XBeach: Non-hydrostatic model: Validation, verification and model description

Pieter Smit
Sofar Ocean Technologies
40 PUBLICATIONS **1,085** CITATIONS

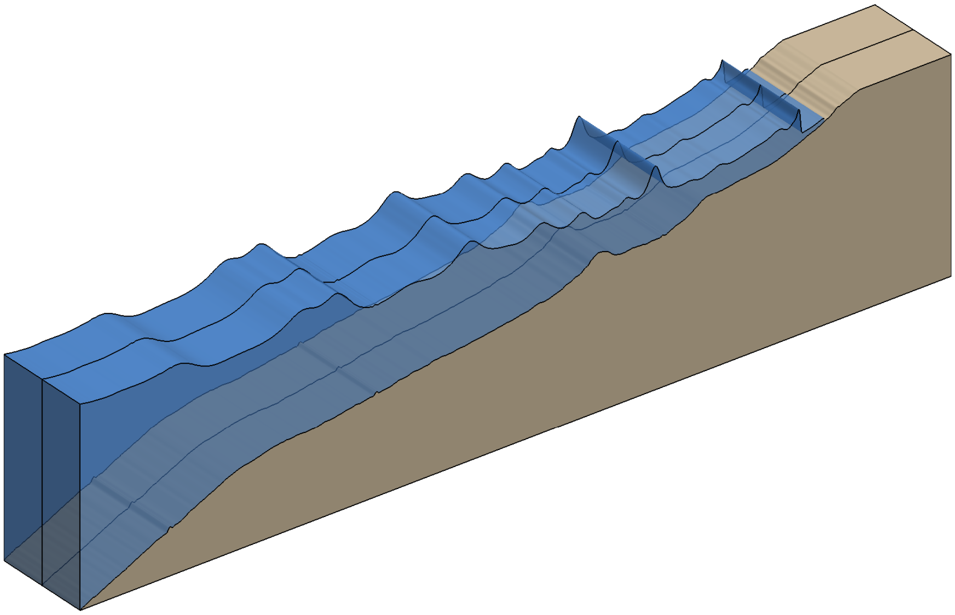
Guus S. Stelling
Delft University of Technology
95 PUBLICATIONS **6,396** CITATIONS

Dano J.A. Roelvink
IHE Delft Institute for Water Education
309 PUBLICATIONS **10,318** CITATIONS

Jaap Van Thiel de Vries
Delft University of Technology
56 PUBLICATIONS **2,637** CITATIONS

XBeach: Non-hydrostatic model

Validation, verification and model description



DRAFT

Delft University of Technology and Deltares

P.B. Smit, G.S.Stelling, D. Roelvink, J. van Thiel de Vries, R. McCall, A. van Dongeren C. Zwinkels, R. Jacobs

Abstract

This report describes the theoretical background, implementation and verification of a non-hydrostatical module in XBeach. XBeach is a two-dimensional model for wave propagation, long waves and mean flow, sediment transport and morphological changes of the near-shore area, beaches, dunes and back barrier during storms. It is a public funded domain model that has been developed with funding and support by the US Army core of engineers, a consortium of UNESCO-IHE, Deltares, Delft University of Technology and the University of Miami. The non-hydrostatic module extends XBeach`s capability to model non-linear waves, wave current interaction and wave breaking in the surf zone. Implementation was done by P.B. Smit of Delft University of Technology in cooperation with Deltares and was funded by Deltares.

The non-hydrostatic module is based upon Stelling and Zijlema (2003). Vertically a compact scheme is used which allows a very natural inclusion of the boundary condition of the dynamic pressure at the free surface. In this way dispersive waves can be modelled using a depth average flow model with similar accuracy to that of lower order Boussinesq models.

The application of momentum conservative numerical schemes allows the accurate modelling of wave breaking without the need of a separate breaking model. Second order accuracy in space and time has been achieved by the implementation of a flux limited variant of the scheme by MacCormack (1969).

Verification of linear dispersion and the balance between non-linearity and dispersion was done by comparison to analytical solutions for an oscillating basin and a solitary wave. Momentum conservation and the capability to capture shock waves were verified using the analytical solution for the dam break problem.

The model was validated using experimental results by Berkhoff et al. (1982) and Boers (2005). The elliptic shoal from the Berkhoff experiment was used to validate refraction and diffraction for monochromatic waves. The Boers case 1C was used to validate the propagation and breaking of irregular waves.

Results show that the model performs well when waves remain in relatively shallow water. This is as expected as the numerical dispersion relation for a depth averaged model only approximates the linear dispersion relation for relatively shallow water. The Boers experiment furthermore showed that initiation of the breaking process is well captured but the dissipation rate of wave energy is underestimated. This is probably due to an inaccurate balance between non-linearity and dispersion.

Table of contents

List of Main symbols

Roman symbols

Symbol	Description	Unit
c	Wave celerity	m/s
c_g	Group velocity	m/s
d	Depth, measured positive downward from z_0	m
d_0	Still water depth	m
g	Gravitational acceleration	
H	(i) Water depth	m
	(ii) Wave Height	m
		m/s^2
i	Mesh point index (subscript)	[-]
I	Total number of meshes in the X-direction	[-]
j	Mesh point index (subscript)	[-]
J	Total number of meshes in the Y-direction	[-]
k	(i) Mesh point index (subscript)	[-]
	(ii) Wave number	rad/m
k_x, k_y	Wave number components	rad/m
K	Total number of vertical points	[-]
L, L_x, L_y	Typical Flume/Basin/Area length	m
L_{sol}	Length scale for solitary waves	m
p	Normalized dynamic pressure	m^2/s^2
p_0	Atmospheric pressure	N/m^2
P	Pressure	N/m^2
${}^xq, {}^yq$	Specific discharge in the x-/y-direction	m^2/s
t	Time	s
T	Wave period	s
$\mathbf{u}$	Velocity vector	m/s
u	x-component of the velocity vector $\mathbf{u}$.	m/s
$\mathbf{U}$	Depth averaged velocity vector	m/s
U	x-component of the depth averaged velocity $\mathbf{U}$	m/s
$U_{i,j}$	Discrete velocity U	m/s
v	y-component of the velocity vector $\mathbf{u}$.	m/s
V	y-component of the depth averaged velocity $\mathbf{U}$	m/s
$V_{i,j}$	Discrete velocity V	m/s
w	z-component of the velocity vector $\mathbf{u}$.	m/s
W	Depth averaged vertical velocity	m/s

Symbol	Description	Unit
$W_{i,j,0}$	Vertical velocity at $z=-d(\mathbf{x})$	m/s
$W_{i,j,1}$	Vertical velocity at $z=-\eta(\mathbf{x},t)$	m/s
$\mathbf{x}$	Point at (x,y)	m
$\mathbf{X}$	Point at (x,y,z)	
$\mathbf{x}_{i,j}$	Point at (x_i,y_j)	
x	Principle horizontal coordinate	m
x_w	World coordinate	m
x_0	World x-coordinates of origin.	m
$x_i, x_{i+\frac{1}{2}}$	x-location of ith-gridline	m
y	Lateral horizontal coordinate	m
y_w	World coordinate	m
y_w	World y-coordinate of origin.	m
$y_j, y_{j+\frac{1}{2}}$	y-location of jth-gridline	m
z	Vertical coordinate	m
z_0	Reference level	m

Greek symbols

Symbol	Description	Unit
α	Underrelaxation coefficient in the SIP solver	[-]
Δ_s	the characteristic length scale of the smallest resolvable eddy.	m
Δx_i	Local mesh interval in x-direction	m
Δy_j	Local mesh interval in y-direction	m
δ	Relative accuracy, used as a stopping criteria in the iterative matrix solver	[-]
$\delta\Omega$	Rectangular shaped boundary curve of domain Ω	m
$\delta\Omega_{back}$	*"Back"* boundary, usually a land boundary.	m
$\delta\Omega_{front}$	*"Front"* boundary, usually the Seaward boundary.	m
$\delta\Omega_{left}, \delta\Omega_{right}$	"*Left/right*" boundaries of the domain.	m
ε	Measure of solitary wave non-linearity	[-]
η	Free surface elevation, measured positive upwards from z_0	m
λ	Wave length	m
ν_t	Eddy viscosity	m^2/s
τ_{ij}	Stress	m^2/s^2
Ω	Horizontal domain enclosed by boundary $\delta\Omega$	m^2

Acronyms/Abbreviations

Symbol	Description
NSWE	Non-linear shallow water equations
XBeach	Extreme Beach behaviour model
LES	Large eddy simulation
FOU	First order upwind
CFL	Courant-Friedrichs-Levy
SIP	Strongly implicit procedure
TDMA	Tri-diagonal matrix algorithm

1 Introduction

1.1 Motivation

The XBeach model by Roelvink et al. (2009) is a numerical model of near shore processes intended as a tool to assess the natural coastal response during time-varying storm and hurricane conditions, including dune erosion, overwash and breaching. The model consists of formulations for short wave envelope propagation, non-stationary shallow water equations, sediment transport and bed update.

XBeach solves the depth averaged non-linear shallow water (NLSW) equations cast in a Generalized Lagrangian Mean (GLM) formulation (Andrews and Mcintyre, 1978). These equations are forced by a time-dependent wave action balance similar to the 2nd generation spectral HISWA model Holthuijsen et al. (1989). The time dependant action balance is solved on the time-scale of wave groups. In this way the swash motion due to infragravity waves that are forced by the wave groups can be simulated.

An alternative and potentially more accurate approach for modelling the wave-current interaction is found in the phase resolving models. Examples of these are the Boussinesq (e.g. Chen et al., 2000) and non-hydrostatic models (e.g. Casulli and Stelling, 1998). These models resolve the wave field on the timescale of individual waves and are as such capable of modelling the non-linear evolution of the wave field accurately. For research purposes, adding the capability to model the wave field using a phase resolving model, would be an interesting extension to XBeach.

Traditionally the Boussinesq models, which have been designed specifically for wave propagation, were the more efficient models. Non-hydrostatic models needed a high resolution in the vertical (~ twenty layers) to obtain similar results to the depth averaged formulated Boussinesq models. The resulting difference in computational time resulted in a focus on the development of Boussinesq models for coastal engineering practice.

Recently Stelling and Zijlema (2003) showed that, using an edge based finite difference scheme in the vertical, it is possible to construct a non-hydrostatic model that is competitive to the Boussinesq models. For linear wave propagation their model gives similar results to higher order Boussinesq models while with only two computational layers and for a single layer the model is comparable to lower order Boussinesq models.
In Zijlema and Stelling (2008) the authors also showed that, when using momentum conservative numerical schemes as described in Stelling and Duinmeijer (2003), the effect of wave breaking can be captured accurately without the use of a breaking model. This is an advantage over Boussinesq models which generally need a separate breaking model to initiate the breaking process.

The absence of a separate breaking model was the main motivation to choose to implement the non-hydrostatic model into XBeach. However, because XBeach is a depth averaged model, only the single layer version was implemented. Experience already showed that first order schemes generally introduced to much damping to accurately model short wave propagation. This was the reason to simultaneously implement a second order accurate scheme into XBeach. To allow for the accurate prediction of shock waves and wave breaking a momentum conservative flux limited version of the scheme by MacCormack (1969) was chosen for this purpose.

Both the non-hydrostatic and the McCormack scheme could be formulated as corrections to the first order hydrostatic calculations already present in XBeach. This allows them to be implemented as subroutine calls that can be enabled when the user needs higher accuracy or wants to run a non-hydrostatic model.

1.2 Objective

The main objective of this study was the implementation of the non-hydrostatic model as described by Zijlema and Stelling (2008) into XBeach. A second order scheme based on a flux limited version of the McCormack scheme was also implemented to allow for the accurate modelling of short waves. The secondary objective was the validation and verification of the non-hydrostatic model using similar test cases as described in Smit (2008).

1.3 Readers Guide

This report starts with an overview of the theoretical background in **chapter 2**. Here the governing equations that are used as a starting point are briefly introduced. Furthermore it contains a short description of how wave breaking is incorporated into the model.

Chapter 3 describes the numerical implementation of the governing equations. Both the discretisation and solution strategy are explained in this chapter.

Chapter 4 presents the verification and validation of the model. Model results are compared to analytical cases and experimental data.

The implementation and usage of the model is described in **Chapter 5**. This chapter gives a brief introduction on how the non-hydrostatic model is incorporated into XBeach and a short guide on how to use the non-hydrostatic model.

2 Theoretical background

2.1 Governing equations

Coordinate system and domain

XBeach uses a global Cartesian coordinate system that is used to define the location and orientation of the local coordinate system. This local or computational coordinate system is defined with the primary axis pointing in the direction of the coast and the lateral coordinate pointing approximately in the parallel direction (Figure 2-1). The computational coordinate system is related to the global coordinate system by an anti-clockwise rotation α and translation of the origin.

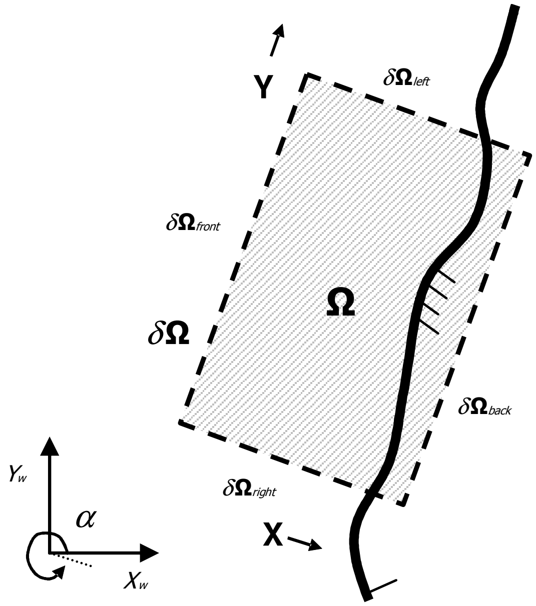

Y
$\delta\mathbf{\Omega}_{left}$
$\delta\mathbf{\Omega}_{front}$
$\mathbf{\Omega}$
$\delta\mathbf{\Omega}$
$\delta\mathbf{\Omega}_{back}$
Y_w
α
$\delta\mathbf{\Omega}_{right}$
X
X_w

Figure 2-1 Horizontal domain Ω .in relation to the global domain

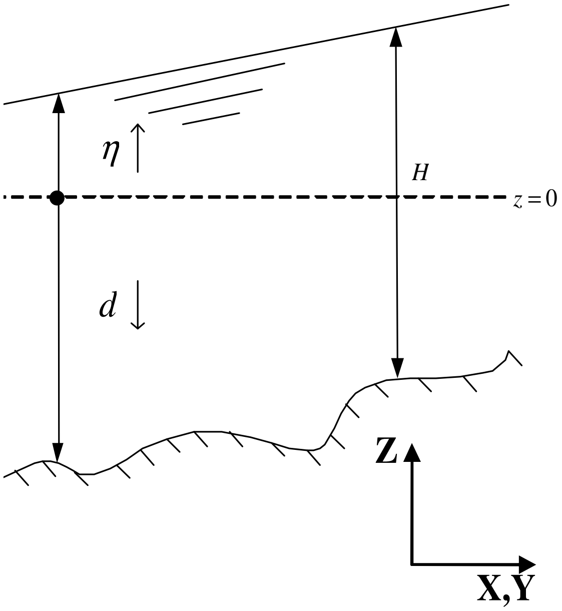


Figure 2-2 Vertical coordinate system.

The computational domain Ω is bound horizontally by a rectangular shaped boundary $\delta\Omega_h$ that consists of four vertical planes (Figure 2-1). The seaward boundary $\delta\Omega_{front}$ and the landward boundary $\delta\Omega_{back}$ are parallel to the *y*-axis, while the two side boundaries $\delta\Omega_{left}$ and $\delta\Omega_{right}$ are parallel to the *x*-axis. Vertically the domain is bound by the single valued free surface $z=\eta(\mathbf{x},t)$ (this excludes overtopping waves) and the bottom $z=-d(\mathbf{x},t)$ (see Figure 2-2).

Incompressible Navier-Stokes equations

Free surface flows in coastal areas are governed by the Navier Stokes equations. For the modelling of free surface gravity waves it suffices to restrict ourselves to: (i) Incompressible and (ii) Homogeneous (constant density, temperature, etc.) and (iii) Newtonian flow. In this case the Navier-stokes equations can be written in Cartesian coordinates as (Batchelor, 1967):

$$\frac{\partial \mathbf{u}}{\partial t}+\nabla\left(\mathbf{u}\otimes\mathbf{u}\right)=-\frac{1}{\rho}\nabla P+\mathbf{g}+\nabla\cdot\mathbb{T} \tag{1.1}$$

Where $\mathbf{u}$ = [$u(\mathbf{x},t),v(\mathbf{x},t),w(\mathbf{x},t)$] is the velocity vector, $P(\mathbf{x},t)$ the pressure, $\mathbf{g}$ = [$0,0,-g$] the gravitational body force and $\mathbb{T}$ the deviatoric viscous stress tensor. Furthermore $\mathbf{a}\otimes\mathbf{b}$ denotes the tensor product of the vectors **a** and **b**.

