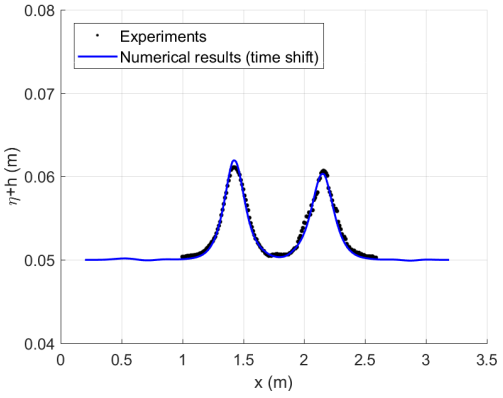
(e) $t = 0.8510s$



(f) $t = 0.8940s$



(g) $t = 1.0291s$



(h) $t = 1.2012s$

Figure 4.3: Head-on collision between two solitary waves with amplitudes $A_1 = 0.01217m$ and $A_2 = 0.01063m$ with shifted numerical results

4.2.2 Overtaking collision

For the overtaking collision two waves are generated shortly after each other. They will propagate in the same direction, but the second one has a larger amplitude and thus a larger speed than the first one. This will result in the second wave overtaking the first one. As the velocity difference is small, the collision is going on over a relatively large distance. To capture the wave height the instrumental cart moved with the waves in the experiments. In the numerical simulation the domain is $10m$ long and the numerical results are translated with the average wave velocity multiplied by the time to make them coincide with the experimental results. The domain has a spatial resolution of $\Delta x = 0.01m$ and a time step based on $C = 0.5$. Initially the model was run without any friction, but it can be seen in figure 4.4 that this gives incorrect results. In the real situation there is friction, so it should also be modelled. A quadratic friction function with a fixed friction coefficient is implemented in the code according to:

$$\mathbf{R_{bottom}} = \frac{C_d}{H}\mathbf{U}|\mathbf{U}| \tag{4.6}$$

It was found a friction factor of $C_d = 0.03$ gives the best agreement with the experimental data. The numerical results show a good agreement with the experimental data.

The translations calculated with the wave velocities according to equation 4.3 were found to be insufficient. The numerical results needed to be translated with a small extra percentage (+2% at most) to make them coincide. The movement of the cart in the experiments is not exactly known either, so the extra translation is regarded to be within this error margin. The used translations are given in table 4.1.

Frame	$t[s]$	$Translation[m]$
2	2.5989	2.1150
3	3.5021	2.8500
4	4.1472	3.3750
5	4.6971	3.8225
6	5.5972	4.5550
7	6.6017	5.3462
8	8.3989	6.7680

Table 4.1: Translations overtaking collision

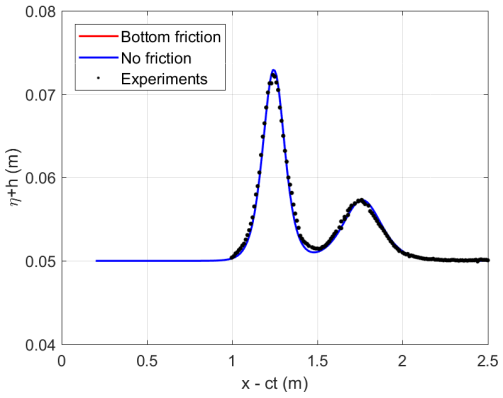
There was found some unwanted behaviour. Without friction the overtaking wave has a larger amplitude after the collision than at the start. This could be a sign of an instability in the numerical model and result in increasing energy during the simulation. As FUNWAVE estimates the energy and outputs it in a log file it is possible to see if the energy increases during the simulation. The estimated energy is calculated with

$$E_{est} = \sum_{i,j}\left\{\frac{1}{2}[gH^2(i,j) + u_\alpha^2(i,j) + v_\alpha^2(i,j)]\Delta x \Delta y\right\} \tag{4.7}$$

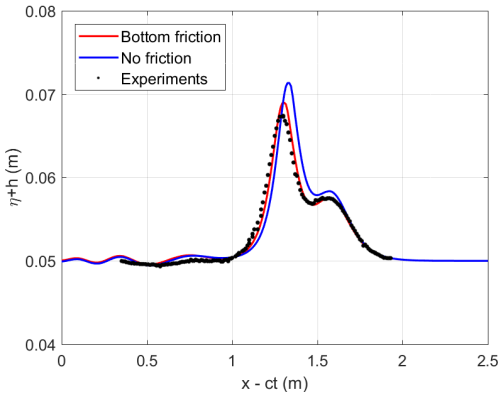
which is the energy divided by density. The estimation is plotted in figure 4.5 against time, where it can be seen that the energy does not increase during the simulation. There can be seen some negative spikes, which could not be explained.

To look into possible instability issues in the code the CFL condition is changed to $C = 0.1$. It is done for the cases with and without friction. The final frame for both CFL conditions is plotted in figures 4.6b and 4.7b. There can be seen no differences between these, so it can be concluded the CFL condition of $C = 0.5$ is suitable for both stability and accuracy.

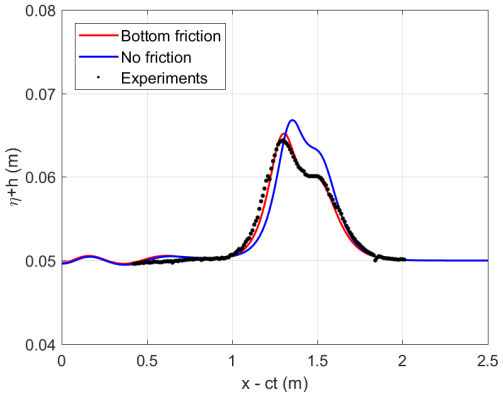
Next to this the fourth and second order spatial schemes are compared as instabilities were found in the past for the fourth order scheme. The same comparison is made as for the CFL condition. In figures 4.6a and 4.7a it can be seen there are only small differences between the two schemes. In total it can be concluded the higher output amplitude is not caused by numerical issues, but comes from the governing equations itself. The performance of the model when friction is modelled shows that it performs good.