Usually the largest contribution to the pressure in a water column is due to the weight of the water above, or the hydrostatic pressure component. This can be made explicit in the equations by decomposing the pressure in a hydrostatic and hydrodynamic part or

$$P=\rho g\left(\eta-z\right)+\rho p+p_0 \tag{1.2}$$

Where p is the dynamic pressure normalized with the reference density ρ and p_0 is the atmospheric pressure, that is assumed to be uniform and stationary.

Furthermore the equations in (1.1) are averaged (over a suitable space or time scale) and the turbulent stress tensor is introduced to model the unresolved part of the flow. The turbulent stresses are then expressed in mean flow quantities using the Boussinesq hypothesis as

$$\tau_{ij}=\rho\nu_t\left(\frac{\partial\mathbf{u}_i}{\partial\mathbf{x}_j}+\frac{\partial\mathbf{u}_j}{\partial\mathbf{x}_i}\right) \tag{1.3}$$

Here the eddy viscosity ν_t is introduced that needs to be determined from the mean flow variables using a suitable closure model (e.g. a Smagorinsky sub-grid model or a $k-\varepsilon$ model). The resulting equations, in tensor notation, are

$$\frac{\partial \mathbf{u}}{\partial t}+\nabla\left(\mathbf{u}\otimes\mathbf{u}\right)=-\frac{1}{\rho}\nabla\left(p+\rho g\eta\right)+\nabla\cdot\boldsymbol{\tau} \tag{1.4}$$

Where $\boldsymbol{\tau}$ represents the turbulent shear stress tensor. The momentum equations are solved together with the conservation of mass. For an incompressible fluid the conservation of mass reduces to the conservation of volume

$$\nabla\cdot\mathbf{u}=0 \tag{1.5}$$

This equation will also be referred to as the (local) continuity equation. Because there is no separate evolution equation for the pressure the local continuity equation acts as a constraint on the flow field.

Because free surface flows are considered a separate equation to determine the free surface location η is required. To obtain an expression for the free surface the continuity equation is integrated over the depth:

$$\int_{-d}^{\zeta} \nabla \cdot \mathbf{u} dz = \int_{-d}^{\zeta} \frac{\partial u}{\partial x} dz + \int_{-d}^{\zeta} \frac{\partial v}{\partial y} dz + w\big|_{z=\zeta} - w\big|_{z=-d} \tag{1.6}$$

Now it is assumed that the water surface is always composed of the same particles. This is justified due to exclusion of wave overturning. The vertical velocity of a particle located at the free surface is therefore equal to the material derivative of the free surface and this results in the kinematic boundary condition at free surface:

$$w(x,y,\zeta,t) = \frac{D\zeta}{Dt} = \frac{\partial \zeta}{\partial t} + u\frac{\partial \zeta}{\partial x} + v\frac{\partial \zeta}{\partial y} \tag{1.7}$$

A similar condition is imposed at the bottom and this leads to the kinematic boundary condition at the bottom

$$w(x,y,-d,t) = \frac{\partial d}{\partial t} - u\frac{\partial d}{\partial x} - v\frac{\partial d}{\partial y} \tag{1.8}$$

Because we assume that the timescales at which the bed changes are much larger than the timescales of the fluid motion the time derivative in (1.8) is neglected. When the equations (1.7) and (1.8) are substituted into equation (1.6) and use is made of the Leibniz rule of integration the integrated continuity equation becomes

$$\frac{\partial \zeta}{\partial t} + \frac{\partial UH}{\partial x} + \frac{\partial VH}{\partial y} = 0 \tag{1.9}$$

Where U and V are the depth averaged velocities given by

$$U = \frac{1}{H}\int_{-d}^{\zeta} u dz, V = \frac{1}{H}\int_{-d}^{\zeta} v dz \tag{1.10}$$

From now on equation (1.9) is referred to as the global continuity equation. This equation gives a relationship between the depth averaged velocity and the surface elevation.

Depth averaged equations

XBeach is currently formulated using depth averaged quantities and therefore the momentum equations (1.4) are integrated over the water depth. Because the procedure is very similar for each of the three momentum equations only the *u*-momentum equation will be dealt with in detail.

In component form the conservation of momentum in the x-direction can be written as:

$$\frac{\partial u}{\partial t} + \underbrace{\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} + \frac{\partial uw}{\partial z}}_{\text{advective terms}} = \underbrace{-\frac{\partial p}{\partial x} - g\frac{\partial \eta}{\partial x}}_{\text{Pressure terms}} + \underbrace{\frac{1}{\rho}\frac{\partial \tau_{xx}}{\partial x} + \frac{1}{\rho}\frac{\partial \tau_{yx}}{\partial y} + \frac{1}{\rho}\frac{\partial \tau_{zx}}{\partial z}}_{\text{Turbulent stress terms}} \tag{1.11}$$

This equation is integrated over the depth and for clarity we consider the following contributions separately: (i) the time derivative, (ii) the advective terms, (iii) pressure terms and (iv) the stress terms. First integrating the time derivative over the depth which results in:

$$\int_{-d}^{\eta} \frac{\partial u}{\partial t} dz = \frac{\partial}{\partial t}(HU) - u\frac{\partial \eta}{\partial t}\bigg|_{z=\eta} + u\frac{\partial d}{\partial t}\bigg|_{z=-d} \tag{1.12}$$

The second and third terms on the right hand side are a result of the movement of the free surface and bottom. Integrating the advective terms leads to:

$$\begin{aligned}\int_{-d}^{\eta} dz\left(\frac{\partial u^2}{\partial x}+\frac{\partial uv}{\partial y}+\frac{\partial uw}{\partial z}\right)=\frac{\partial}{\partial x}\left(HU^2-\int_{-d}^{\eta}(u-U)^2\,dz\right)-\frac{\partial}{\partial y}\left(HUV-\int_{-d}^{\eta}(v-V)(u-U)\,dz\right)\\ -u\left[u\frac{\partial z}{\partial x}+v\frac{\partial z}{\partial y}+w\right]_{z=-d}^{z=\eta}\end{aligned}\tag{1.13}$$

The integrals on the right hand side of (1.13) are the dispersions terms that are due to the non-uniformity of the flow in the vertical. When the vertical distribution of the flow over the layer does not deviate significantly from the average velocity these integrals are small and common practice is to consider this dispersion effect as diffusion. For situations where the vertical profile of the flow is no longer uniform (e.g. undertow due to breaking waves) a higher resolution in the vertical will be required.
The final term in (1.13) is due to the application of Leibniz rule of integration and when combined with (1.12) these boundary terms cancel out.

Integrating the dynamic and hydrostatic pressure over the vertical results in:

$$\int_{-d}^{\eta} dz\left(\frac{\partial p}{\partial x}+g\frac{\partial \eta}{\partial x}\right)=\frac{\partial H\bar{p}}{\partial x}-\left[p\frac{\partial z}{\partial x}\right]_{z=-d}^{z=\eta}+gH\frac{\partial \eta}{\partial x}\tag{1.14}$$

Where $\bar{p}$ denotes the depth averaged dynamic pressure. Because the atmospheric pressure is set to zero the boundary term at the surface in (1.14) drops out.

Finally Integrating the stress terms results in

$$\frac{1}{\rho}\int_{-d}^{\eta} dz\left(\frac{\partial \tau_{xx}}{\partial x}+\frac{\partial \tau_{yx}}{\partial y}+\frac{\partial \tau_{zx}}{\partial z}\right)=\frac{1}{\rho}\frac{\partial}{\partial x}\left(H\bar{\tau}_{xx}\right)+\frac{1}{\rho}\frac{\partial}{\partial y}\left(H\bar{\tau}_{yx}\right)+\frac{\tau_{sx}}{\rho}-\frac{\tau_{bx}}{\rho}\tag{1.15}$$

Here $\bar{\tau}_{xx}$ and $\bar{\tau}_{xy}$ are the depth averaged (turbulent/viscous) stresses. Furthermore we defined the total stress at the bottom τ_{bx} (due to bottom friction) and free surface τ_{sx} (due to wind) as

$$\begin{aligned}\tau_{sx}&=+\tau_{xx}\frac{\partial \eta}{\partial x}+\tau_{zx}\quad \text{at}\quad z=\eta\\ \tau_{bx}&=-\tau_{xx}\frac{\partial d}{\partial x}+\tau_{zx}\quad \text{at}\quad z=-d\end{aligned}\tag{1.16}$$

Finally combining equations (1.12), (1.13), (1.14) and (1.15) results in the depth integrated horizontal momentum equation in the x-direction, written in conservative form

$$\frac{\partial}{\partial t}(HU)+\frac{\partial}{\partial x}\left(HU^2+\tfrac{1}{2}gH^2+H\bar{p}-\frac{1}{\rho}H\bar{\tau}_{xx}\right)+\frac{\partial}{\partial y}\left(HUV-\frac{1}{\rho}H\bar{\tau}_{yx}\right)=gH\frac{\partial d}{\partial x}-p\frac{\partial d}{\partial x}+S_x\tag{1.17}$$

Here the source/sink terms due to the stresses at the free surface and bottom are included in S_x .Using a similar derivation as for the x-direction the depth integrated equations in the y and z-direction read:

$$\frac{\partial}{\partial t}(HV)+\frac{\partial}{\partial x}\left(HUV-\frac{1}{\rho}H\bar{\tau}_{xy}\right)+\frac{\partial}{\partial y}\left(HV^2+\tfrac{1}{2}gH^2+H\bar{p}-\frac{1}{\rho}H\bar{\tau}_{yy}\right)=gH\frac{\partial d}{\partial y}-p\frac{\partial d}{\partial y}+S_y\tag{1.18}$$

$$\frac{\partial}{\partial t}(HW)+\frac{\partial}{\partial x}(HUW)+\frac{\partial}{\partial y}(HVW)=-p_{surface}+p_{bottom}+\frac{1}{\rho}\frac{\partial}{\partial x}(H\overline{\tau}_{xz})+\frac{1}{\rho}\frac{\partial}{\partial y}(H\overline{\tau}_{yz}) \tag{1.19}$$

The depth integrated equations are here presented in their conservative form and therefore appear different to those given in Roelvink, Reniers et al. (2009). However if the global continuity equation (1.9) is multiplied with U and subtracted from equation (1.17) it is easy to show that both forms are mathematically equivalent. The conservative form is preferred here as it is easier to derive the conservative numerical approximations from this form (see chapter 3).

Boundary conditions

Boundary conditions for the tangential and normal velocities/stresses need to be prescribed along the entire boundary, including bottom and free surface, to get a unique solution.

free surface and bottom

At the free surface the normal and tangential stresses are assumed to be continuous and the total pressure is set equal to the atmospheric pressure, therefore neglecting surface tension effects. Furthermore the atmospheric pressure at the free surface is assumed to be uniform and stationary and taken to be zero for convenience. This, combined with (1.2) leads to the boundary condition for the dynamic pressure:

$$p(\mathbf{x},\eta(\mathbf{x},t),t)=0 \tag{1.20}$$

The values of the tangential stresses at the free surface τ_{sx},τ_{sy} due to wind can be described using suitable expressions to capture the large scale influence of the wind. Notice that local generation of wind waves is not included.

At the bottom the kinematic boundary condition (1.8) is used to enforce the normal component of the flow. The two tangential stresses due to bottom friction are specified using an expression based on the mean flow of the form of

$$\tau_b \sim c_f \rho \mathbf{U}|\mathbf{U}| \tag{1.21}$$

where c_f is a dimensionless friction coefficient.

Closed boundaries

At closed boundaries the discharge through the boundary is assumed to be zero and the boundary is fully reflective. In this case the boundary represents a solid vertical wall. For such a situation the normal velocity component to the boundary is zero. For the tangential velocities free-slip conditions are applied and the tangential velocity gradient is set to zero.

Open boundaries

At the seaward and landward boundaries $\delta\Omega_{front}$, $\delta\Omega_{back}$ only the normal time dependent velocity component is prescribed whereas for the tangential velocities again the gradient is set to zero. The prescribed normal velocity is the superposition of both short and long wave signals, where the short wave signal is usually obtained using linear wave theory (e.g. Stelling and Zijlema, 2003). For situations where reflections play a significant role an absorbing-generating boundary condition based on the Sommerfeld radiation condition is used (see section 3.3). Alternatively a boundary condition based on van Dongeren and Svendsen (1997) can be used.

2.2 Wave breaking

Wave breaking is traditionally a difficult phenomenon to capture accurately. The free surface acts as an air-water interface which, due to for example wave overturning, can assume complex shapes. Furthermore, due the mixing between air and water in breaking waves the interface is sometimes hard to define. Computational methods that can handle these types of problems (Volume of Fluid methods, Marker and Cell) are numerically intensive and yield more information than is necessary for coastal engineering practice.

In the present model, as is typical for non-hydrostatic models (e.g. Casulli and Stelling, 1998), the free surface is tracked as a single valued function of the horizontal plane. This approach is more efficient and makes the simulation of wave transformations in the coastal zone feasible. However this does mean that breaking waves can no longer be captured in detail. Instead, wave breaking is regarded as a sub-grid process. Thus the waves are allowed to steepen until the front face is almost vertical, but then the detailed process of breaking (spilling / overturning) is not modelled.

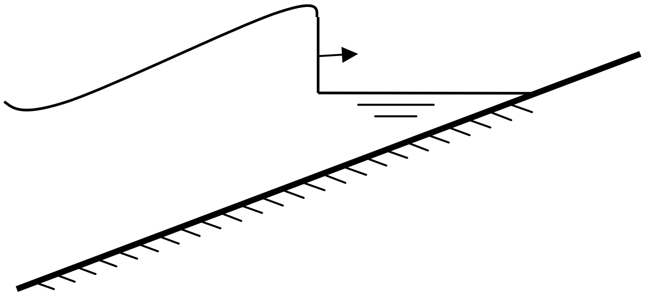


Figure 2-3 During breaking and run-up the wave is modelled as a bore

Such an approach towards wave breaking is not unique for non-hydrostatic models but has already been applied successfully in models based on the NSW-equations (e.g. Hibberd and Peregrine, 1979). In these models the analogy between a bore and a breaking wave is used to simulate wave evolution during breaking and run-up. This is justified because the breaking process itself appears to stabilize the wave form into a turbulent almost vertical front (Peregrine and Svendsen, 1978). Thus during breaking a permanent form long wave develops for which mass and momentum are conserved. Such a wave dissipates energy at the same rate as in a bore of similar height (Svendsen, 2006).

In the region just before the surf-zone the wave is still steepening and both frequency dispersion and non-linear effects are important. On the one hand the non-linear properties tend to steepen the wave front while frequency dispersion counteracts this steeping. Because frequency dispersion is absent in the NSW-equations the balancing effect of frequency dispersion is missing and the wave is transformed into a bore prematurely. This is the reason that the previously mentioned models based on the NSW-equations are only valid after breaking has been initiated and cannot be used to determine the breaking point accurately.

Non-hydrostatic models do not have this deficiency as they include frequency dispersion and are therefore applicable in the region prior to breaking. Furthermore, because they reduce to the NSW-equations for shallow water they can also be used after breaking has been initiated. It should be noted that it is important that the numerical schemes involved must threat shock propagation accurately to model broken waves in the surf zone.

The validity of this approach is illustrated in Zijlema and Stelling (2008) where the authors show that their non-hydrostatic model is capable of predicting the breakpoint accurately using a conservative scheme for mass and momentum. The most attractive feature of this approach is that there are no external parameters (such as a maximum steepness) which tell the model when breaking should be initiated.

This stands in contrast to the Boussinesq models that have been equally successful in modelling waves before breaking. In Boussinesq models the dispersive effects that stabilize the wave shape become so strong that they prevent the front from steepening further. For this reason wave breaking is always initiated using artificial empirical mechanism such as limiting the steepness. They also require additional dissipation model to achieve the correct amount of energy dissipation.

The model described in this report is based on an adapted version of the depth averaged non-hydrostatic model presented in Zijlema and Stelling (2008). Momentum and mass conservation are guaranteed using a conservative numerical method based on Stelling and Duinmeijer (2003). The model behaviour for wave breaking is therefore similar to their model. The largest differences are due to the assumption of depth averaged flow. Due to this assumption linear dispersion is modelled less accurately and this can lead to overestimation of wave energies in the high frequency range. The position of the breaking point appears to be unaffected but the amount of energy dissipation is underestimated and this may result in an overestimation of wave heights in the surf zone. Furthermore the vertical structure of the flow is not taken into account (undertow, roller) and this also gives underestimation of the total dissipation.

2.3 Subgrid turbulent mixing

In the surf zone large amounts of turbulence are generated due to wave breaking. To account for the exchange of momentum due to the unresolved scales on the grid a Smagorinsky-type sub grid model (Smagorinsky, 1963) is used. Formally such a sub-grid model is achieved after filtering the small scale motions out of the governing equations by filtering them over a volume of one or several grid-cells. The resulting equations apply to the large scale motions of the flow. In this context the sub-grid stress can be formulated as

$$\tau_{ij} = \nu_t \left(\frac{\partial \mathbf{u}_i}{\partial \mathbf{x}_j} + \frac{\partial \mathbf{u}_j}{\partial \mathbf{x}_i} \right) \tag{1.22}$$

Where τ_{ij} represents the stress and ν_t the eddy viscosity. This eddy viscosity is expressed in terms of flow properties on the resolved scale as

$$\nu_t = \left(C_s \Delta_s\right)^2 \sqrt{2\sum_{i=1}^{2}\sum_{j=1}^{2}\left(S_{ij}S_{ij}\right)} \quad \text{with} \quad S_{ij} = \tfrac{1}{2}\left(\frac{\partial \mathbf{u}_i}{\partial \mathbf{x}_j} + \frac{\partial \mathbf{u}_j}{\partial \mathbf{x}_i}\right) \tag{1.23}$$

Here C_s is the Smagorinsky constant (typically ~0.1-0.3) and Δ_s is the characteristic length scale of the smallest resolvable eddy. The characteristic length scale is essentially the filter width employed and is therefore dependent on the mesh-size. Here the following formulation is used

$$\Delta_s = \sqrt{\Delta x \Delta y} \tag{1.24}$$

This choice appears to be common in literature for depth averaged flows. It was for instance used in the Boussinesq model by Chen et al. (1999) for the modelling of a Rip Current system.

Because the magnitude of the eddy viscosity depends on the gradients in the velocity field the Smagorinsky sub-grid model adds very little dissipation in smooth regions of the flow. For wave propagation this is important as in the regions outside the surf zone the influence of (turbulent) viscosity is negligible. On the other hand in the surf zone where large gradients can develop extra dissipation is introduced. This characteristic of the model is considered beneficial, especially in its depth averaged form, as generally energy dissipation due to breaking is underestimated.

3 Numerical model

3.1 Grid schematization

Spatial grid

A cell with its centre at $\mathbf{x}_{i,j} = \left(x_i, y_i\right)$ is bounded by the horizontal grid lines $x_{i\pm\frac{1}{2},j}$ and $y_{i,j\pm\frac{1}{2}}$ and between the bottom $-d = -d\left(\mathbf{x}_{i,j}\right)$ and free surface $\eta_{i,j} = \eta\left(\mathbf{x}_{i,j}\right)$. The local mesh sizes $\Delta x_{i,j}$ $\Delta y_{i,j}$ and the local water dept $H_{i,j}$ are then described by

$$\Delta x_{i,j} = x_{i+\frac{1}{2},j} - x_{i-\frac{1}{2},j}, \quad \Delta y_{i,j} = y_{i,j+\frac{1}{2}} - y_{i,j-\frac{1}{2}}, \quad H_{i,j} = \eta_{i,j} + d_{i,j} \tag{2.1}$$

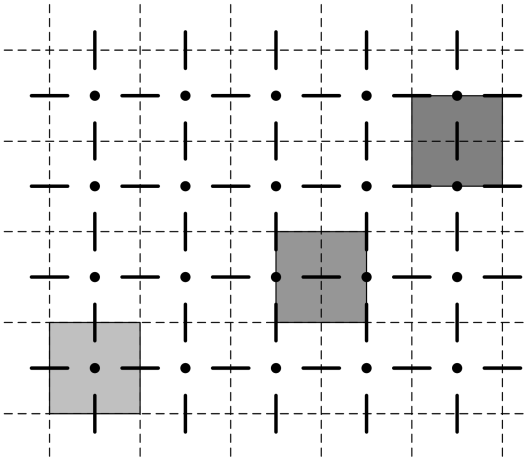


Figure 3-1 Horizontal location of variables.

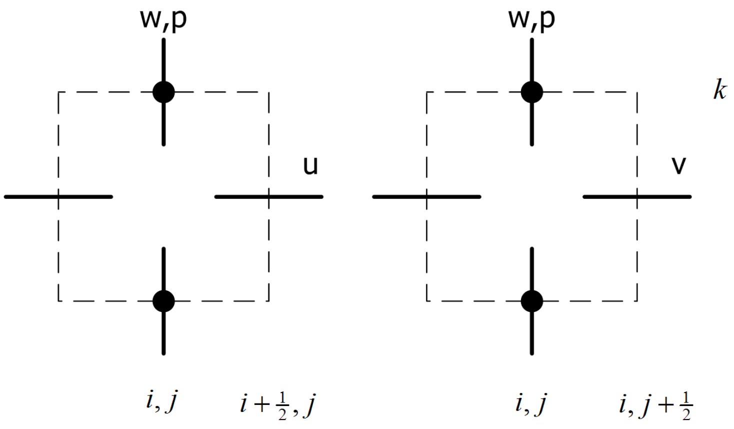


Figure 3-2 Vertical location of variables.

For the horizontal variable layout a staggered arrangement is employed. In the staggered arrangement the dynamic pressure, bottom and free surface variables are all located at the cell centre. The depth averaged horizontal velocity components *U*, V,on the other hand, are respectively located at the cell faces $\mathbf{x}_{i+\frac{1}{2},j}$ and $\mathbf{x}_{i,j+\frac{1}{2}}$.

To allow for the application of a compact scheme in the vertical the variable arrangement is not staggered. Instead, both the dynamic pressure and the vertical velocity component are located at the cell face (Figure 3-2). This allows for a very natural inclusion of the boundary condition of the dynamic pressure at the free surface. And it appears that a correct approximation of the pressure distribution in the top cell is key to modelling dispersive waves correctly (Stelling and Zijlema, 2003).

Note that the equations in (2.1) allow for non-uniform mesh sizes. Thus it is possible to increase the number of points locally in regions where strong variation in the flow are expected. This variable grid spacing is already optional in XBeach and has also been included into the non-hydrostatic extension.

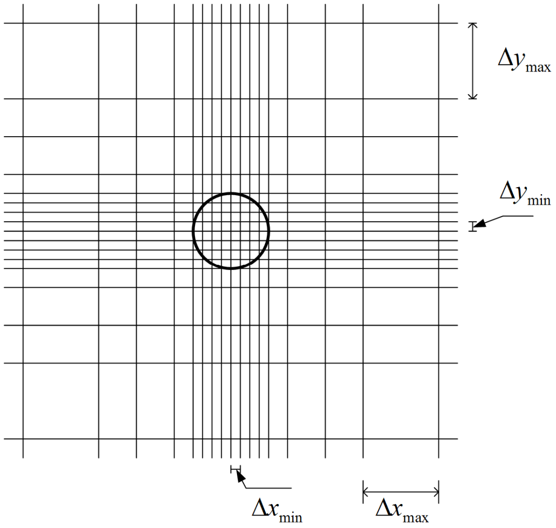


Figure 3-3 Example of a non-uniform structured grid

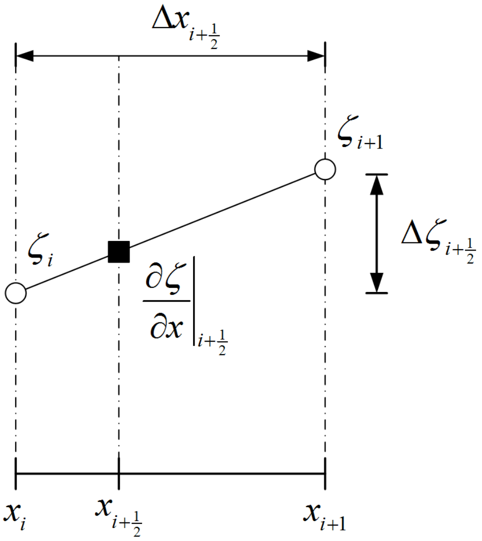

$\Delta x_{i+\frac{1}{2}}$
ζ_{i+1}
ζ_i
$\Delta\zeta_{i+\frac{1}{2}}$
$\left.\frac{\partial\zeta}{\partial x}\right|_{i+\frac{1}{2}}$
x_i
$x_{i+\frac{1}{2}}$
x_{i+1}

Figure 3-4 For non-uniform grids central approximation reduce locally to first order accuracy

A consequence of using non-uniform grids is that locally the central approximations reduce to first order accuracy (Hirsch, 2007). As long as the changes in the grid are not sudden the error made is still substantially smaller than those made in a forward or backward difference scheme. Smooth grid transitions can be achieved using an expansion parameter

$$\Delta x_{i+1} = r_i \Delta x_i \text{ and } \Delta y_{j+1} = r_j \Delta y_j \tag{2.2}$$

When the expansion (or contraction) parameter r is close to unity the first order truncation error made will indeed be small and the scheme is of almost second order. The use of a smoothly varying grid as described in equation (2.2) also creates a smooth transition between zones with a fine grid resolution and regions where a coarser grid is appropriate.

Temporal grid

For explicit schemes usually a constant time step is chosen beforehand which satisfies the CFL condition. This has the disadvantage that the worst case scenario determines the maximum time step which can be taken. During most of the simulation it is likely that a much larger time step is possible which means that the simulation is quite inefficient. To circumvent this problem the XBeach model uses a dynamically adjusted time step. The user supplies a value for the CFL condition beforehand and the program dynamically adjust the time step taken to adhere to this condition using the most up to date system state. In this way the largest possible time step is taken and more efficient time integration is the result.

3.2 Discretisation

Global continuity equation

As was outlined in the previous chapter the global continuity equation, which describes the relation between the free surface and the depth averaged discharge, is given by

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}(UH) + \frac{\partial}{\partial x}(VH) = 0 \tag{2.3}$$

A simple semi-discretisation of (2.3) using central differences for the space derivative and using the Hansen scheme for the coupling between velocity and free surface results in

$$\frac{\eta^{*}_{i,j} - \eta^{n}_{i,j}}{\Delta t} + \frac{{}^{x}q^{*}_{i+\frac{1}{2},j} - {}^{x}q^{*}_{i-\frac{1}{2},j}}{\Delta x} + \frac{{}^{y}q^{*}_{i,j+\frac{1}{2}} - {}^{y}q^{*}_{i,j-\frac{1}{2}}}{\Delta y} = 0 \tag{2.4}$$

With ${}^{x}q^{*}_{i+\frac{1}{2},j} = H^{n}_{i+\frac{1}{2},j}U^{n+\frac{1}{2}}_{i+\frac{1}{2},j}$, ${}^{y}q^{*}_{i,j+\frac{1}{2}} = H^{n}_{i,j+\frac{1}{2}}V^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}$ and the water depth is defined by a first order accurate upwind interpolation

$$H^{n}_{i+\frac{1}{2},j} = \begin{cases} \zeta^{n}_{i,j} + d_{i,j} & \text{if } \mathrm{U}^{\mathrm{n}+\frac{1}{2}}_{\mathrm{i}+\frac{1}{2},j} > 0 \\ \zeta^{n}_{i+1,j} + d_{i+1,j} & \text{if } \mathrm{U}^{\mathrm{n}+\frac{1}{2}}_{\mathrm{i}+\frac{1}{2},j} < 0 \\ \max\left(\zeta^{n}_{i,j}, \zeta^{n}_{i+1,j}\right) + \min\left(d_{i}, d_{i+1,j}\right) & \text{if } \mathrm{U}^{\mathrm{n}+\frac{1}{2}}_{\mathrm{i}+\frac{1}{2},j} = 0 \end{cases} \tag{2.5}$$

The resulting scheme is only first order accurate by virtue of the upwind interpolations and mass conservative. When first order computations are considered accurate enough $\eta^{n+1}_{i,j}$ is set to $\eta^{n*}_{i,j}$. For higher order accuracy the first order prediction is corrected using a limited version of the McCormack scheme. The corrector step reads

$$\frac{\eta^{n+1}_{i,j} - \eta^{*}_{i,j}}{\Delta t} + \frac{{}^{x}\Delta q^{*}_{i+\frac{1}{2},j} - {}^{x}\Delta q^{*}_{i-\frac{1}{2},j}}{\Delta x} + \frac{{}^{y}\Delta q^{*}_{i,j+\frac{1}{2}} - {}^{y}\Delta q^{*}_{i,j-\frac{1}{2}}}{\Delta y} = 0 \tag{2.6}$$

With ${}^{x}\Delta q^{n*}_{i+\frac{1}{2},j} = U^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\Delta H_{i+\frac{1}{2},j}$ and $\Delta H_{i+\frac{1}{2},j}$ is given for positive flow as

$$\Delta H^{n*}_{i+\frac{1}{2},j} = \tfrac{1}{2}\psi\left({}^{\zeta}r_{i+\frac{1}{2}}\right)\left(\zeta^{n*}_{i+1,j} - \zeta^{n}_{i,j}\right) \qquad {}^{\zeta}\mathrm{r}_{\mathrm{i}+\frac{1}{2}} = \frac{\zeta^{n*}_{i,j} - \zeta^{n}_{i-1,j}}{\zeta^{n*}_{i+1,j} - \zeta^{n}_{i,j}} \qquad \psi(r) = \max\left(0, \min(r,1)\right) \tag{2.7}$$

Here $\psi(r)$ denotes the minmod limiter. Similar expression can be constructed for negative flow. The expression for ${}^{y}\Delta q^{n*}_{i,j+\frac{1}{2}}$ and $\Delta H_{i,j+\frac{1}{2}}$ are obtained in a similar manner. Note that the total flux ${}^{x}q^{n+\frac{1}{2}}_{i+\frac{1}{2},j}$ at the cell boundaries thus reads

$${}^{x}q^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = {}^{x}q^{*}_{i+\frac{1}{2},j} + {}^{x}\Delta q^{*}_{i+\frac{1}{2},j}, \qquad {}^{y}q^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} = {}^{y}q^{*}_{i,j+\frac{1}{2}} + {}^{y}\Delta q^{*}_{i,j+\frac{1}{2}} \tag{2.8}$$

The predictor-corrector set is second order accurate in regions where the solution is smooth, and reduces locally to first order accuracy near discontinuities. Furthermore, the method remains mass conservative. Note that other flux limiters can be used instead of the minmod limiter. However, as the minmod limiter performed adequately, this has not been investigated. (For an overview of flux limiters see Hirsch, 2007)

Local continuity equation

The depth averaged local continuity equation is given by

$$\frac{\partial HU}{\partial x} + \frac{\partial HV}{\partial y} + w\Big|_{z=\eta} - u\frac{\partial z}{\partial x}\Big|_{z=-\zeta} - v\frac{\partial z}{\partial y}\Big|_{z=-\zeta} = 0 \tag{2.9}$$

This equation is discretized using central differences

$$\frac{H^{n+1}_{i+\frac{1}{2},j}U^{n+1\frac{1}{2}}_{i+\frac{1}{2},j} - H^{n+1}_{i-\frac{1}{2},j}U^{n+1\frac{1}{2}}_{i-\frac{1}{2},j}}{\Delta x} + \frac{H^{n+1}_{i+\frac{1}{2},j}V^{n+1\frac{1}{2}}_{i,j+\frac{1}{2}} - H^{n+1}_{i,j-\frac{1}{2}}V^{n+1\frac{1}{2}}_{i,j-\frac{1}{2}}}{\Delta y} + w^{n+1\frac{1}{2}}_{i,j,s} - \bar{U}^{n+1\frac{1}{2}}_{i,j}\frac{\eta^{n+1}_{i+\frac{1}{2},j} - \eta^{n+1}_{i-\frac{1}{2},j}}{\Delta x} - \bar{V}^{n+1\frac{1}{2}}_{i,j}\frac{\eta^{n+1}_{i,j+\frac{1}{2}} - \eta^{n+1}_{i,j-\frac{1}{2}}}{\Delta x} = 0 \tag{2.10}$$

Missing grid variables $\eta^{n+1}_{i+\frac{1}{2},j}, \eta^{n+1}_{i,j+\frac{1}{2}}$ are approximated with upwind interpolation. Because there is no separate time evolution equation for the pressure the local continuity equation will be used to setup a discrete set of poison type equations in which the pressures are the only unknown quantities.