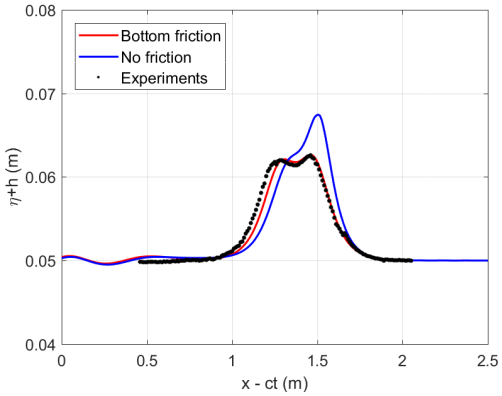
(a) $t = 0s$



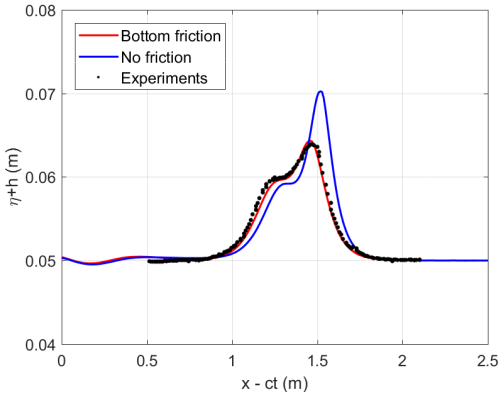
(b) $t = 2.5989s$



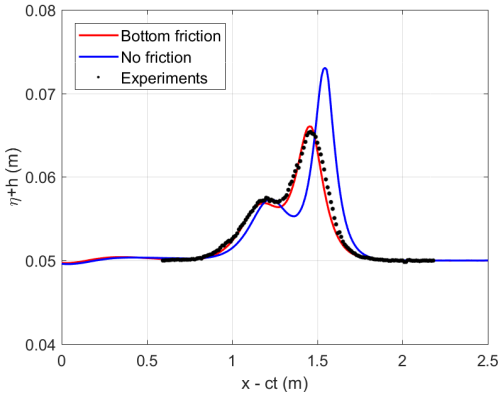
(c) $t = 3.5021s$



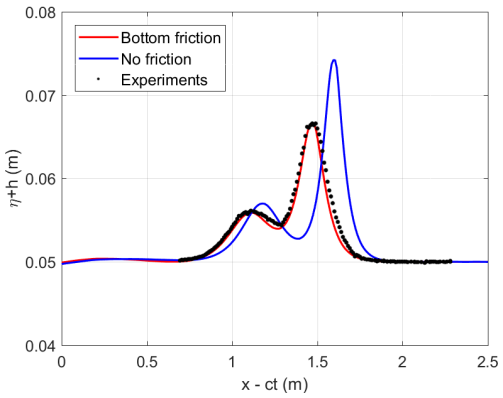
(d) $t = 4.1472s$



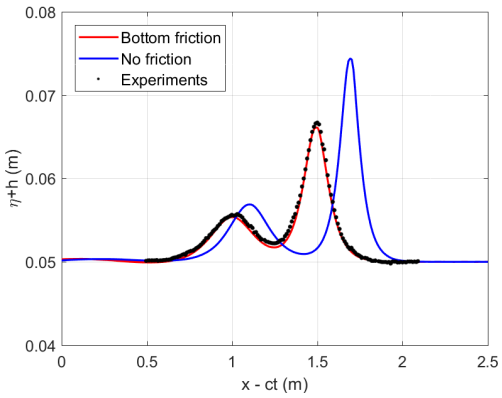
(e) $t = 4.6971s$



(f) $t = 5.5972s$



(g) $t = 6.6017s$



(h) $t = 8.3989s$

Figure 4.4: Overtaking collision between two solitary waves with amplitudes $A_1 = 0.023m$ and $A_2 = 0.0073m$

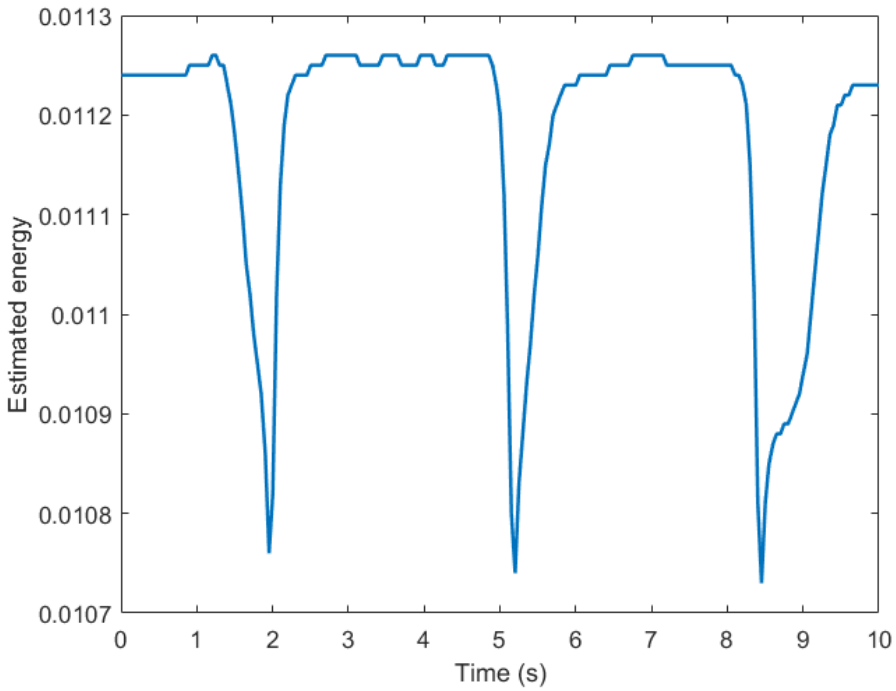
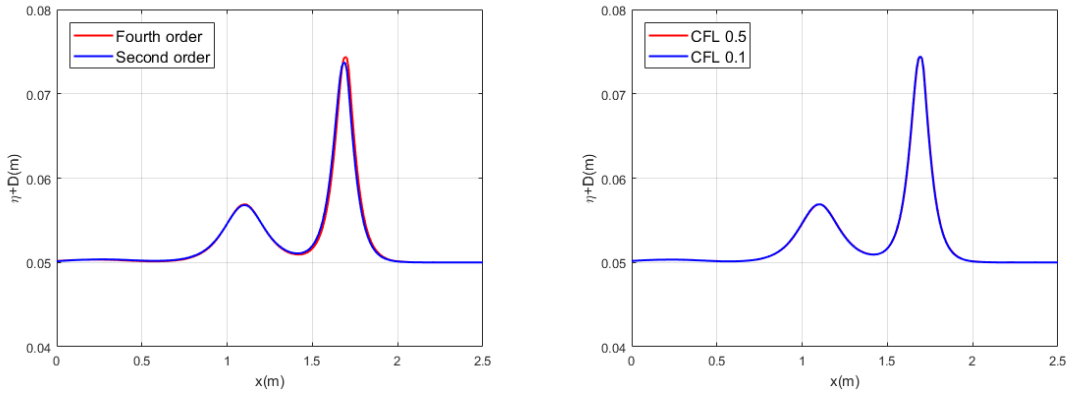


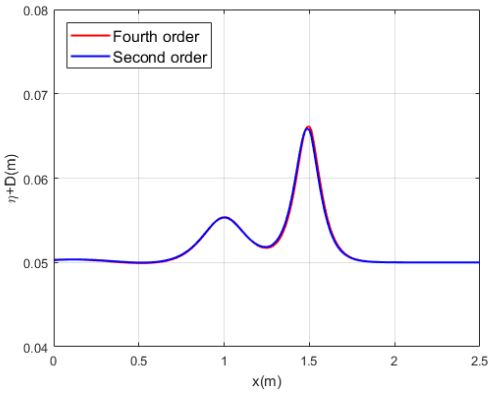
Figure 4.5: Estimated energy vs time



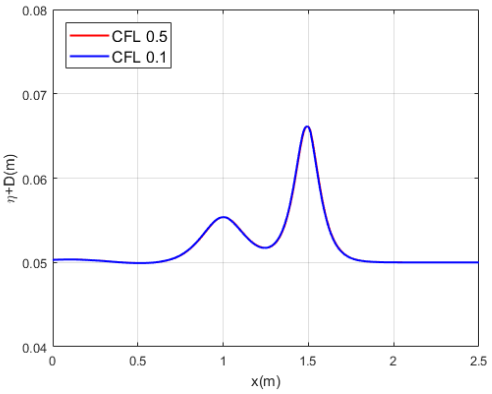
(a) Second order and fourth order spatial scheme

(b) CFL conditions $C = 0.1$ and $C = 0.5$

Figure 4.6: Last frame of overtaking collision with different numerical conditions and without friction



(a) Second order and fourth order spatial scheme

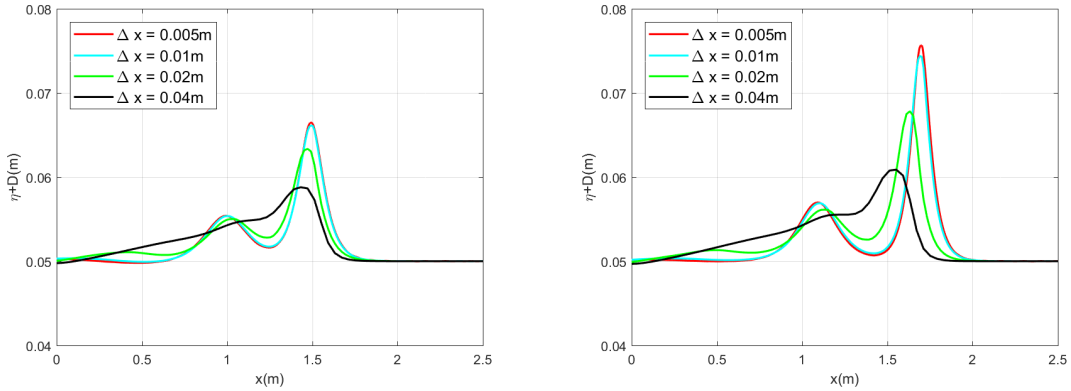


(b) CFL conditions $C = 0.1$ and $C = 0.5$

Figure 4.7: Last frame of overtaking collision with different numerical conditions and with bottom friction

4.2.3 Accuracy

The overtaking collision is also performed on different grid sizes. The results are shown in figure 4.8 for both the case with and without friction. The grid sizes used are $\Delta x = 0.005m$,$\Delta x = 0.01m$,$\Delta x = 0.02m$ and $\Delta x = 0.04m$. There only exist minor differences between the two smallest grid sizes, but the two larger grid sizes show more deviations. Especially $\Delta x = 0.04m$ is too coarse as on this grid the numerical results are not able to capture the right flow anymore. The smaller grid sizes and changing the spatial scheme from second to fourth order converge towards the same solution, which is another indicator that the numerical results are correct.



(a) With bottom friction

(b) Without bottom friction

Figure 4.8: Last frame of overtaking collision on different grid sizes

Chapter 5

Large scale simulations

The simulation of a large scale case is one of the end goals of this study. It is to demonstrate the possibilities of using FUNWAVE for modelling a wavefield as part of the full SprayIce project. The large scale simulation will be used with a variable bathymetry and model nesting is used to resolve waves on a finer grid.

5.1 Bathymetry

In order to simulate a realistic case water depth data is needed. This data can be obtained from the Norwegian Mapping Authority (Kartverket), which has depth data available all around the Norwegian coast. The data is available at different resolutions ($5m$,$25m$ and $50m$), but only the $50m$ grid is openly available. Within the 12 nautical mile coastal zone the higher resolutions are subject to military restrictions. If this would be necessary in the future permission can be requested. For now the $50m$ resolution is used and in case finer grids are needed the depth is interpolated.

The shoreline is of quite some importance for the reflections of waves. Steep shorelines will cause more reflection than gentle slopes. To this end also land data is obtained and combined into one map. The resulting map can be seen in figure 5.1.

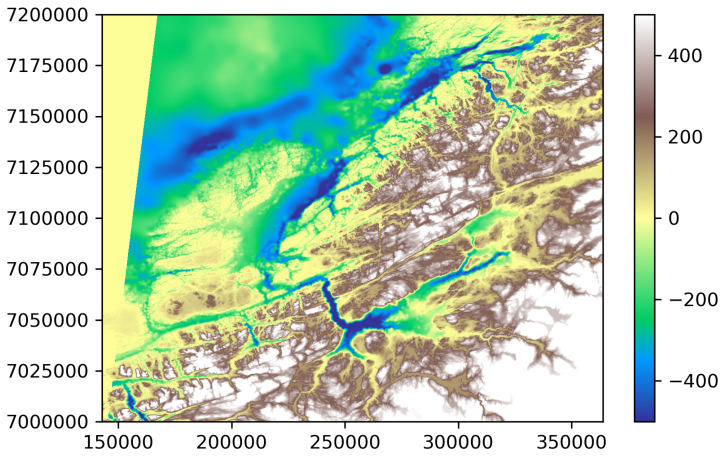
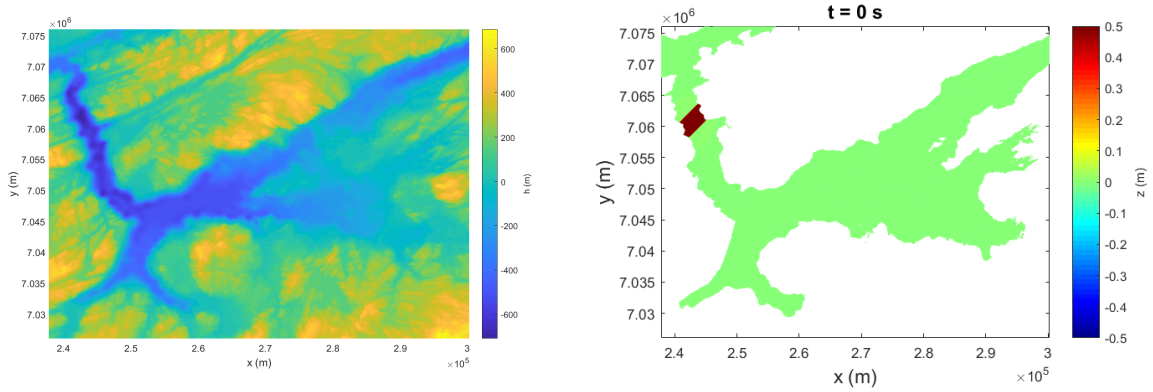


Figure 5.1: Combined water depth and land height map

5.2 Wave propagation in the Trondheimfjord

To demonstrate the capabilities of FUNWAVE a region around Trondheim is chosen where a demo flow case is performed. The water depth of this region can be seen in figure 5.2a. The region consists of 1000x1250 grid points with the grid spacing of $50m$ mentioned earlier. The flow is calculated by making use of 25 CPU's on a computing cluster, for which the domain is divided into rectangles of 200x250 grid points. The initialisation is done by setting part of the initial water height to $1m$ above the still water level, which is shown in figure 5.2b.



(a) Height map

(b) Initial water height

Figure 5.2: Height map and initial water height of the simulation region

The total simulation time is $1000s$ and the adaptive time stepping is used with $C = 0.5$. Running the simulation with the original bathymetry results in instabilities arising at the transition from water to land, mainly due to high gradients at some places. To resolve this issue the bathymetry is adapted where high gradients are present. This is done by a build-in iterative process in FUNWAVE. The resulting bathymetry is only slightly different from the original one and the influence on the flow of the waves is regarded as minor.