Horizontal Momentum

To obtain a conservative discretisation of the momentum equation the approach from Stelling and Duinmeijer (2003) is followed. However, to improve the accuracy of the method the combined space-time discretisation of the advection is done using a variant of the MacCormack (1969) is used. This scheme consists of a first order predictor step and a flux limited corrector step. The hydrostatic pressure is integrated using the midpoint rule and central differences, while the source terms and the turbulent stresses are integrated using an explicit Euler time integration. Formally the time integration is therefore first order accurate, but in regions where the turbulent stresses are negligible the scheme is of almost second order accuracy.

Predictor step

The depth averaged horizontal momentum equation for *HU* is given by

$$\frac{\partial}{\partial t}\left(HU\right)+\frac{\partial}{\partial x}\left(HU^2+\tfrac{1}{2}gH^2+H\overline{p}-\tau_{xx}\right)+\frac{\partial}{\partial y}\left(HUV-\tau_{yx}\right)=gH\frac{\partial d}{\partial x}-p\frac{\partial d}{\partial x}+S_x \tag{2.11}$$

A first order accurate predictor step in time and space is then given as

$$\begin{aligned}&\frac{\left(HU\right)^{*}_{i+\frac{1}{2},j}-\left(HU\right)^{n+\frac{1}{2}}_{i+\frac{1}{2},j}}{\Delta t}+\frac{{}^{x}\overline{q}^{n+\frac{1}{2}}_{i+1,j}U^{n+\frac{1}{2}}_{i+1,j}-{}^{x}\overline{q}^{n+\frac{1}{2}}_{i,j}U^{n+\frac{1}{2}}_{i,j}}{\Delta x}+\frac{{}^{y}\overline{q}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}}U^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}}-{}^{y}\overline{q}^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2}}U^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2}}}{\Delta y}\\&+g\frac{\left(H^2\right)^{n+1}_{i+1,j}-\left(H^2\right)^{n+1}_{i,j}}{2\Delta x}=g\overline{H}^{n}_{i+\frac{1}{2},j}\frac{d_{i+\frac{1}{2},j}-d_{i-\frac{1}{2},j}}{\Delta x}+{}^{x}\mathrm{Pr}^{n+\frac{1}{2}}_{i+\frac{1}{2},j}+{}^{x}\mathrm{S}^{n+\frac{1}{2}}_{i+\frac{1}{2},j}+{}^{x}\mathrm{T}^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\end{aligned} \tag{2.12}$$

Here Pr represents a discretisation of the dynamic pressure; T the effect of (turbulent) viscosity and S includes all other source terms. The discretisation of the (turbulent) viscous terms is given by central differences:

$$\begin{aligned}{}^{x}T^{n+\frac{1}{2}}_{i+\frac{1}{2},j}=&\frac{2}{\Delta x_{i+\frac{1}{2}}}\left[\nu^{n}_{i+1,j}H^{n+1}_{i+1,j}\frac{U^{n+\frac{1}{2}}_{i+1\frac{1}{2},j}-U^{n+\frac{1}{2}}_{i+\frac{1}{2},j}}{\Delta x_{i+1}}-\nu^{n}_{i,j}H^{n+1}_{i,j}\frac{U^{n+\frac{1}{2}}_{i+\frac{1}{2},j}-U^{n+\frac{1}{2}}_{i-\frac{1}{2},j}}{\Delta x_{i}}\right]\\&+\frac{1}{\Delta y_i}\left[\overline{\overline{\nu}}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}}\overline{\overline{H}}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}}\frac{U^{n+1}_{i+\frac{1}{2},j+1}-U^{n+\frac{1}{2}}_{i+\frac{1}{2},j}}{\Delta y_{i+\frac{1}{2}}}-\overline{\overline{\nu}}^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2}}\overline{\overline{H}}^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}}\frac{U^{n+\frac{1}{2}}_{i+\frac{1}{2},j}-U^{n+\frac{1}{2}}_{i+\frac{1}{2},j-1}}{\Delta y_{i-\frac{1}{2}}}\right]\\&+\frac{1}{\Delta y_i}\left[\overline{\overline{\nu}}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}}\overline{\overline{H}}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}}\frac{V^{n+1}_{i+1,j+\frac{1}{2}}-V^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2}}}-\overline{\overline{\nu}}^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2}}\overline{\overline{H}}^{n+1}_{i+\frac{1}{2},j-\frac{1}{2}}\frac{V^{n+\frac{1}{2}}_{i+1,j-\frac{1}{2}}-V^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}}{\Delta x_{i+\frac{1}{2}}}\right]\end{aligned} \tag{2.13}$$

Here $\overline{\overline{\nu}}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+1}$ and $\overline{\overline{H}}^{n+\frac{1}{2}}_{i+\frac{1}{2},j+1\frac{1}{2}}$ are obtained from the surrounding points by simple linear interpolation.

Due to the incompressible flow assumption the dynamic pressure does not have a separate time evolution equation, but instead it satisfies an elliptical equation in space. As such its effect cannot be calculated explicitly using values at the previous time level. However to improve the accuracy of the predictor step the effect of the dynamic pressure is included explicitly. To do this first the unknown pressure is decomposed as:

$$p_{i,j}^{n+1\frac{1}{2}} = p_{i,j}^{n+\frac{1}{2}} + \Delta p_{i,j}^{n+1\frac{1}{2}} \tag{2.14}$$

where the difference in pressure $\Delta p_{i,j}^{n+1\frac{1}{2}}$ is generally small. In the predictor step the effect of the pressure is included explicitly using $p_{i,j}^{n+\frac{1}{2}}$. In the corrector step the full Poisson equation is then solved for $\Delta p_{i,j}^{n+1\frac{1}{2}}$. The pressure term in the predictor step is thus given as

$${}^{x}\mathrm{Pr}_{i+\frac{1}{2},j}^{n+\frac{1}{2}} = \frac{H_{i+1,j}^{n+1}\bar{p}_{i+1,j}^{n+\frac{1}{2}} - H_{i,j}^{n+1}\bar{p}_{i,j}^{n+\frac{1}{2}}}{\Delta x} - p_{i+\frac{1}{2},j}^{n+\frac{1}{2}}\frac{d_{i+\frac{1}{2},j} - d_{i-\frac{1}{2},j}}{\Delta x} = \frac{\left(\eta_{i+1,j}^{n+1} + d_{i,j}^{n+1}\right)p_{i+1,j}^{n+\frac{1}{2}} - \left(\eta_{i,j}^{n+1} - d_{i+1,j}^{n+1}\right)p_{i,j}^{n+\frac{1}{2}}}{2\Delta x} \tag{2.15}$$

Here $\bar{p}_{i+1,j}^{n+\frac{1}{2}}$ represents the average pressure over the vertical which is approximated with $\bar{p}_{i+1,j}^{n+\frac{1}{2}} = \frac{1}{2}p_{i+1,j}^{n+\frac{1}{2}}$, in which $p_{i+1,j}^{n+\frac{1}{2}}$ is the pressure at the bottom. Furthermore $p_{i+\frac{1}{2},j}^{n+\frac{1}{2}}$ is given as $p_{i+\frac{1}{2},j}^{n+\frac{1}{2}} = \frac{1}{2}\left(p_{i+1,j}^{n+\frac{1}{2}} + p_{i,j}^{n+\frac{1}{2}}\right)$.

Currently (2.12) is formulated with the depth integrated momentum as the primitive variable, and not the depth averaged velocity. To reformulate (2.12) in terms of U we use the method by Stelling and Duinmeijer (2003). First note that $\left(HU\right)_{i+\frac{1}{2},j}^{n+\frac{1}{2}}$ and $\left(HU\right)_{i+\frac{1}{2},j}^{*}$ are approximated as $\bar{H}_{i+\frac{1}{2},j}^{n}U_{i+\frac{1}{2},j}^{n+\frac{1}{2}}$ and $\bar{H}_{i+\frac{1}{2},j}^{n+1}U_{i+\frac{1}{2},j}^{*}$. Now using (2.4) $\left(HU\right)_{i+\frac{1}{2},j}^{n+\frac{1}{2}}$ is equivalent to:

$$\left(HU\right)_{i+\frac{1}{2},j}^{n+\frac{1}{2}} = \bar{H}_{i+\frac{1}{2},j}^{n+1}U_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - U_{i+\frac{1}{2},j}^{n+\frac{1}{2}}\Delta t\frac{{}^{x}\bar{q}_{i+1,j}^{n+\frac{1}{2}} - {}^{x}\bar{q}_{i,j}^{n+\frac{1}{2}}}{\Delta x} - U_{i+\frac{1}{2},j}^{n+\frac{1}{2}}\Delta t\frac{{}^{y}\bar{q}_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} - {}^{y}\bar{q}_{i+\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta y} \tag{2.16}$$

And

$$\bar{H}_{i+\frac{1}{2},j}^{n+1} = \frac{1}{2}\left(H_{i+1,j}^{n+1} + H_{i,j}^{n+1}\right), \quad {}^{x}\bar{q}_{i,j}^{n+\frac{1}{2}} = \frac{1}{2}\left({}^{x}q_{i+\frac{1}{2},j}^{n+\frac{1}{2}} + {}^{x}q_{i-\frac{1}{2},j}^{n+\frac{1}{2}}\right), \quad {}^{y}\bar{q}_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}\left({}^{y}q_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}} + {}^{y}q_{i+\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}}\right)$$

Substituting (2.16) into (2.12) the full expressions (including those for $V_{i,j+\frac{1}{2}}^{*}$) become:

$$\begin{aligned}
&\frac{U_{i+\frac{1}{2},j}^{*} - U_{i+\frac{1}{2},j}^{n+\frac{1}{2}}}{\Delta t} + \frac{{}^{x}\bar{q}_{i+1,j}^{n+\frac{1}{2}}U_{i+1,j}^{n+\frac{1}{2}} - {}^{x}\bar{q}_{i,j}^{n+\frac{1}{2}}U_{i,j}^{n+\frac{1}{2}}}{\bar{H}_{i+\frac{1}{2},j}^{n+1}\Delta x} + \frac{{}^{y}\bar{q}_{i,j+1}^{n+\frac{1}{2}}U_{i,j+1}^{n+\frac{1}{2}} - {}^{y}\bar{q}_{i,j}^{n+\frac{1}{2}}U_{i,j}^{n+\frac{1}{2}}}{\bar{H}_{i+\frac{1}{2},j}^{n+1}\Delta y} - \frac{U_{i+\frac{1}{2},j}^{n+\frac{1}{2}}}{\bar{H}_{i+\frac{1}{2},j}^{n+1}}\frac{{}^{x}\bar{q}_{i+1,j}^{n+\frac{1}{2}} - {}^{x}\bar{q}_{i,j}^{n+\frac{1}{2}}}{\Delta x} \\
&\qquad - \frac{U_{i+\frac{1}{2},j}^{n+\frac{1}{2}}}{\bar{H}_{i+\frac{1}{2},j}^{n+1}}\frac{{}^{y}\bar{q}_{i,j+1}^{n+\frac{1}{2}} - {}^{y}\bar{q}_{i,j}^{n+\frac{1}{2}}}{\Delta y} + g\frac{\eta_{i+1,j}^{n+1} - \eta_{i,j}^{n+1}}{\Delta x} = \frac{{}^{x}\mathrm{Pr}_{i+\frac{1}{2},j}^{n+\frac{1}{2}} + {}^{x}\mathrm{S}_{i+\frac{1}{2},j}^{n+\frac{1}{2}} + {}^{x}\mathrm{T}_{i+\frac{1}{2},j}^{n+\frac{1}{2}}}{\bar{H}_{i+\frac{1}{2},j}^{n+1}} \\
&\frac{V_{i,j+\frac{1}{2}}^{*} - V_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta t} + \frac{{}^{y}\bar{q}_{i,j+1}^{n+\frac{1}{2}}V_{i,j+1}^{n+\frac{1}{2}} - {}^{y}\bar{q}_{i,j}^{n+\frac{1}{2}}V_{i,j}^{n+\frac{1}{2}}}{\bar{H}_{i,j+\frac{1}{2}}^{n+1}\Delta y} + \frac{{}^{x}\bar{q}_{i+1,j}^{n+\frac{1}{2}}V_{i+1,j}^{n+\frac{1}{2}} - {}^{x}\bar{q}_{i,j}^{n+\frac{1}{2}}V_{i,j}^{n+\frac{1}{2}}}{\bar{H}_{i,j+\frac{1}{2}}^{n+1}\Delta x} - \frac{V_{i+\frac{1}{2},j}^{n+\frac{1}{2}}}{\bar{H}_{i,j+\frac{1}{2}}^{n+1}}\frac{{}^{y}\bar{q}_{i,j+1}^{n+\frac{1}{2}} - {}^{y}\bar{q}_{i,j}^{n+\frac{1}{2}}}{\Delta y} \\
&\qquad \frac{V_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}}{\bar{H}_{i,j+\frac{1}{2}}^{n+1}}\frac{{}^{x}\bar{q}_{i+1,j}^{n+\frac{1}{2}} - {}^{x}\bar{q}_{i,j}^{n+\frac{1}{2}}}{\Delta x} + g\frac{\eta_{i,j+1}^{n+1} - \eta_{i,j}^{n+1}}{\Delta x} = \frac{{}^{y}\mathrm{Pr}_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} + {}^{y}\mathrm{S}_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} + {}^{y}\mathrm{T}_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}}{\bar{H}_{i,j+\frac{1}{2}}^{n+1}}
\end{aligned} \tag{2.17}$$

Where we again use a first order upwind interpolation for $U_{i+1,j}^{n+\frac{1}{2}}$ and $U_{i,j+1}^{n+\frac{1}{2}}$. This is exactly the approximation used by Stelling and Duinmeijer (2003) and is fully momentum conservative.

Corrector step

The predictor step (2.17) is first order accurate in both space and time due to the use of upwind approximations for and Euler explicit time integration for the advective terms, and first order time integration for the source/viscous terms. This level of accuracy is acceptable near shore, where strong non-linearity (wave breaking, flooding and drying) will force the use of small steps in space and time anyway. However, in the region where waves only slowly change (e.g. shoaling/refraction on mild slopes), the first order approximations suffer from significant numerical damping. To improve the accuracy of the numerical model in these regions a corrector step is implemented after the predictor step.

The corrector step is given by:

$$\frac{(HU)_{i+\frac{1}{2},j}^{n+1\frac{1}{2}} - (HU)_{i+\frac{1}{2},j}^{n*}}{\Delta t} + \frac{{}^{x}\overline{q}_{i+1,j}^{n+\frac{1}{2}}\Delta U_{i+1,j} - {}^{x}\overline{q}_{i,j}^{n+\frac{1}{2}}\Delta U_{i,j}}{\Delta x} + \frac{{}^{y}\overline{q}_{i,j+1}^{n+\frac{1}{2}}\Delta U_{i,j+1} - {}^{y}\overline{q}_{i,j}^{n+\frac{1}{2}}\Delta U_{i,j}}{\Delta y} + \\ \frac{\left(\eta_{i+1,j}^{n+1} + d_{i,j}^{n+1}\right)\Delta p_{i+1,j}^{n+1\frac{1}{2}} - \left(\eta_{i,j}^{n+1} - d_{i+1,j}^{n+1}\right)\Delta p_{i,j}^{n+1\frac{1}{2}}}{2\Delta x} = 0 \tag{2.18}$$

Or, when formulated in terms of the depth averaged velocity

$$\frac{U_{i+\frac{1}{2},j}^{n+1\frac{1}{2}} - U_{i+\frac{1}{2},j}^{n*}}{\Delta t} + \frac{{}^{x}\overline{q}_{i+1,j}^{n+\frac{1}{2}}\Delta U_{i+1,j} - {}^{x}\overline{q}_{i,j}^{n+\frac{1}{2}}\Delta U_{i,j}^{n*}}{\overline{H}_{i+\frac{1}{2},j}^{n+1}\Delta x} + \frac{{}^{y}\overline{q}_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}}\Delta U_{i+\frac{1}{2},j+\frac{1}{2}} - {}^{y}\overline{q}_{i+\frac{1}{2},j-\frac{1}{2}}^{n+\frac{1}{2}}\Delta U_{i+\frac{1}{2},j-\frac{1}{2}}}{\overline{H}_{i+\frac{1}{2},j}^{n+1}\Delta y} + \ldots \\ \ldots + \frac{\left(\eta_{i+1,j}^{n+1} + d_{i,j}^{n+1}\right)\Delta p_{i+1,j}^{n+1\frac{1}{2}} - \left(\eta_{i,j}^{n+1} - d_{i+1,j}^{n+1}\right)\Delta p_{i,j}^{n+1\frac{1}{2}}}{2\overline{H}_{i+\frac{1}{2},j}^{n+1}\Delta x} = 0 \\ \frac{V_{i,j+\frac{1}{2}}^{n+1\frac{1}{2}} - V_{i,j+\frac{1}{2}}^{n*}}{\Delta t} + \frac{{}^{y}\overline{q}_{i,j+1}^{n+\frac{1}{2}}\Delta V_{i,j+1} - {}^{y}\overline{q}_{i,j}^{n+\frac{1}{2}}\Delta V_{i,j}}{\overline{H}_{i,j+\frac{1}{2}}^{n+1}\Delta y} + \frac{{}^{x}\overline{q}_{i+\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}}\Delta V_{i+\frac{1}{2},j+\frac{1}{2}} - {}^{y}\overline{q}_{i-\frac{1}{2},j+\frac{1}{2}}^{n+\frac{1}{2}}\Delta V_{i-\frac{1}{2},j+\frac{1}{2}}}{\overline{H}_{i,j+\frac{1}{2}}^{n+1}\Delta y} + \ldots \\ \ldots + \frac{\left(\eta_{i,j+1}^{n+1} + d_{i,j}^{n+1}\right)\Delta p_{i,j+1}^{n+1\frac{1}{2}} - \left(\eta_{i,j}^{n+1} - d_{i,j+1}^{n+1}\right)\Delta p_{i,j}^{n+1\frac{1}{2}}}{2\overline{H}_{i,j+\frac{1}{2}}^{n+1}\Delta x} = 0 \tag{2.19}$$

The values of $\Delta U_{i+1,j}^{n*}$ are obtained from slope limited expressions. For positive flow these read:

$$\Delta U_{i,j}^{n*} = \tfrac{1}{2}\psi\left({}^{u}\mathrm{r}_{\mathrm{i}+\frac{1}{2}}\right)\left(U_{i+\frac{1}{2},j}^{*} - U_{i-\frac{1}{2},j}^{n}\right) \qquad {}^{u}\mathrm{r}_{\mathrm{i}+\frac{1}{2}} = \frac{U_{i-\frac{1}{2},j}^{*} - U_{i-1\frac{1}{2},j}^{n}}{U_{i+\frac{1}{2},j}^{*} - U_{i-\frac{1}{2},j}^{n}} \qquad \text{if } \mathrm{q}_{\mathrm{i},j}^{\mathrm{n}+\frac{1}{2}} > 0 \tag{2.20}$$

Where ψ again denotes the minmod limiter. Similar expressions can be constructed for $\Delta U_{i+\frac{1}{2},j+\frac{1}{2}}$, $\Delta V_{i,j}$ and $\Delta V_{i+\frac{1}{2},j+\frac{1}{2}}$.

The predictor-corrector set is second order accurate in regions where the solution is smooth, and reduces to first order accuracy near sharp gradients in the solutions to avoid unwanted oscillations. Furthermore, the method remains momentum conservative.

Vertical momentum equations

The vertical momentum equation (1.19) is discretized in a similar manner to the horizontal momentum equations using the McCormack scheme. In terms of the depth averaged vertical velocity the predictor step is:

$$\frac{\overline{W}_{i,j}^{*} - \overline{W}_{i,j}^{n+\frac{1}{2}}}{\Delta t} + \frac{{}^{x}q_{i+\frac{1}{2},j}^{n+\frac{1}{2}}\overline{W}_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - {}^{x}q_{i-\frac{1}{2},j}^{n+\frac{1}{2}}\overline{W}_{i-\frac{1}{2},j}^{n+\frac{1}{2}}}{H_{i,j}^{n+1}\Delta x} - \frac{\overline{W}_{i,j}^{n+\frac{1}{2}}}{H_{i,j}^{n+1}}\frac{{}^{x}q_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - {}^{x}q_{i-\frac{1}{2},j}^{n+\frac{1}{2}}}{\Delta x} + \frac{{}^{y}q_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}\overline{W}_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - {}^{y}q_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}\overline{W}_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}}{H_{i,j}^{n+1}\Delta y} \\ - \frac{\overline{W}_{i,j}^{n+\frac{1}{2}}}{H_{i,j}^{n+1}}\frac{{}^{y}q_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - {}^{y}q_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta y} + \frac{p_{i,j,1}^{n+\frac{1}{2}} - p_{i,j,0}^{n+\frac{1}{2}}}{H_{i,j}^{n+1}} = \frac{{}^{w}\mathrm{S}_{i,j}^{n+\frac{1}{2}} + {}^{w}\mathrm{T}_{i,j}^{n+\frac{1}{2}}}{H_{i,j}^{n+1}} \tag{2.21}$$

The pressures are defined on the cell faces and therefore do not have to be interpolated. Furthermore, we can exactly set the dynamic pressure at the free surface $p_{i,j,1}^{n+\frac{1}{2}}$ to zero. The vertical velocities are defined on the cell faces and therefore the depth averaged velocity $\overline{W}_{i,j}^{n+\frac{1}{2}}$ needs to be expressed in terms of the bottom and surface velocities. Using a simple central approximation gives

$$\overline{W}_{i,j}^{n+\frac{1}{2}} = \tfrac{1}{2}\left(w_{i,j,1}^{n+\frac{1}{2}} + w_{i,j,0}^{n+\frac{1}{2}}\right), \qquad \overline{W}_{i,j}^{*} = \tfrac{1}{2}\left(w_{i,j,1}^{*} + w_{i,j,0}^{*}\right) \tag{2.22}$$

At the bottom the kinematic boundary condition is used for the vertical velocity:

$$w^{*}_{i,j,0} = \tfrac{1}{2}\left(U^{*}_{i+\frac{1}{2},j} + U^{*}_{i-\frac{1}{2},j}\right)\frac{d_{i+\frac{1}{2},j} - d_{i-\frac{1}{2},j}}{\Delta x_i} + \tfrac{1}{2}\left(V^{*}_{i,j+\frac{1}{2}} + V^{*}_{i,j-\frac{1}{2}}\right)\frac{d_{i,j+\frac{1}{2}} - d_{i,j-\frac{1}{2}}}{\Delta x_j} \tag{2.23}$$

Horizontal interpolation of $\bar{W}^{n+\frac{1}{2}}_{i+\frac{1}{2},j}$ and $\bar{W}^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}$ is done using first order upwind similar to (2.5). The turbulent stresses are again approximated using a central scheme as

$$\begin{aligned} {}^{w}T^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = & \frac{1}{\Delta x_i}\left[{}^{x}\bar{\nu}^{n}_{i+\frac{1}{2},j}\,{}^{x}\bar{H}^{n+1}_{i+\frac{1}{2},j}\frac{W^{n+\frac{1}{2}}_{i+1,j} - W^{n+\frac{1}{2}}_{i,j}}{\Delta x_{i+\frac{1}{2}}} - {}^{x}\bar{\nu}^{n}_{i-\frac{1}{2},j}\,{}^{x}\bar{H}^{n+1}_{i-\frac{1}{2},j}\frac{W^{n+\frac{1}{2}}_{i,j} - W^{n+\frac{1}{2}}_{i-1,j}}{\Delta x_{i+\frac{1}{2}}}\right] \\ & + \frac{1}{\Delta y_i}\left[{}^{y}\bar{\nu}^{n}_{i,j+\frac{1}{2}}\,{}^{y}\bar{H}^{n+1}_{i,j+\frac{1}{2}}\frac{W^{n+\frac{1}{2}}_{i,j+1} - W^{n+\frac{1}{2}}_{i,j}}{\Delta x_{j+\frac{1}{2}}} - {}^{y}\bar{\nu}^{n}_{i,j-\frac{1}{2}}\,{}^{y}\bar{H}^{n+1}_{i,j-\frac{1}{2}}\frac{W^{n+\frac{1}{2}}_{i,j} - W^{n+\frac{1}{2}}_{i,j-1}}{\Delta y_{j+\frac{1}{2}}}\right] \end{aligned} \tag{2.24}$$

Thus combining(2.21), (2.22) and (2.23) explicit expressions for $w^{*}_{i,j,1}$ and $w^{*}_{i,j,0}$ are obtained.

Corrector

The predicted values are again corrected using a variant of the McCormack scheme and including the pressure difference implicitly gives the corrector step:

$$\frac{\bar{W}^{n+1\frac{1}{2}}_{i,j} - \bar{W}^{*}_{i,j,1}}{\Delta t} + \frac{{}^{x}q^{n+\frac{1}{2}}_{i+\frac{1}{2},j}\Delta\bar{W}_{i+\frac{1}{2},j} - {}^{x}q^{n+\frac{1}{2}}_{i-\frac{1}{2},j}\Delta\bar{W}_{i-\frac{1}{2},j}}{H^{n+1}_{i,j}\Delta x} + \frac{{}^{y}q^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}\Delta\bar{W}_{i,j+\frac{1}{2}} - {}^{y}q^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}\Delta\bar{W}_{i,j-\frac{1}{2}}}{H^{n+1}_{i,j}\Delta y} - \frac{\Delta p^{n+1\frac{1}{2}}_{i,j}}{H^{n+1}_{i,j}} = 0 \tag{2.25}$$

Where $\Delta\bar{W}_{i+\frac{1}{2},j}$ and $\Delta\bar{W}_{i,j+\frac{1}{2}}$ are obtained using relations similar to (2.20). Note that similar to (2.22) $\bar{W}^{n+1\frac{1}{2}}_{i,j} = \tfrac{1}{2}\left(w^{n+1\frac{1}{2}}_{i,j.1} + w^{n+1\frac{1}{2}}_{i,j,0}\right)$ and again the kinematic boundary conditions is substituted for $w^{n+1\frac{1}{2}}_{i,j,0}$.

The discrete vertical momentum balance of (2.21) and (2.25) looks very different from the relations found in Zijlema and Stelling (2005), Zijlema and Stelling (2008) and Smit (2008). This is mainly due to the application of the McCormack scheme for the advection. The discretisation of the pressure term is numerically fully equivalent to either the Keller box scheme as used in Zijlema and Stelling (2005), Zijlema and Stelling (2008) or the Hermetian relation used in Smit (2008).

3.3 Non reflective Boundary Condition

The non-reflective boundary conditions used in the default XBeach model have all been derived on the assumption that both the incoming and reflective waves are shallow water waves. In situations where the incident wave field also contains shorter wave components the application of these boundary conditions can lead to significant local distortions of the shape and height of the waves. For this reason a modified version of the radiation boundary in XBeach is used that works better if the incident wave field also contains shorter components.

In order to construct a boundary that can both generate waves and is weakly reflective a few simplifications are made. The boundary is considered to be a straight line. The local coordinate system is such that the x-axis is perpendicular to the boundary and directed positive inwards and the y-axis is parallel to the boundary. Locally, the bottom is approximated as flat and the wave motion is assumed to be linear.

At the boundary the free surface elevation and depth averaged velocity are the summation of the incoming and reflected signal

$$\eta = \bar{\eta} + \eta^{r} + \eta^{in}, \quad \mathbf{U} = \bar{\mathbf{U}} + \mathbf{U}^{r} + \mathbf{U}^{in} \tag{2.26}$$

Where both $\bar{\eta}$ and $\bar{U}$ are the mean free surface location and velocity which may vary on timescales that are much longer than the typical timescales of the incident and reflective signals.

The incident wave signal at the boundary is composed of *N* incoming long crested harmonic free linear waves. Each wave travels with its own celerity c_k along a straight ray in the direction θ_k. Along each wave ray a local coordinate system *s,t* is prescribed with s parallel to the wave ray and *t* perpendicular to the ray. The time varying depth averaged velocity due to harmonic *k* in the direction of *s* is denoted by $\hat{u}_k$. At a certain point on the boundary the total depth averaged velocity and surface elevation due to the incident waves is given by

$$\mathbf{u}^{in} = \sum_{k=1}^{N} \mathbf{u}_k^{in}, \quad \eta^{in} = \sum_{k=1}^{N} \eta_k^{in} \quad \text{with } u_k = \left(u_k, v_k\right), \; u_k = \hat{u}_k \cos\theta_k \; \text{ and } v_k = \hat{u}_k \sin\theta_k \tag{2.27}$$

Assuming that the wave form remains constant along the individual wave rays each of the harmonics obeys

$$\frac{\partial \hat{u}_k^{in}}{\partial t} - c_k \frac{\partial \hat{u}_k^{in}}{\partial s} = 0 \tag{2.28}$$

For linear waves this condition can also be rewritten as

$$\begin{aligned} u_k^{in} - c_k \cos\theta_k \frac{\eta_k^{in}}{H} = 0 \\ v_k^{in} - c_k \sin\theta_k \frac{\eta_k^{in}}{H} = 0 \end{aligned} \tag{2.29}$$

For application in XBeach we are only interested in the velocity component perpendicular to the boundary and we will therefore ignore the *v* components.

The reflected signal also consists of different wave components each travelling in its own direction. However it is difficult to obtain accurate information of the directional spectrum of the reflected signal from the model. Therefore the assumption is made that the reflected wave signal only contains relatively long waves. For typical XBeach applications this is a reasonable assumption as most short wave energy is lost due to breaking and only the infragravity waves are reflected. The second approximation is that the long waves travel roughly in a direction perpendicular to the boundary. In this case the radiation condition becomes

$$\frac{\partial u^{r}}{\partial t} - c\frac{\partial u^{r}}{\partial x} = 0 \tag{2.30}$$

where *c* is the shallow water wave celerity. Again assuming a linear wave (2.30) can be rewritten as

$$U^{r} + \frac{c}{H}\eta^{r} = 0 \tag{2.31}$$

This relation is only valid for long waves travelling perpendicular to the boundary. For waves with a parallel component a boundary based on (2.31)will generate reflections.

To obtain an expression for the velocity at the boundary we take the sum of the relation (2.29) and (2.31) resulting in

$$\sum_{k=1}^{N} U_k^{in} + U^r + \bar{U} - \sum_{k=1}^{N} c_k \frac{\eta_k^{in}}{H} + \frac{c}{H}\left(\eta - \bar{\eta} - \sum_{k=1}^{N} \eta_k^{in}\right) = \bar{U} \tag{2.32}$$

Recognizing (2.26) in the first few terms this is written as

$$U_b = U_i + \bar{U} + \sqrt{\frac{g}{H}}\left(\eta - \bar{\eta} - \eta^{in}\right) \tag{2.33}$$

This is the weakly reflective boundary that is used in the non-hydrostatic simulations. Notice that the most restrictive condition on this boundary is that the reflective waves are assumed to be perpendicular to the boundary. This is acceptable for comparisons with flume type experiments, however for more complicated geometries this will result in significant errors.

Notice that the boundary condition (2.33) is very similar to the radiation boundary that is implemented into XBeach. The only significant difference is that not only an incident velocity signal has to be provided but also an incident free surface elevation.

3.4 Flooding and drying

The flooding and drying procedures and XBeach are similar to those described in Stelling and Duinmeijer (2003). Application of their momentum conservative scheme while adhering to the CFL condition generates strictly positive water depths and there is therefore no need to include special flooding and drying procedures.

For efficiency reasons velocity points are removed from the computation when the local water level is below a certain threshold value ε. If a certain velocity point is marked dry the velocity is set to zero and the calculation of the momentum equations is skipped. . This also means that no coupling is introduced in the pressure coefficient matrix between i and $i+1$.

Pressure points are considered dry when all of the surrounding velocity points are dry. When this is the case the pressure point is no longer included in the computations and the main diagonal of the linear system is set to one while the right hand side is set to zero.

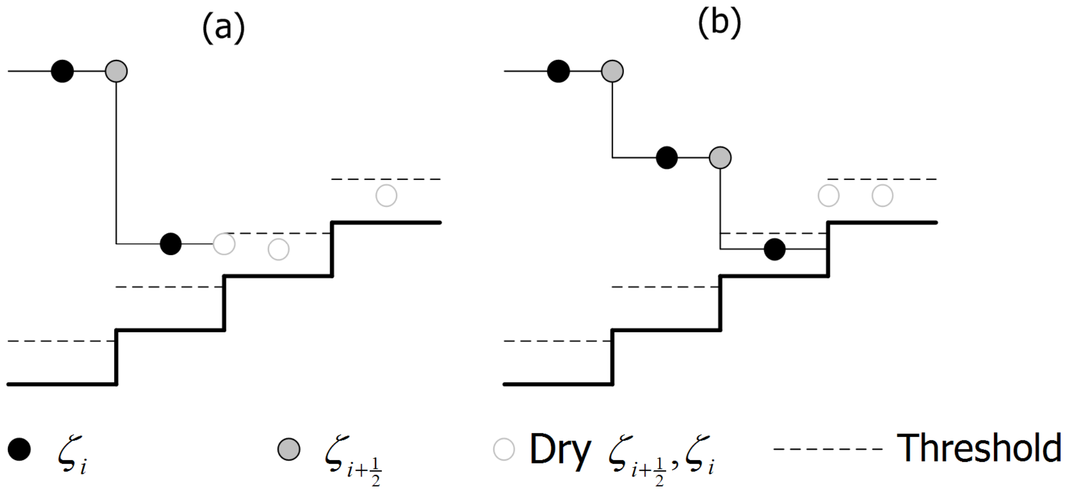

(a)
(b)
● ζ_i
● $\zeta_{i+\frac{1}{2}}$
○ Dry $\zeta_{i+\frac{1}{2}},\zeta_i$
------- Threshold

Figure 3-5 (a) A velocity point is considered dry if the total water depth is under a certain threshold. (b) A water level point is considered wet if any of the surrounding velocity points are wet, even if the water depth itself is below the threshold.