Next to adapting the bathymetry there is another parameter which causes instabilities. This is the minimum water depth, which is the minimum layer of water on the land. It is a numerical feature which is also used to determine if a cell is wet or dry. If H is smaller than the minimum water depth a cell is dry and if it is larger a cell is wet. It is preferably low as this resembles nature the best. However setting it too low lead to unstable simulations in the cases that were reviewed. In combination with adapting the bathymetry a value of $0.5m$ was used. Further information regarding the treatment of wet and dry cells can be found in the FUNWAVE manual [6] and Shi [7].

The resulting wave propagation in the fjord is animated in figure 5.3, which can be started by a click. In the animation the wave height is plotted every $5s$. In the animation flow features like build-up and withdrawal can be seen around the shoreline. Furthermore the amplitude is decreasing through the narrow entrance and diminishing when the flow enters the wider region. Besides the animation a sequence of figures is made, where every $60s$ the wave height is plotted. These can be seen in figures 5.4 and 5.5. To enlarge the contrast the colour scale varies from $-0.5m$ to $0.5m$

Although the numerical simulation is stable there are some waves which appear to come from an artificial source. They start out of nowhere in a point on the shoreline and do not seem to be a natural result of the flow shoreline interference. In figure 5.6 an example of these waves is shown at several time steps. It can be seen they initiate on the shoreline and spread in a circular pattern like a point source.

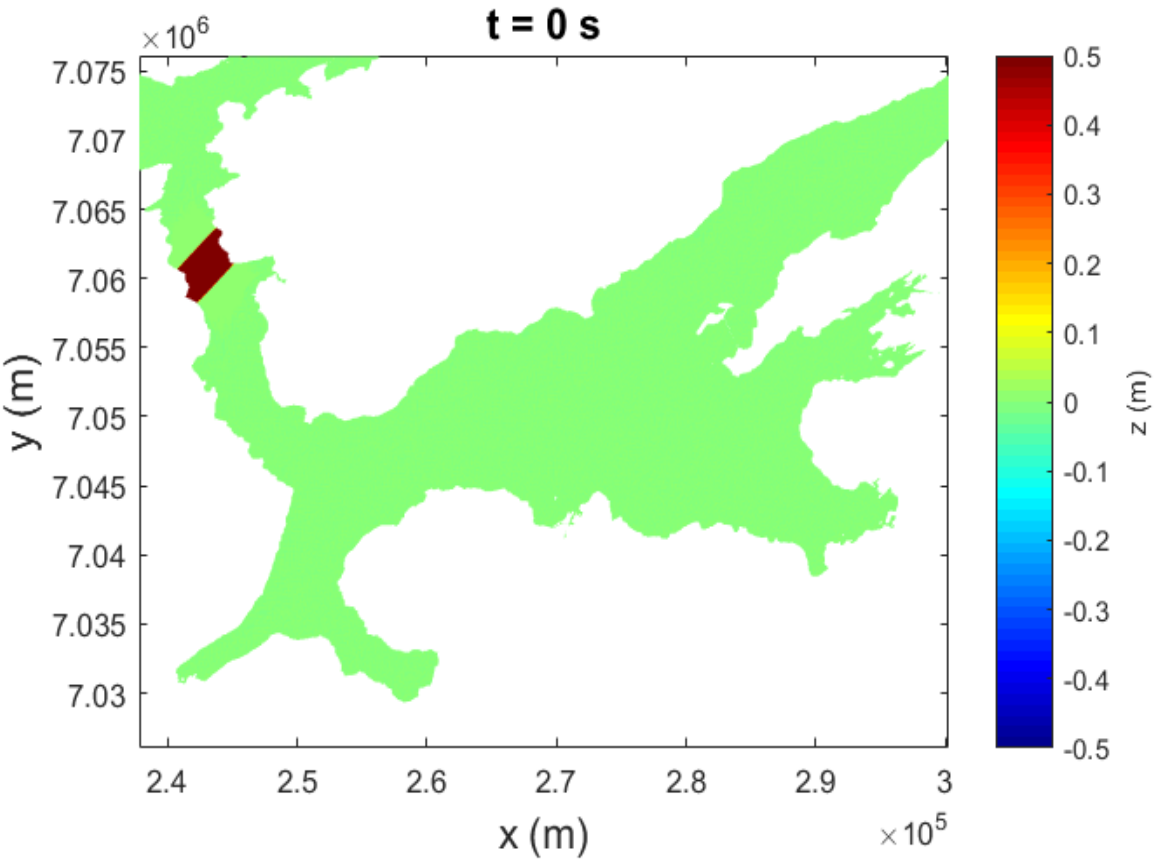
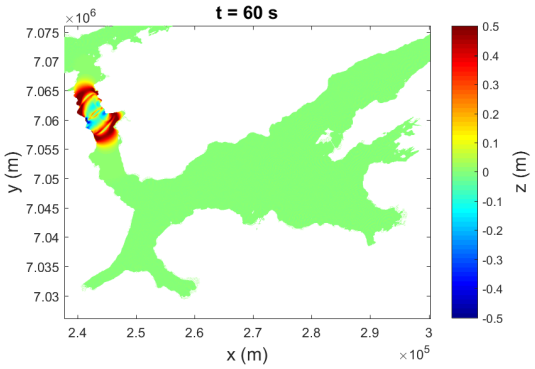
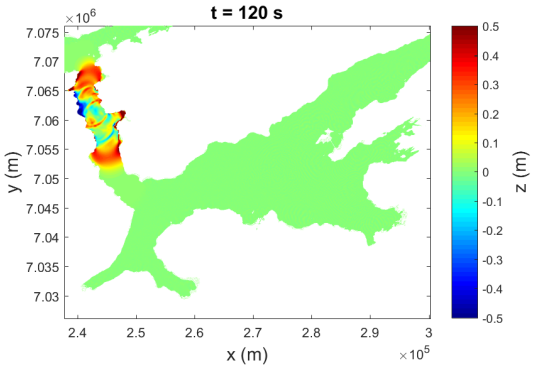


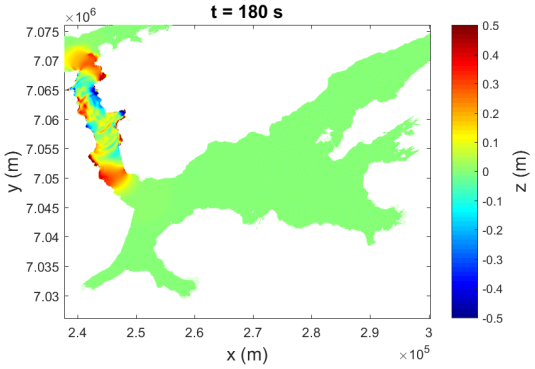
Figure 5.3: Wave propagation in the Trondheimfjord