Near the wet dry boundary the application of a higher order scheme can result in erroneous behaviour. Especially the inter-/extrapolation of the free surface along a sloping bottom can lead to irregular flooding and drying behaviour. To prevent this, the higher order scheme is only applied when the lowest free surface value in the computational stencil is higher than the highest bottom level in the stencil. Thus in case of positive flow direction the limiter in (2.7) is only applied when

$$\min\left(\zeta_{i-1},\zeta_i,\zeta_{i+1}\right) > \max\left(-d_{i-1},-d_i,-d_{i+1}\right) \tag{2.34}$$

Using this criterion only FOU approximations are used near to wet/dry boundary and on steep slopes. Experience has shown that application of this criterion leads to much smoother flooding and drying behaviour.

Disabling the pressure correction in shallow water

If the local water depth is relatively small most waves of practical interest have become shallow water waves. From a numerical point of view the smallest wave that can be represented on the grid has a wavelength of $\lambda = 2\Delta x$. If we define a shallow water wave as a wave for which $kH < \left(kH\right)_{\min}$ then the smallest wave has become a shallow water wave when the depth is:

$$H < H_{threshold} \qquad H_{threshold} = \frac{\Delta x}{\pi}\left(kH\right)_{\min} \tag{2.35}$$

If $H < H_{threshold}$ then it appears appropriate to disable the calculation of the non-hydrostatic pressure by removing the point from the linear system in a similar way as for a dry point. Removing these points might improve the convergence rate and stability of the pressure correction algorithm. Presently it appears that for low values of $\left(kH\right)_{\min}$ there is hardly any difference in simulations without the removal of these points. For this reason the default value of $\left(kH\right)_{\min}$ is set to zero, which practically means that points are never removed for this reason.

3.5 Solution method

Linear system for the non-hydrostatic pressure

Due to the assumption of incompressible flow there is no longer a time evolution equation for the non-hydrostatic pressure. Because of this the non-hydrostatic pressure can only be determined by enforcing that the resultant flow field is divergence free. Thus the non-hydrostatic pressure cannot be determined explicitly. Instead a discrete Poisson equation for the pressure has to be solved at each time step.

The linear system is obtained after substitution of the momentum equations (2.19) and (2.25) into the local continuity equation(2.10). Because each horizontal component couples two pressure points and the vertical velocity potentially couples (depending on the bottom) all five surrounding points, the resulting equations have the form

$$A^{n}_{i,j,k}\Delta p_{i,j-1} + A^{w}_{i,j}\Delta p_{i-1,j} + A^{p}{}_{i,j}\Delta p_{i,j} + A^{e}_{i,j}\Delta p_{i+1,j} + A^{s}_{i,j}\Delta p_{i,j+1} = rhs_{i,j} \tag{2.36}$$

Rearranging the pressure points in a single vector the system can be written as a penta-diagonal band matrix. The main diagonal consists of the coefficients A^p and A^e, A^w, A^n, A^s are the diagonals relating to the surrounding points (west, east, north, south). Inverting this system gives the pressure difference and using (2.14) we obtain the dynamic pressure.

Matrix Solver

To solve the resulting linear system the strongly implicit procedure (SIP) due to Stone (1968) has been used. This method was specifically designed for elliptic problems and sparse banded matrices. In general SIP solver requires more iterations per time step than for instances the preconditioned BiCGSTAB method. However, the work done per iteration is usually much lower and on balance the SIP method is faster than the BICGSTAB method.

The SIP solver is based on an incomplete lower-upper factorization of the matrix **A**. This factorization is constructed in such a way that it has the same sparsity as the original matrix. The resulting system can then be solved very efficiently in an iterative manner using forward and backward substitutions.

To ensure convergence the SIP solver uses a relaxation factor α . For convergence α should be set lower than one and even then there are certain situations in which the method may fail to converge. Experience has shown that when $\alpha < 0.92$ the method virtually always converges and this is therefore the default value used in XBeach.

The method relies on an iterative solution of the linear system and generates intermediate solutions which improve in accuracy when more iterations are taken. There is usually no need to continue this procedure until machine precision is reached, since the errors made in the underlying discretisation process are usually much larger than the accuracy of the computer arithmetic. A stopping criterion is needed that determines when the solution is considered to be sufficiently accurate. Here, similar as in Stelling and Zijlema (2003), the following criterion is adopted

$$\frac{\left\| \mathbf{A}\Delta\mathbf{p}^{s} - \mathbf{Q} \right\|}{\left\| \mathbf{Q} \right\|} < \delta \tag{2.37}$$

Where δ is a pre determined threshold, $\mathbf{Q}$ the right hand side of the linear problem, $\|\ \|$ is the L_2 norm of the respective vector and $\Delta\mathbf{p}^{s}$ the intermediate solution. Interestingly it appears that the solution of the pressure matrix is not very sensitive to the threshold δ and it usually suffices to set it to about $\delta = 10^{-2}$ (see Stelling and Zijlema, 2002). If for some reason the method has not reached convergence after $S_{\max}$ iterations the iteration process is also terminated and the last iterative solution is used. By default this is set to $S_{\max} = 20$ iterations.

Because the SIP solver is based on an incomplete LU factorisation it is an inherently serial algorithm. This makes the SIP solver ill suited for parallelization. Parallelisation through domain decomposition using an explicit coupling between the domains most likely will work, but will require a large number of outer iterations for convergence.

For one-dimensional situations (e.g. when simulation a flume experiment) the linear system reduces to a tri-diagonal system. For such linear systems a direct solver, known as the Thomas or tri-diagonal matrix algorithm (TDMA), can be used to solve the linear system. The TDMA is generally faster and more accurate than the SIP solver and therefore always recommended for one dimensional simulations.

Computation sequence

The full computational sequence for a single time step can now be described as follows

1. Calculate the intermediate velocities $U^{*}_{i+\frac{1}{2}}, V^{*}_{i+\frac{1}{2}}, W^{*}_{i}$ with equations (2.17) and (2.21) using the velocities $U^{n+\frac{1}{2}}_{i+\frac{1}{2}}, V^{n+\frac{1}{2}}_{i+\frac{1}{2}}, W^{n+\frac{1}{2}}_{i}$, fluxes ${}^{x}q^{n+\frac{1}{2}}$, ${}^{y}q^{n+\frac{1}{2}}$ pressure $p^{n+\frac{1}{2}}$ and the free surface ζ^{n+1}.
2. Correct the intermediate velocities $U^{*}_{i+\frac{1}{2}}, V^{*}_{i+\frac{1}{2}}, W^{*}_{i}$ using the limited version of the McCormack scheme by applying the equations (2.19) and (2.25) ignoring the terms involving the pressure difference to obtain $U^{**}_{i+\frac{1}{2}}, V^{**}_{i+\frac{1}{2}}, W^{**}_{i}$. If the second order calculations are disabled set $U^{**}_{i+\frac{1}{2}}, V^{**}_{i+\frac{1}{2}}, W^{**}_{i}$ equal to $U^{*}_{i+\frac{1}{2}}, V^{*}_{i+\frac{1}{2}}, W^{*}_{i}$.
3. To obtain the linear system for the pressure difference $\Delta p^{i+1\frac{1}{2}}$ substitute the equations (2.19) and (2.25) including the terms involving the pressure difference into equation (2.10).
4. Use to SIP solver to generate iteratively approximations to the linear system until the required accuracy has been reached or when the number of iterations is larger than the maximum number of iterations.
5. Calculate $U^{i+1\frac{1}{2}}, V^{i+1\frac{1}{2}}$ by substituting $\Delta p^{i+1\frac{1}{2}}$ in equations (2.19)
6. Calculate $W^{i+1\frac{1}{2}}$ by substituting $U^{i+1\frac{1}{2}}, V^{i+1\frac{1}{2}}$ in equation (2.10) or alternatively by substituting $\Delta p^{i+1\frac{1}{2}}$ in equation (2.25).
7. Calculate $p^{i+1\frac{1}{2}}$ by substituting $\Delta p^{i+1\frac{1}{2}}$ in equation (2.14)
8. Calculate the fluxes ${}^{x}q^{*}$, ${}^{y}q^{*}$ using $U^{i+1\frac{1}{2}}, V^{i+1\frac{1}{2}}$ and ζ^{n+1}
9. Calculate the intermediate free surface ζ^{*} by substitution of ${}^{x}q^{*}$, ${}^{y}q^{*}$ in equation (2.4)
10. Calculate ζ^{n+2} by substitution of ζ^{*} into equatiob (2.6)
11. Calculate the corrected fluxes ${}^{x}q^{n+1\frac{1}{2}}$, ${}^{y}q^{n+1\frac{1}{2}}$ using equation (2.8).

4 Verification and validation

4.1 Linear dispersion

The inclusion of the non-hydrostatic pressures introduces a coupling between the wave number and frequency into the classical NSWE. However, the dispersion relation that is represented by the numerical model in not exact and it is therefore necessary to asses how well the approximations perform.

Dispersive properties of the numerical method

To investigate the linear dispersive properties of the depth averaged system we perform a semi-discretisation in the vertical and linearizing the resulting equations. By substituting harmonic components and demanding that the resulting system has more solutions than only the trivial solution it can be shown that (see Appendix A) the dispersion relation is of the form:

$$\omega_{num} = k\sqrt{\frac{gH}{1+\frac{1}{4}\left(kH\right)^2}} \tag{2.1}$$

Furthermore the group velocity and wave celerity for the semi-discrete system become:

$$c_{g,num} \equiv \frac{d\omega}{dk} = \sqrt{\frac{gH}{1+\frac{1}{4}\left(kH\right)^2}}\left[1-\frac{1}{1+4\left(kH\right)^{-2}}\right] \tag{2.2}$$

$$c_{num} \equiv \frac{\omega}{k} = \sqrt{\frac{gH}{1+\frac{1}{4}\left(kH\right)^2}} \tag{2.3}$$

These solutions are similar to the dispersion relations present in lower order Boussinesq models (Dingemans, 1997).

In Figure 4-1 these relations are compared to the expression from linear wave theory. For small values of *kd* the expression (2.1) - (2.3) approximate linear wave theory well. For *kd<1* the relative error remains approximately below the 5%. However for *kd>1* the error grows rapidly and the equations are no longer a good approximation to linear wave theory. For these shorter waves a higher resolution in the vertical is required.

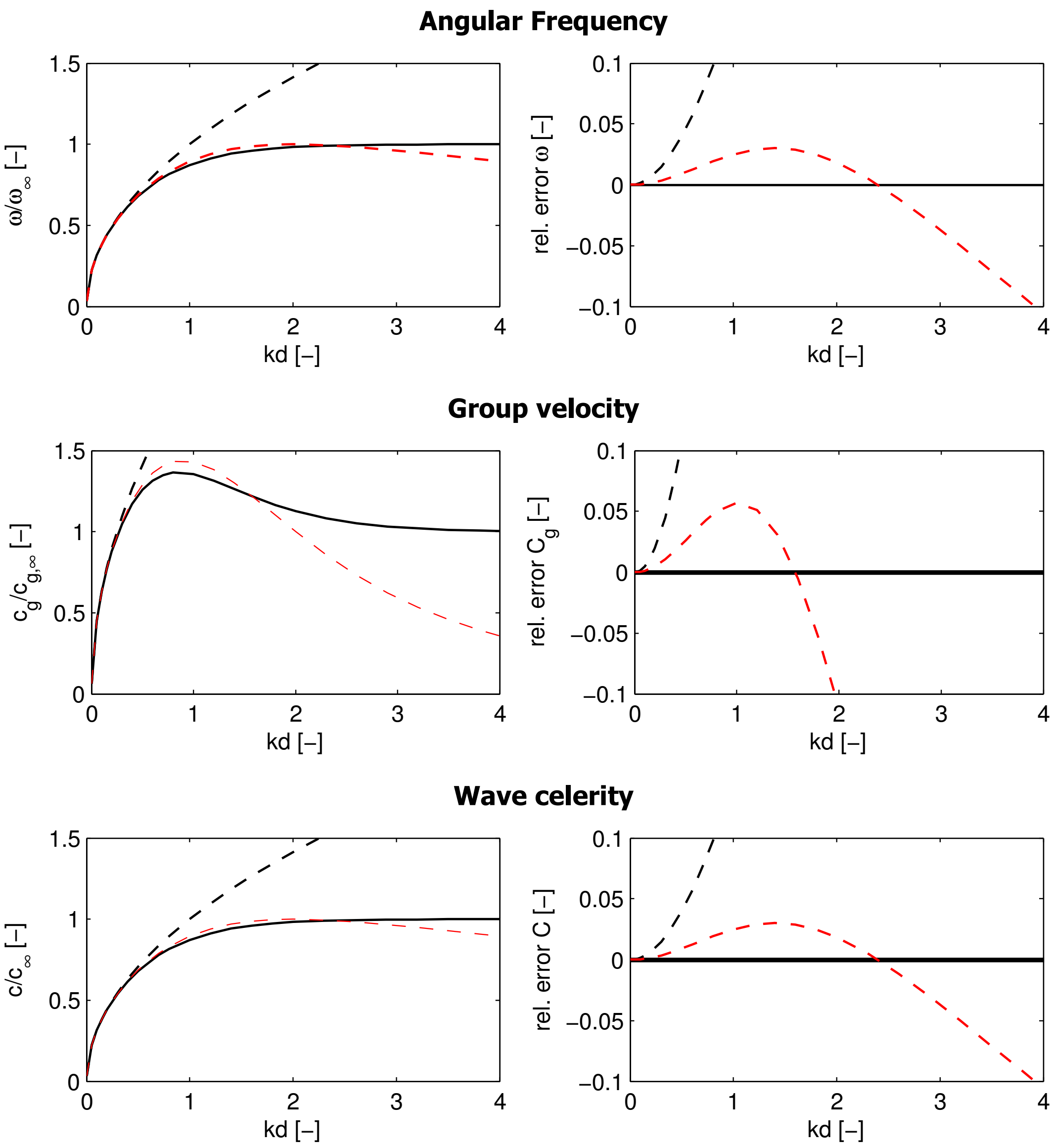


Figure 4-1 The angular frequency, group velocity and wave celerity for a constant wave length in varying depth. Computed from linear wave theory (black line), the shallow water approximation (dotted black line) and the semi discrete relations (dotted red line). The left hand panels give the values scaled with their respective deep water values, while the right hand panels show the relative errors compared to linear wave theory.

Oscillating basin

One of the standard tests in literature to test the dispersive properties of models based on Boussinesq like equations and on the Non-hydrostatic approach is a standing linear wave in a basin. Here we use this test to verify the dispersive properties of the model as derived in the previous paragraph.

Consider a three dimensional square basin with sides of length *L* and still water depth *H*. The fundamental mode of such a basin can be described with a wave like solution given by:

$$\zeta(x,y,t)=\zeta_0\cos(k_x x)\cos(k_y y)\cos(\omega t) \tag{2.4}$$

Where $k_x = k_y = \pi L^{-1}$ and ω is determined by the linear dispersion relation. As only waves with a small steepness are considered the depth averaged velocities can be approximated safely with linear wave theory and are given by:

$$\begin{aligned} U(x,y,t)&=\frac{\eta_0 g k_x}{kH\omega}\sin(k_x x)\cos(k_y y)\sin(\omega t)\\ V(x,y,t)&=\frac{\eta_0 g k_y}{kH\omega}\cos(k_x x)\sin(k_y y)\sin(\omega t)\end{aligned} \tag{2.5}$$

The velocity field and the surface elevation at *t=0s* are used as initial conditions for the test. The basin length is set at $L=100m$ with a relative depth of *kH=0*.5. The basin is discretized using $N_x \times N_y = 100\times 100$ gridpoints with a CFL condition of 0.5.