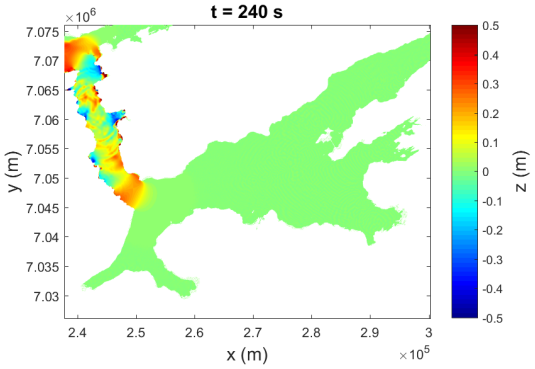
(a) $t = 60s$



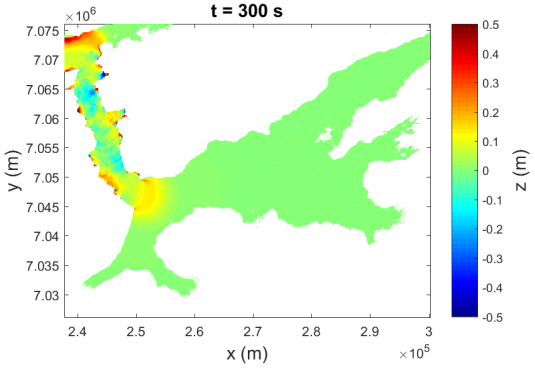
(b) $t = 120s$



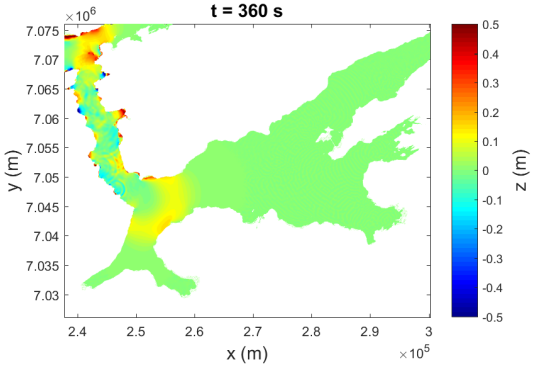
(c) $t = 180s$



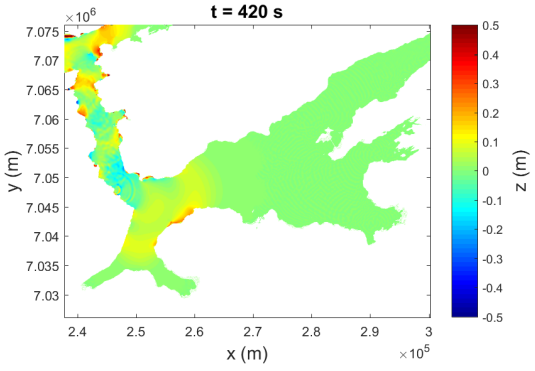
(d) $t = 240s$



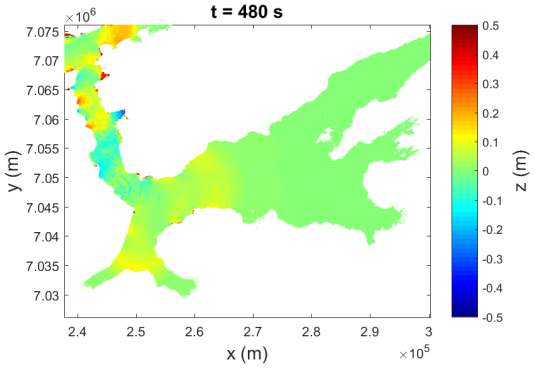
(e) $t = 300s$



(f) $t = 360s$

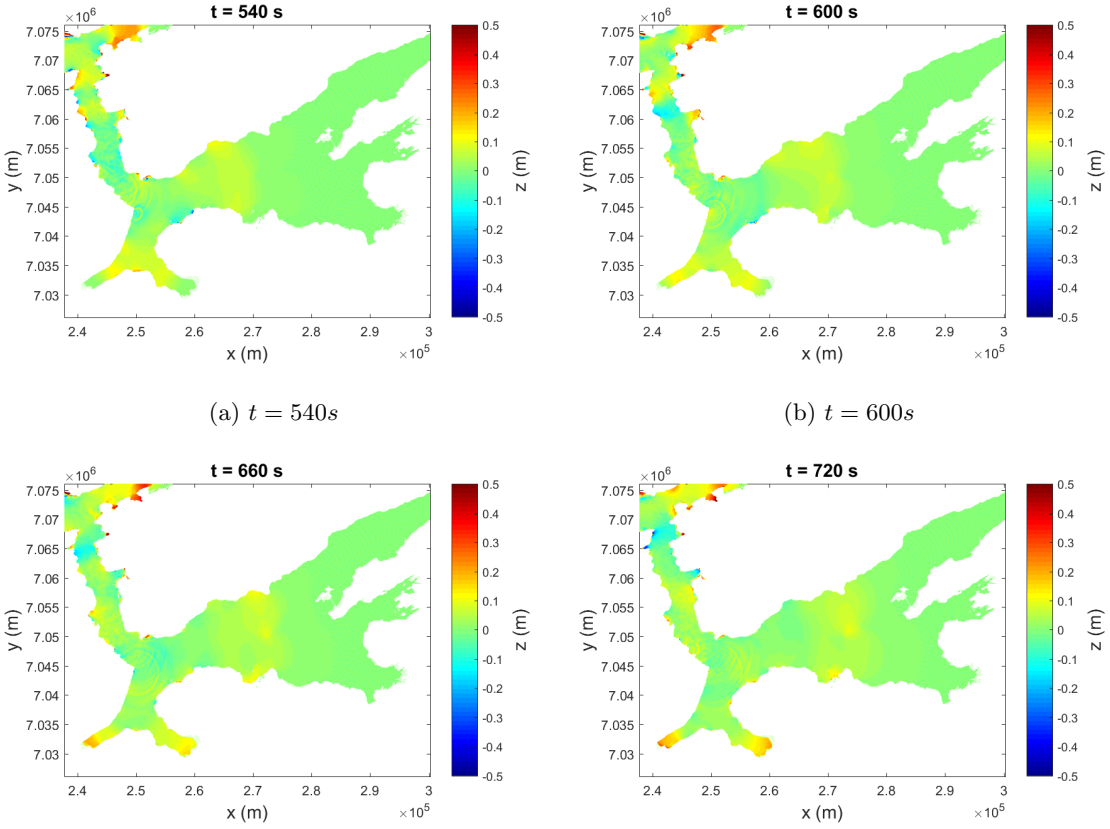


(g) $t = 420s$



(h) $t = 480s$

Figure 5.4: Flow propagation every $60s$ from $60s$ to $480s$



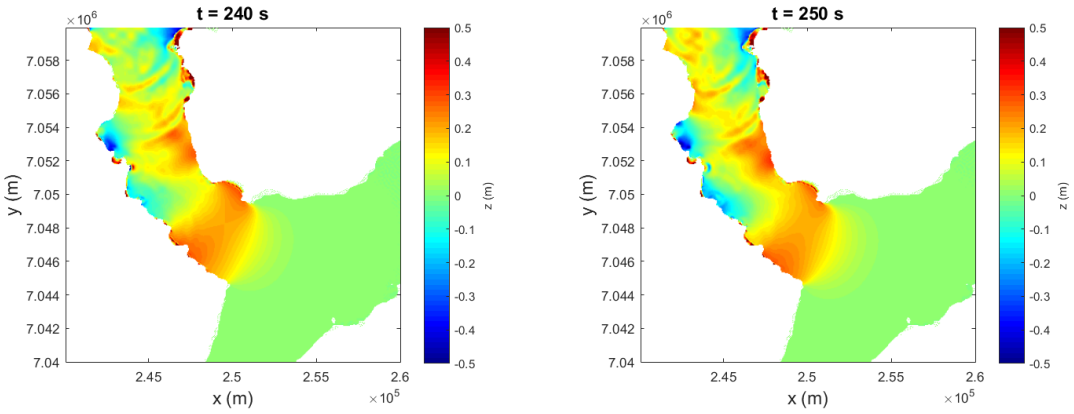
(a) $t = 540s$

(b) $t = 600s$

(c) $t = 660s$

(d) $t = 720s$

Figure 5.5: Flow propagation every $60s$ from $540s$ to $720s$



(a) $t = 240s$

(b) $t = 250s$

(c) $t = 265s$

(d) $t = 285s$

Figure 5.6: Artificial waves

5.3 Model nesting

Around a vessel or structure wave behaviour is required in more detail. In FUNWAVE coarse grid output data can be transferred to a finer grid simulation as imposed boundary conditions. To this end η, u and v data in grid points on the new boundary are outputted every time step. After the first simulation the data is interpolated and written to a file which will be read in the next simulation.