Figure 4-2 **a-b** shows model results of the free surface and velocity in ***x*** *= (1,1).* For both the velocities and the free surface the amplitude is correctly modelled for the five periods shown. There does appear to be a slight increase in period but this is to be expected from (2.1). In Figure 4-3 contours are shown for four different times during the first half of the fourth period, together with the velocity vectors and the contours of the analytical solution. Aside from the difference in phase the numerical solution reproduces the correct behaviour. The solution stays symmetric around the diagonals and the velocity pattern is correct

In order to investigate the range for which the one layer model return valid results a similar setup is used for a range of kh values. Keeping the wave length, and thus the basin size, constant the depth is gradually increased. In this way standing waves with a range of $0.1 \le kh \le 7$ were modelled. Using the distance between two consecutive upward zero-crossings in the time-signal as a measurement for the wave period the angular frequency for each run is determined. Comparing the results (Figure 4-4) to (2.1), using a sufficiently fine horizontal grid, the model exactly follows the semi-discrete dispersion relation. Therefore the conclusions regarding the semi-discrete model are also applicable to the fully discrete model.

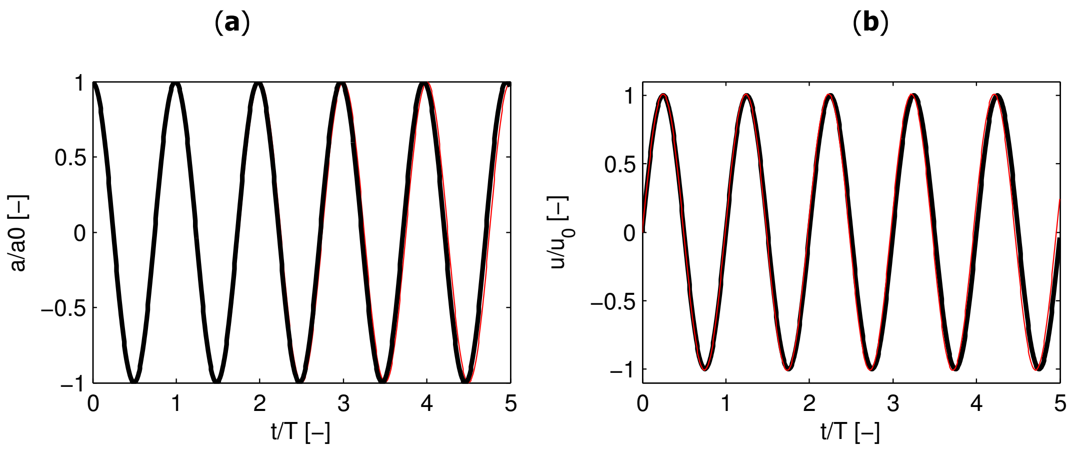


Figure 4-2 (**a**) Free surface elevation at (1,1) scaled with the amplitude. Analytical solution (black) compared to the numerical solution (red). (**b**) Horizontal velocity scaled with the velocity amplitude. Analytical solution (black) compared to the numerical solution (red).

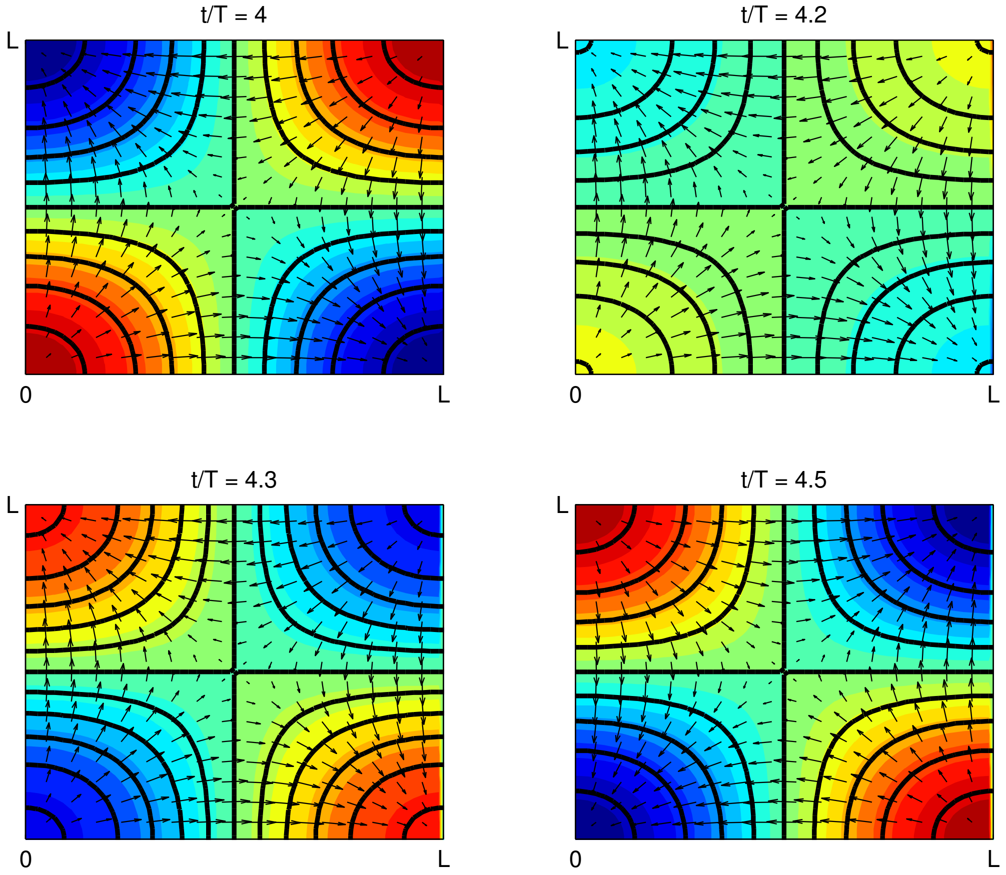


Figure 4-3 Contour plots of the free surface at different time intervals. The analytical solution is indicated with the thick black lines while the numerical solution is indicated by the coloured patches.

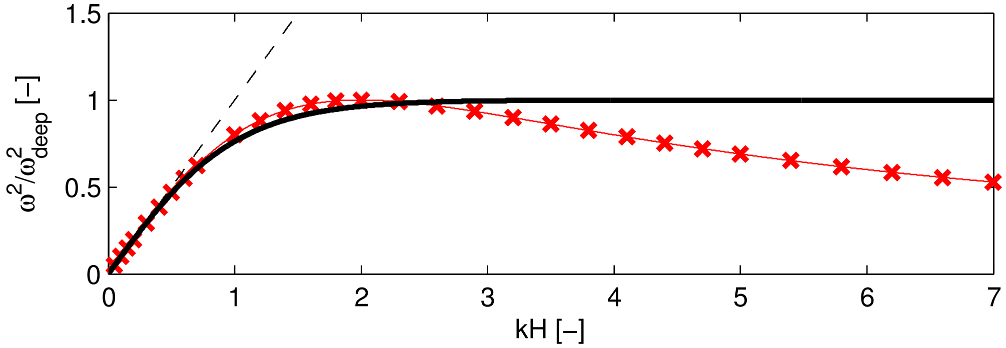


Figure 4-4 The dispersion relation for a wave with constant wavelength in varying depth. Shown are the linear dispersion relation (thick black line),semi-discrete dispersion relation (red line), model results (red crosses) and the shallow water dispersion relation (striped black line).

Enhanced dispersion

Because of the low resolution in the vertical, the dispersion relation for the one layer system only approximates the linear dispersion well in a certain range. In order to control this range it is possible to introduce a tuning coefficient α into the vertical momentum equation:

$$\frac{Dw}{Dt} + \alpha \frac{\partial p}{\partial z} = 0 \tag{2.6}$$

Although this coefficient has no physical meaning it introduces an extra degree of freedom in the numerical dispersion relation. Equation (2.1) now becomes

$$\omega_{num} = k\sqrt{\frac{gH}{1 + \frac{1}{4\alpha}(kH)^2}} \tag{2.7}$$

In Figure 4-5 the numerical dispersion relation and group velocity are compared to linear wave theory for different values α in the range of $0.7 - 1.3$. The figures show that for $kH < 2$ it is generally better to set $\alpha < 1$ while for $kH > 2$ a better correspondence is achieved with $\alpha > 1$. Because we are able to control the shape of the function it is possible for a certain combination of kH and ω to find a value for α for which $\omega_{num} = \omega_{airy}$ holds. From (2.7) we get:

$$\alpha = \left(\frac{4g}{\omega^2 d} - \frac{4}{(kH)^2}\right)^{-1} \tag{2.8}$$

Where ω and k are related through the usual linear dispersion relation. Thus if the characteristic frequency of a harmonic wave is known in advance, the coefficient can be set in such a way that the dispersion relation is locally exact. To illustrate this, the oscillating basin is modelled again, but now with an optimal value of α for each run. Figure 4-6 shows that in this way it is in principle possible to exactly follow the linear dispersion relation. However, when optimizing for a harmonic with a large *kH*, this results in larger errors for the longer wave components. Unless the frequency range is narrowly focussed it is therefore better to optimize α for a certain range.

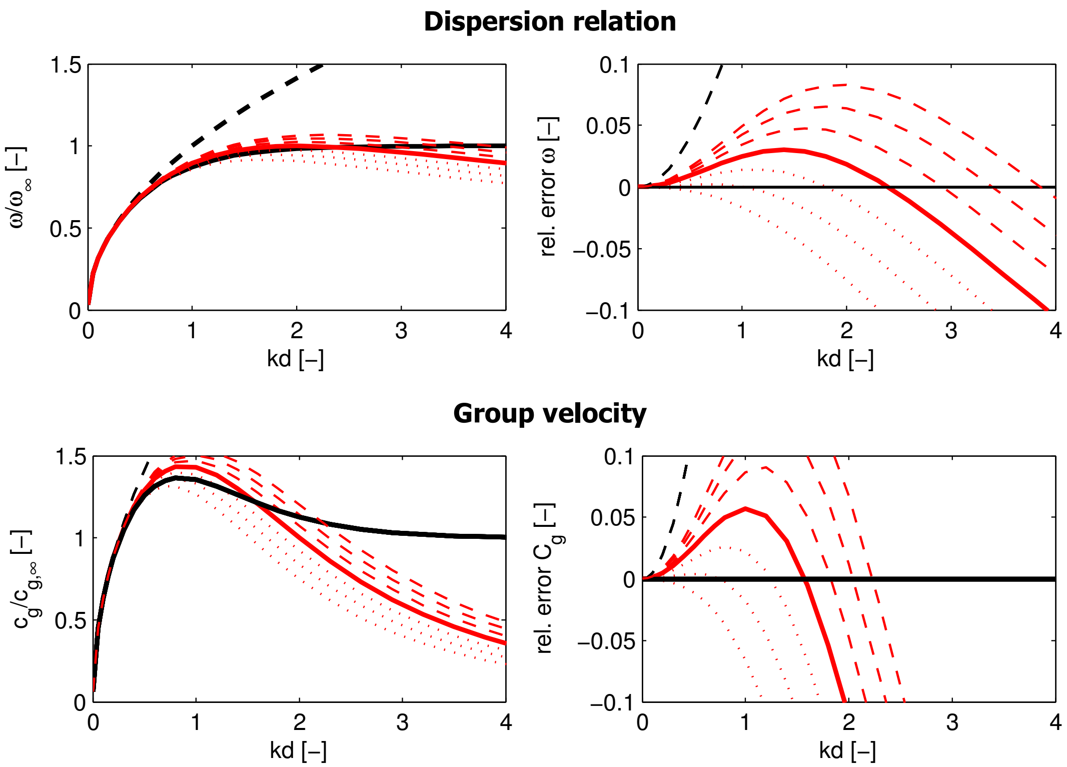


Figure 4-5 The dispersion relation and group velocity for a constant wave length in varying depth. Computed from linear wave theory (black line), the shallow water approximation (dotted black line) and the semi discrete relations (red lines) with different values of alpha. The left hand panels give the values scaled with their respective deep water values, while the right hand panels show the relative errors compared to linear wave theory.

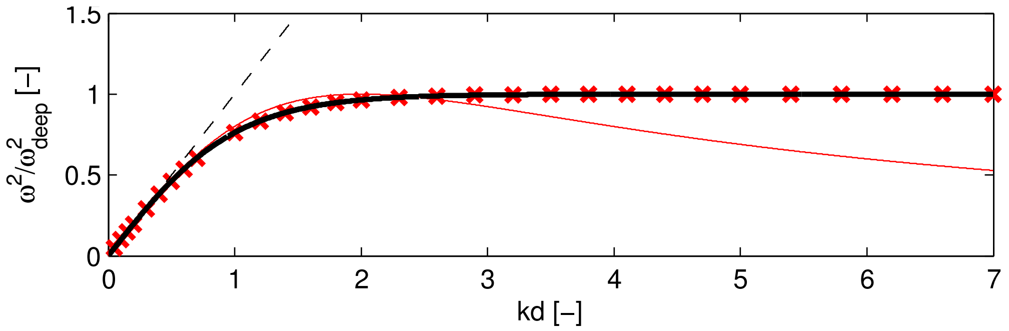


Figure 4-6 Linear dispersion relation (thick black) line compared to the numerical dispersion relation (red line) and the numerical dispersion relation with optimized coefficients (red crosses).

As XBeach is specifically designed for near shore applications the most interesting region to optimise for is $0<kH<kH_{max}$. *With* $kH_{max}\sim 1$. For instance, for $kH_{max}=1$, the maximum relative error over the interval is minimized for $\alpha \approx 0.8$ (see Figure 4-7**a**). In Figure 4-7**b** the optimum value for α is given as a function of kH_{max} while Figure 4-7**c** indicates the maximum relative error at this optimum value. Notice that the relative error in the group velocity is always the largest.

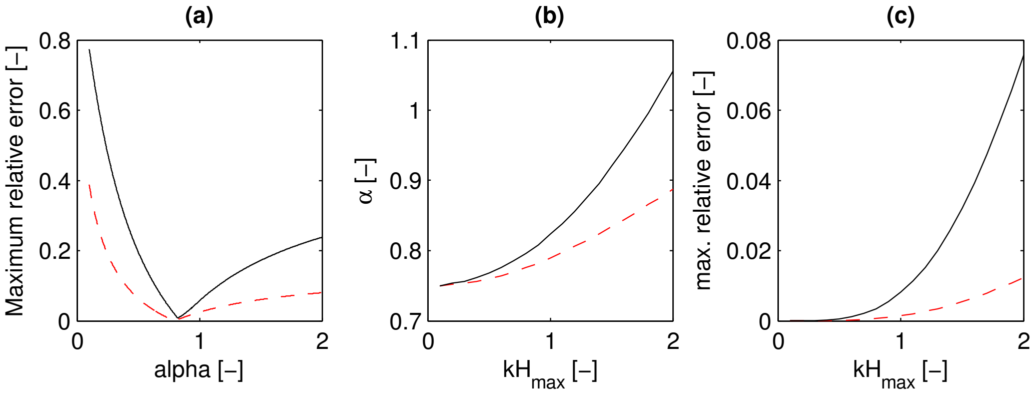


Figure 4-7 **(a)** The maximum relative error in the range $0<kH<1$ as a function of α for c_g (black) and ω (red). **(b)** The optimum value of α minimizing the maximum relative error over the range $0<kH<kH_{max}$. Optimum value for c_g (black) and ω (red) **(c)** The maximum relative error at the optimum value of α over the range $0<kH<kH_{max}$.. Error indicated for c_g (black) and ω (red).

From Figure 4-7**(b)** clearly shows that, when optimizing XBeach for linear dispersion, α should be set lower than one to get optimal results. The maximum relative error in the group velocity can be greatly reduced in this way. This should improve the correspondence with results from linear wave theory for shoaling and refraction. Currently it is unknown how the coefficient will influence non-linear processes (triad interactions, bound harmonics etc.), therefore some care should be taken in cases where non-linear effects are significant.

4.2 Solitary wave in a channel

introduction

The solitary wave is a non-linear wave of finite amplitude with the characteristic property that it is neither preceded nor followed by any surface disturbance. An interesting aspect of this test is that the solitary wave is not a solution of the shallow water equations and, as such, cannot be reproduced in models based on the hydrostatic pressure assumption.