The part of the large domain which is used for the smaller case is shown in figure 5.7. The grid size is halved ($25m$) and for this purpose the height map is linearly interpolated. On a part of the eastern boundary the boundary conditions are imposed. The clock time is the same as the original simulation, which means waves will flow into the domain after $300s$.

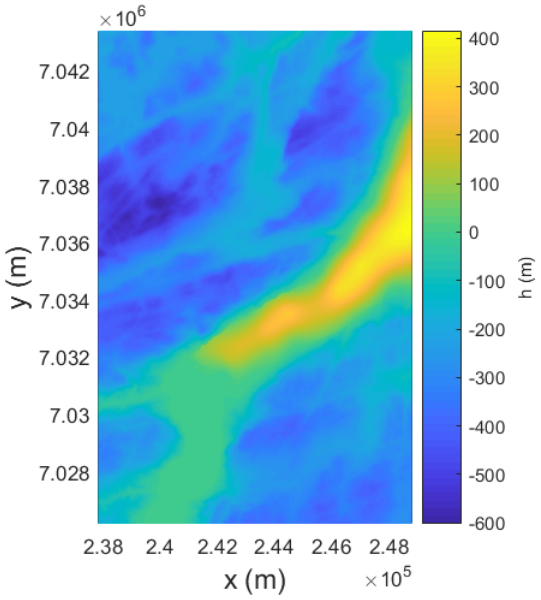


Figure 5.7: Height map of nesting region

Again an animation is made of the results and these are shown in figure 5.8, which can be started by a click. In this case the water elevation η is plotted every second. In the animation the build up of the wave height can be seen when it is running into the region with lower depth. Especially when it reaches the end the wave height builds up and the wave shortens. Next to this some run up onto the land is present. A sequence of images is plotted in figures 5.9 and 5.10.

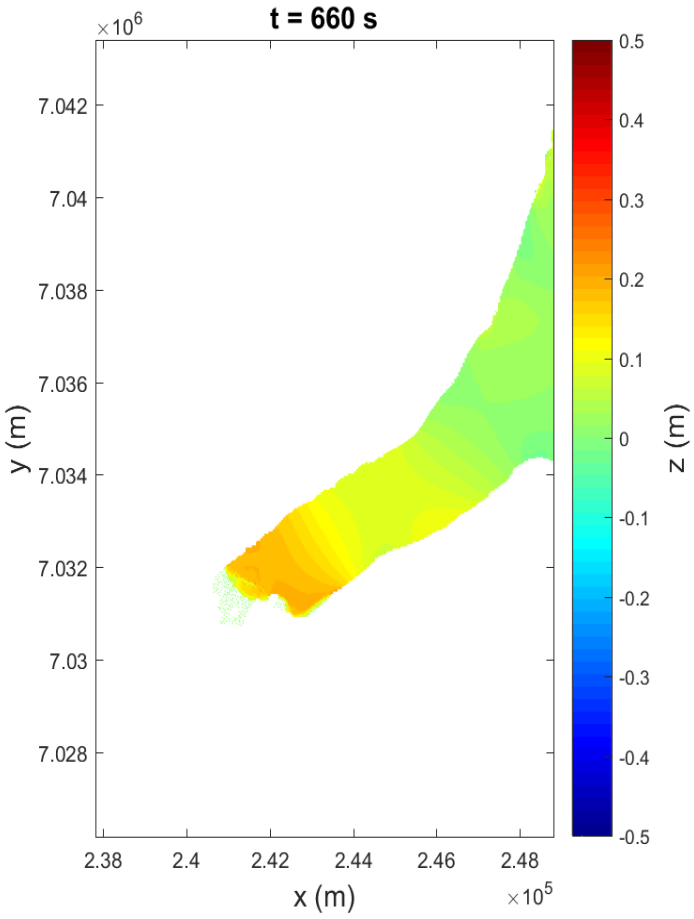
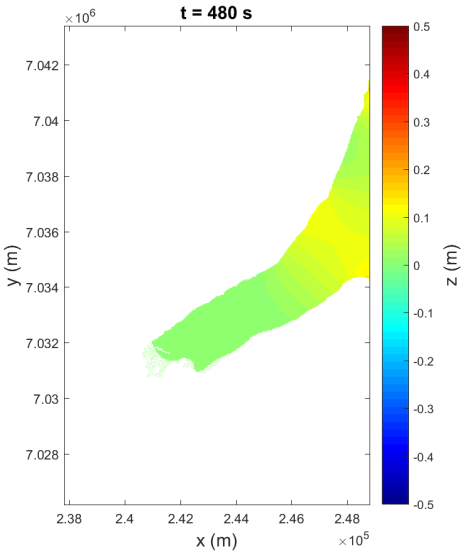
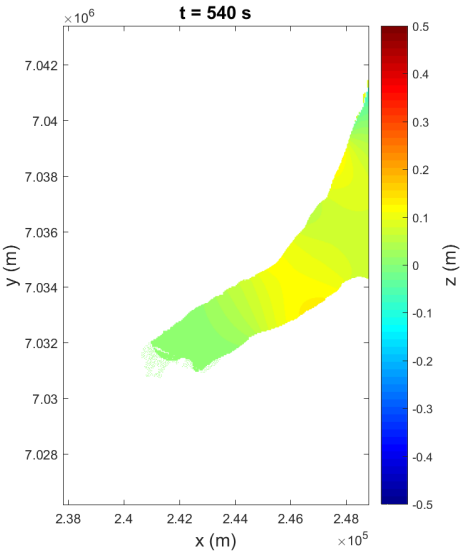


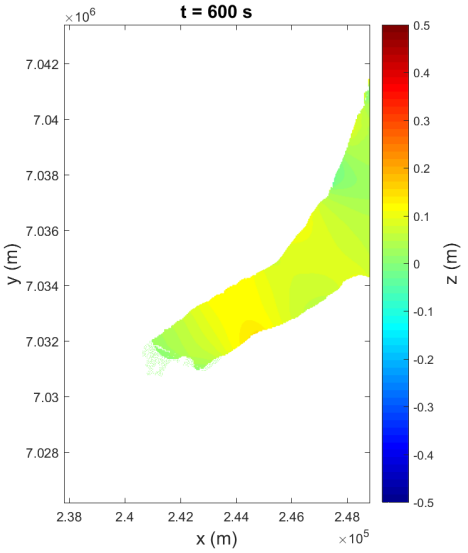
Figure 5.8: Propagation in the nested domain



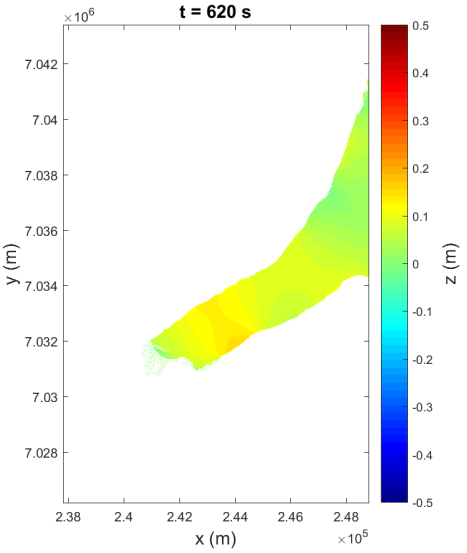
(a) $t = 480s$



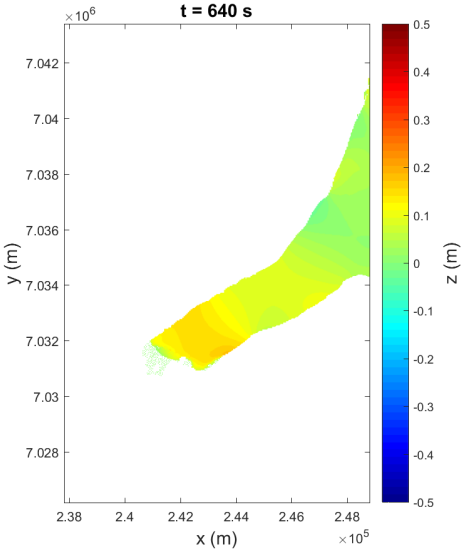
(b) $t = 540s$



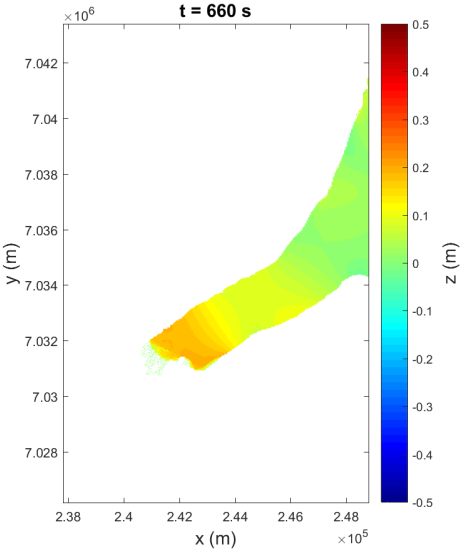
(c) $t = 600s$



(d) $t = 620s$

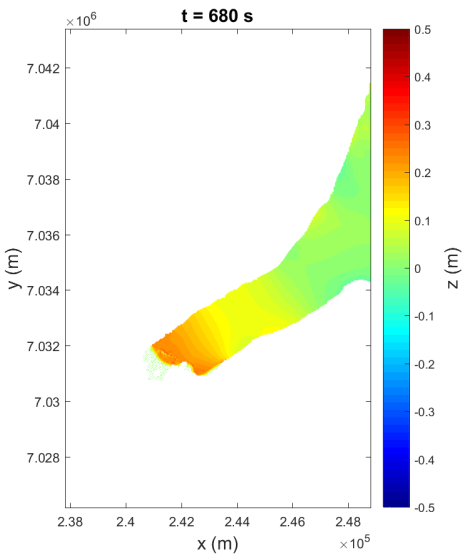


(e) $t = 640s$

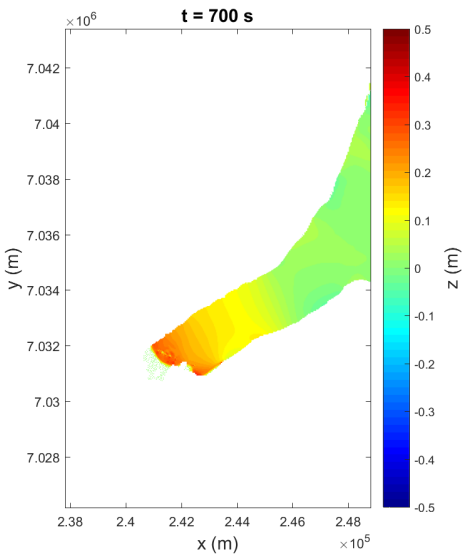


(f) $t = 660s$

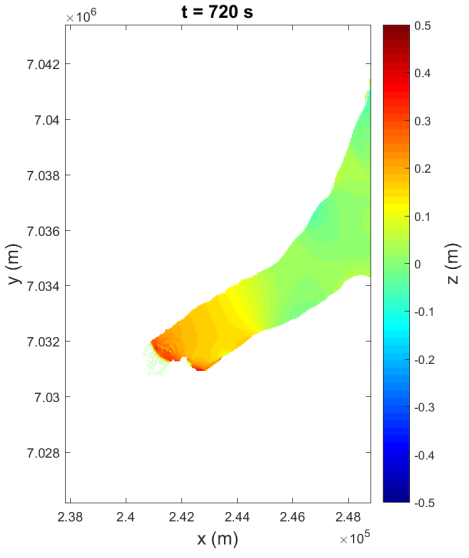
Figure 5.9: Flow propagation in nesting region



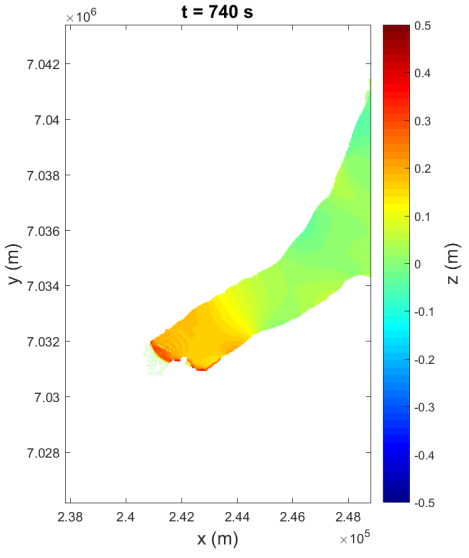
(a) $t = 680s$



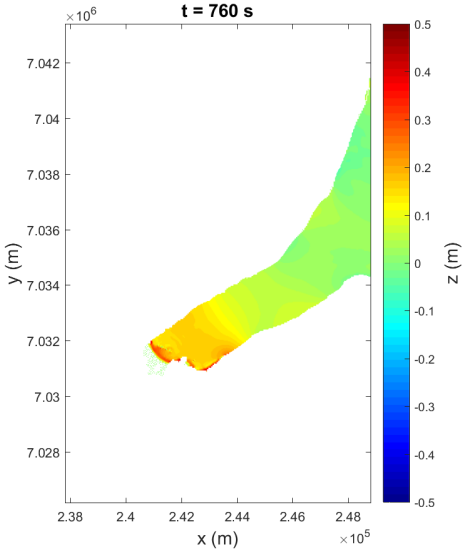
(b) $t = 700s$



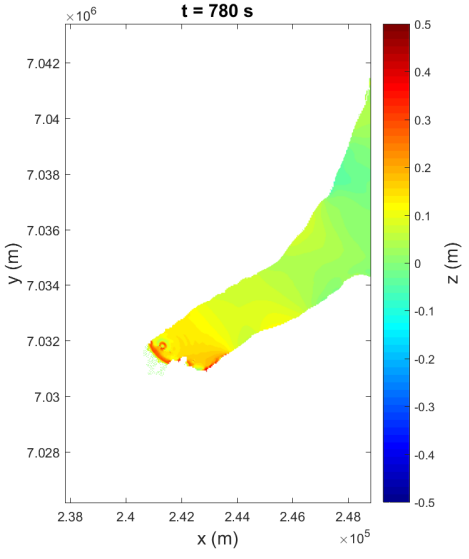
(c) $t = 720s$



(d) $t = 740s$



(e) $t = 760s$



(f) $t = 780s$

Figure 5.10: Flow propagation in nesting region

5.4 3D velocity reconstruction

As the goal is that the output of the 2D FUNWAVE simulations will be used as input cases for 3D simulations the horizontal output velocities have to be used to create a velocity profile over the depth. Let's recall the horizontal velocity from equation 2.7 in chapter 2.

$$\begin{aligned}\hat{\mathbf{u}} &= \mathbf{u}_\alpha + \mathbf{u}_2(z) \\ &= \mathbf{u}_\alpha + (z_\alpha - z)\nabla A + \frac{1}{2}(z_\alpha^2 - z^2)\nabla B \\ &= \mathbf{u}_\alpha + (z_\alpha - z)\nabla[\nabla\cdot(h\mathbf{u}_\alpha)] + \frac{1}{2}(z_\alpha^2 - z^2)\nabla[\nabla\cdot(\mathbf{u}_\alpha)]\end{aligned} \tag{5.1}$$

There is also need for a vertical velocity reconstruction as it is not part of the model. This can be done by using

$$\begin{aligned}w &= A + zB \\ &= \nabla\cdot(h\mathbf{u}_\alpha) + z\nabla\cdot\mathbf{u}_\alpha\end{aligned} \tag{5.2}$$

which is the equation used in Chen [1] to replace the vertical velocity. The total velocity vector then becomes

$$\begin{aligned}\mathbf{u} = \begin{pmatrix} u \\ v \\ w \end{pmatrix} &= \begin{pmatrix} u_\alpha \\ v_\alpha \\ 0 \end{pmatrix} + \begin{bmatrix} (z_\alpha - z)\frac{\partial}{\partial x} \\ (z_\alpha - z)\frac{\partial}{\partial y} \\ 1 \end{bmatrix} A + \begin{bmatrix} \frac{1}{2}(z_\alpha^2 - z^2)\frac{\partial}{\partial x} \\ \frac{1}{2}(z_\alpha^2 - z^2)\frac{\partial}{\partial y} \\ z \end{bmatrix} B \\ &= \begin{bmatrix} 1 + (z_\alpha - z)\frac{\partial^2 h}{\partial x^2} + \frac{1}{2}(z_\alpha^2 - z^2)\frac{\partial^2}{\partial x^2} & (z_\alpha - z)\frac{\partial^2 h}{\partial x \partial y} + \frac{1}{2}(z_\alpha^2 - z^2)\frac{\partial^2}{\partial x \partial y} \\ (z_\alpha - z)\frac{\partial^2 h}{\partial x \partial y} + \frac{1}{2}(z_\alpha^2 - z^2)\frac{\partial^2}{\partial x \partial y} & 1 + (z_\alpha - z)\frac{\partial^2 h}{\partial y^2} + \frac{1}{2}(z_\alpha^2 - z^2)\frac{\partial^2}{\partial y^2} \\ \frac{\partial}{\partial x} + z\frac{\partial}{\partial x} & \frac{\partial}{\partial y} + z\frac{\partial}{\partial y} \end{bmatrix} \begin{pmatrix} u_\alpha \\ v_\alpha \end{pmatrix}\end{aligned} \tag{5.3}$$

Chapter 6

Conclusions and recommendations

The open-source code FUNWAVE has been used to model water waves. The model is verified by means of modelling two solitary wave interactions. In this verification it has been observed that the models perform correctly. There were found some interesting things regarding these interactions. In the head-on case the collision seemed to be ahead of the experimental case, but the end result of the collision was the same. And in the overtaking wave case the wave speed deviated from the wave speed calculated from theory. The results needed to be translated with a few extra percents. Although these are deviations from the experimental results they are regarded as small and it is concluded that the code performs well.

The large scale tests showed that the code is quite sensible for instabilities. Mainly on the transition from land to water instabilities occur. These can be resolved by lowering the gradients in these regions, which means the simulated bathymetry slightly deviates from the real bahtymetry, and by using a higher minimum water depth, which influences if a cell is wet or dry. Although this results in a stable simulation there are some waves present which appear to be unnatural and could be an instability which does not develop.

Regarding these issues further work should be focused on investigating what causes the numerical issues and if these also occur in a region with high gradients, but without the transition to land. The minimum water depth and its effects should also be further researched, as it is a parameter which has a large influence on the stability.

Furthermore an extension of this work can be done in the direction of creating a realistic sea state as input to these simulations. This could be of extra value to the SprayIce project as it could also include a structure around which the waves are resolved on different scales by means of model nesting.

Bibliography

[1] Qin Chen. Fully nonlinear boussinesq-type equations for waves and currents over porous beds. *Journal of Engineering Mechanics-asce - J ENG MECH-ASCE*, 132, 02 2006.

[2] Andrew B. Kennedy, James T. Kirby, Qin Chen, and Robert A. Dalrymple. Boussinesq-type equations with improved nonlinear performance. *Wave Motion*, 33(3):225 – 243, 2001.

[3] Mara Tonelli and Marco Petti. Hybrid finite volume finite difference scheme for 2dh improved boussinesq equations. *Coastal Engineering*, 56(5):609 – 620, 2009.

[4] L.H. Lei, 2017.

[5] W. Craig, P. Guyenne, J. Hammack, D. Henderson, and C. Sulem. Solitary water wave interactions. *Physics of Fluids*, 18(5):057106, 2006.

[6] Fengyan Shi, James T. Kirby, Babak Tehranirad, Jeffrey C. Harris, Young-Kwang Choi, and Matt Malej. *Fully Nonlinear Boussinesq Wave Model with TVD Solver Documentation and User's Manual*, December 2016.

[7] Fengyan Shi, James T. Kirby, Jeffrey C. Harris, Joseph D. Geiman, and Stephan T. Grilli. A high-order adaptive time-stepping tvd solver for boussinesq modeling of breaking waves and coastal inundation. *Ocean Modelling*, 43-44:36 – 51, 2012.