The analytical approximations for the surface elevation are dependent on the fundamental equations considered (e.g. de Korteweg de Vries equations or the improved Boussinesq equations) but all generally are obtained from a perturbation analysis using the ratio of wave height to water depth ε as an expansion parameter:

$$\varepsilon \equiv \frac{\eta_0}{d_0} \ll 1 \tag{2.9}$$

here η_0 is the amplitude of the solitary wave and d_0 the still water depth. The classical solitary wave solutions are accurate to the lowest order in this expansion parameter. Here we will use a third order accurate solution due to Grimshaw (1971) that can be found in Fenton (1972)[1]. This solution is obtained from a series expansion of the non-linear equation where all terms of order ε^4 and higher are neglected. In this case the expression for the surface elevation becomes:

$$\begin{aligned}\frac{\eta(x,t)}{d_0} &= \varepsilon s^2 - \tfrac{3}{4}\varepsilon^2 s^2 th^2 + \varepsilon^3\left(\tfrac{5}{8}s^2 th^2 - \tfrac{101}{80}s^4 th^2\right)\\ \alpha &= \sqrt{\tfrac{3}{4}\varepsilon}\left(1 - \tfrac{5}{8}\varepsilon + \tfrac{71}{128}\varepsilon^2\right)\\ \frac{c}{\sqrt{gd_0}} &= 1 + \tfrac{1}{2}\varepsilon - \tfrac{3}{20}\varepsilon^2 + \tfrac{3}{56}\varepsilon^3\end{aligned} \tag{2.10}$$

Where $s = \operatorname{sech}\alpha\left(x - x_0 - ct\right)$ and $th = \tanh\alpha\left(x - x_0 - ct\right)$. Notice that the leading order term is the classical Boussinesq solitary wave.

As a characteristic horizontal length scale L_{sol} two times the distance between x_0 and x_δ is used. Here we define x_δ as the position on the axis where the surface elevation has reduced to a small percentage of the maximum elevation $\delta = \eta\left(x_\delta\right)/\eta_0$. For practical purposes the length is defined according to the lowest order terms. As this wavelength will only be used as a scaling parameter this is a reasonable approximation. This leads to:

$$L_{sol} = \sqrt{\frac{4}{3\varepsilon}}\,\operatorname{acosh}\left(\sqrt{\frac{\varepsilon}{\delta}}\right) \tag{2.11}$$

The fraction is set to $\delta = \tfrac{1}{20}\varepsilon$ throughout the tests performed.

[1] The expressions as given by Grimshaw contain a slight error, the correct expressions are given by Fenton.

Setup

The test domain consists of a one-dimensional channel with a length of $L_{channel} = 100L_{sol}$. Furthermore the still water depth is set at 1m throughout the domain. The domain is discretized using 160 points per wavelength and the CFL condition is set at 0.9 throughout the simulation.
At the left hand boundary an incoming solitary wave is prescribed according to (2.10) with $x_0 = -2L_{sol}$. At the right hand side a non reflective boundary by van Dongeren and Svendsen (1997) is used and the total simulation time is $T = 104L_{sol} / c$. In this way the solitary wave traverses the entire domain.

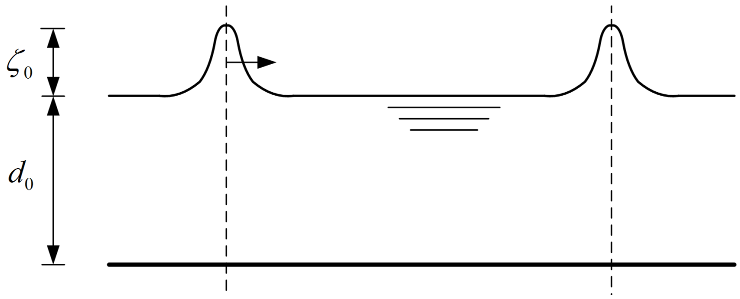


Figure 4-8 Sketch of a solitary wave propagating in an one-dimensional channel

Three different incoming solitary waves are prescribed, each with different value for ε $(0.1,\ 0.2,\ 0.4)$. In this way the effect of the correct balance between non-linearity and dispersion can be assessed. For higher incoming waves the characteristic length scale decreases, and the profile becomes steeper. Note that the domain length and simulation time as defined above also change with changing incident wave height.

Results

In Figure 4-9 the free surface profiles at three different locations along the channel are shown for each incident waves together with the analytical solutions by Grimshaw (1971). Because the solution prescribed at the boundary is not an exact solution for the discrete system the incident wave changes slightly in height and shape over the first part of the channel. Some higher harmonic components are left behind the leading wave which results in an oscillatory tail. Due to dispersion the tail travels slower than the leading component and as such is generally left behind. This is the reason it does not show up prominently in Figure 4-9, even though it was present in all cases.

After a distance of $\sim 10L_{sol}$ the waves have evolved into their final stable shape that tends to be steeper than the prescribed solution. For $\varepsilon = 0.1$ and $\varepsilon = 0.2$ the differences between the analytical solution and the model remain small and surface profiles match favourably. Because the wave height is slightly over predicted in the case of $\varepsilon = 0.1$ the solitary wave travels slightly faster and leads the analytical solution while for $\varepsilon = 0.2$ the situation is reversed. When $\varepsilon = 0.4$ the amplitude is noticeably lower and this results in slower propagation of the profile through the channel. However, after the initial deformation, a stable shape emerges which propagates with constant speed and amplitude.

Better results can be expected if multiple layers are used in the vertical. The enhanced dispersive characteristic of such a model eliminates the formation of an oscillating tail and thus the initial deformation of the profile is greatly reduced. An example of this can be found in Smit (2008) where the two layer model performs significantly better.

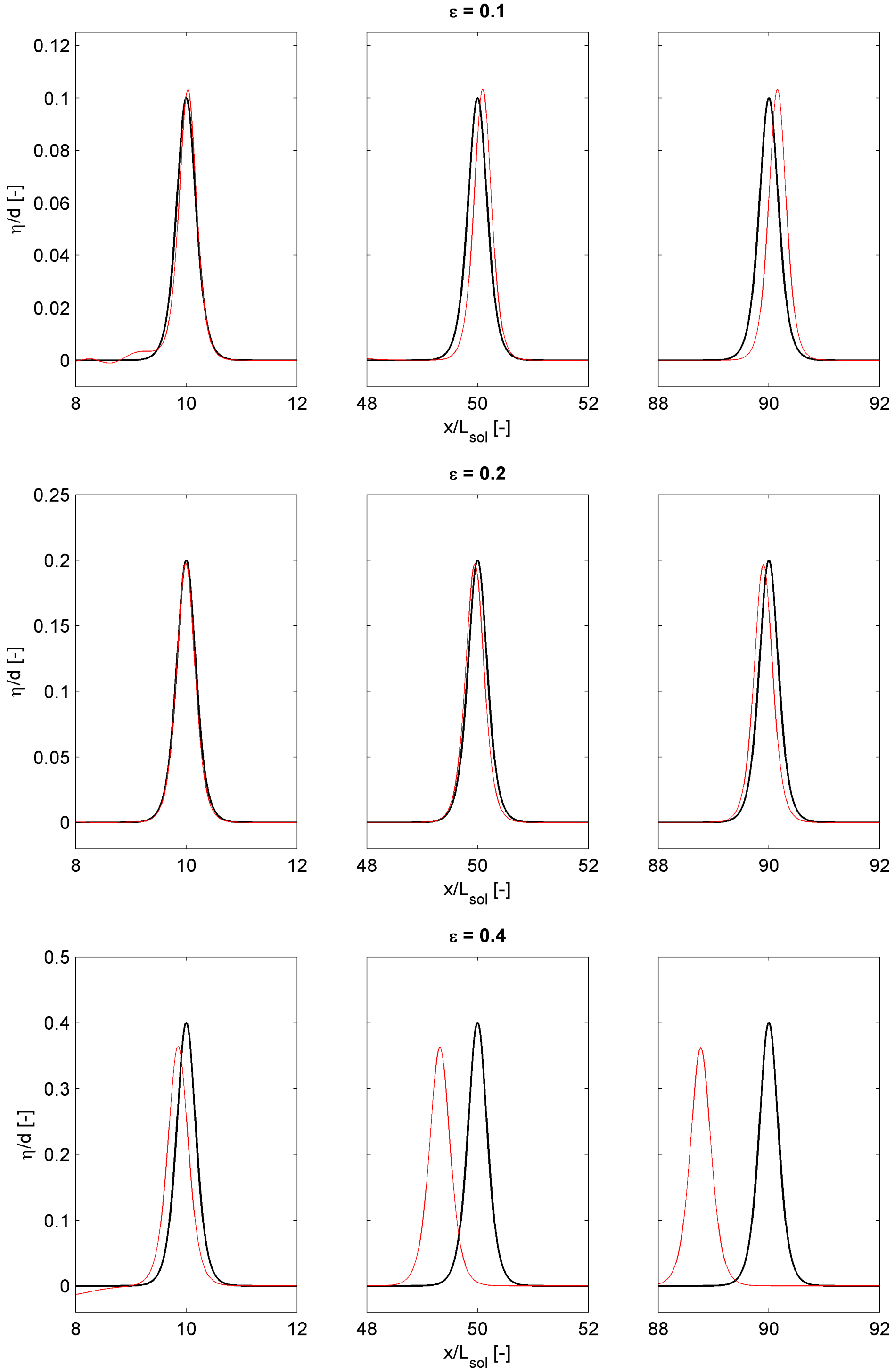


Figure 4-9 The free surface elevation of a solitary wave propagating through a channel of uniform depth. Model (red) and analytical (black) results for three different incident waves shown at three different locations in the channel.

4.3 Dambreak

Introduction

In order to test if the implementation of the advection scheme is indeed momentum conservative the dam break case is considered. This case was also considered in Stelling and Duinmeijer (2003) where they showed that the depth averaged version of their conservative scheme indeed reproduces the dam break wave well. Here we will use this scenario to verify that second order scheme can reproduce these results.

The case considers two regions of fluid with different water levels and fluid velocities which are initially separated by a vertical wall. At t=0 the wall is suddenly removed and a flood wave enters the downstream portion of the canal. The so called dry bed case was first considered by Ritter (1892) and considers a flat bed where the downstream region is completely dry and the initial velocities are zero in the upstream portion. He derived an analytical solution to this problem from the Saint-Venant equations. His solutions were later extended to incorporate a non-zero water level downstream with non-zero initial velocities (see for example Stoker (1957)). Later contributions by Dressler (1952) and Whitham (1955) also added the influence of friction.

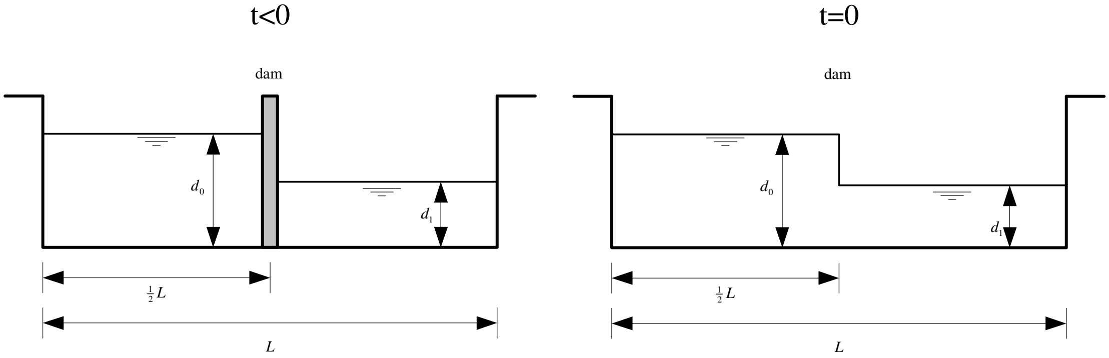


Figure 4-10 Dam break case. Two reservoirs are separated by a structure (left), but at t=0 suddenly the structure is removed (right).

For our purposes the friction is ignored and the model results are only compared to the wet and dry bed analytical solutions. For convenience only zero initial velocities are considered in the wet bed case.

Setup

As noted the analytical solutions where derived from the Saint-Venant equations which in turn means that they implicitly contain the hydrostatic pressure assumption. As vertical accelerations are negligible everywhere except in the region of the discontinuity this seems justified. For this reason the comparisons are made with the non-hydrostatic corrections disabled. Furthermore advection is calculated using both the first order upwind (FOU) and the limited McCormack scheme. In this way the original XBeach code results can be compared to the new implementation.

The computational domain consisted of two thousand grid points with an uniform grid spacing and a total length of $L = 100m$. The dam was located at the centre of the domain ($x = 50m$). For both the dry and wet bed case the upstream water level was $d_0 = 1m$ while the downstream water level in the wet bed case was initially $0.1m$. The CFL condition was set to 0.4. Finally the flooding and drying threshold was set to $10^{-10}m$, this was necessary to capture the wave celerity in the dry bed case correctly.

Results

The free surface after 7 seconds is drawn in Figure 4-11**a**. The results for the FOU and the McCormack schemes are virtually identical and are indistinguishable from the analytical solution.

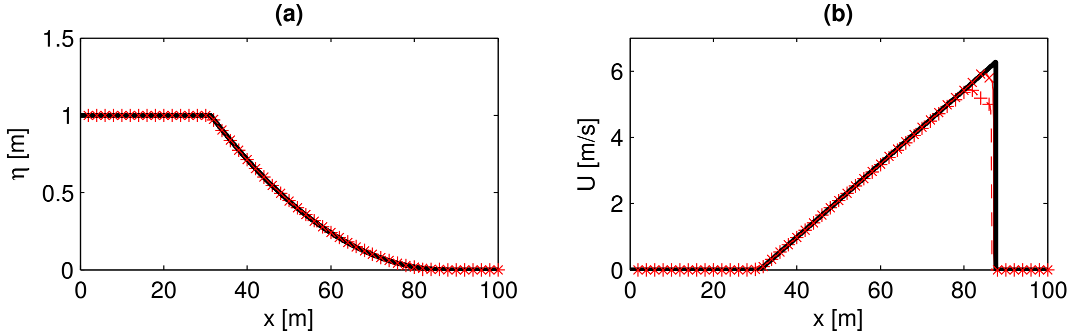


Figure 4-11 Dry dambreak case. (**a**) free surface elevation for the analytical solution (thick black line), the first order scheme (red plusses) and the second order scheme (red crosses). (**b**) horizontal velocity for the analytical solution (thick black line), the first order scheme (red plusses) and the second order scheme (red crosses).

The depth averaged velocities are shown in Figure 4-11**b**. Here we clearly see differences between the two solutions. The Higher order scheme produces a sharper shock as the velocity at the peak of the wave is better resolved. Furthermore the shock propagation speed is better resolved.

It is also important to mention that no wiggles are introduced by the higher order scheme due to the use of the minmod limiter. Altough this limiter still introduces a small amount of diffusion the results are significantly better when compared to the FOU scheme.

The results of the wet bed dam-break are shown in Figure 4-12(**a-b**). Here the difference between the two methods are negligible as both produce the correct discontinuity in both the horizontal velocities and the free surface elevations. Again no wiggles are observed for the higher order scheme. More important is that the McCormack scheme also captures the correct jump height of the discontinuity. Generally non-conservative schemes have trouble to model this accurately and this is a strong indication that the implementation is indeed momentum conservative.

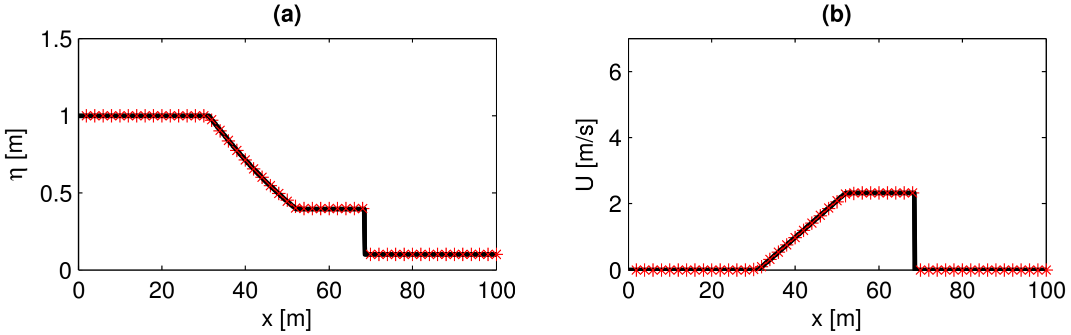


Figure 4-12 Wet dam-break case. (**a**) free surface elevation for the analytical solution (thick black line), the first order scheme (red plusses) and the second order scheme (red crosses). (**b**) horizontal velocity for the analytical solution (thick black line), the first order scheme (red plusses) and the second order scheme (red crosses).

4.4 Wave deformation by an elliptic shoal on sloped bottom

Introduction

The Berkhoff Shoal is a classic experiment conducted by Berkhoff, Booy et al. (1982) which combines refraction, diffraction and shoaling of waves over a complex bathymetry. It was originally setup for a comparison of laboratory measurements with linear wave propagation models but has widely been used to verify models based on Boussinesq like equations and was also discussed in Stelling and Zijlema (2003).

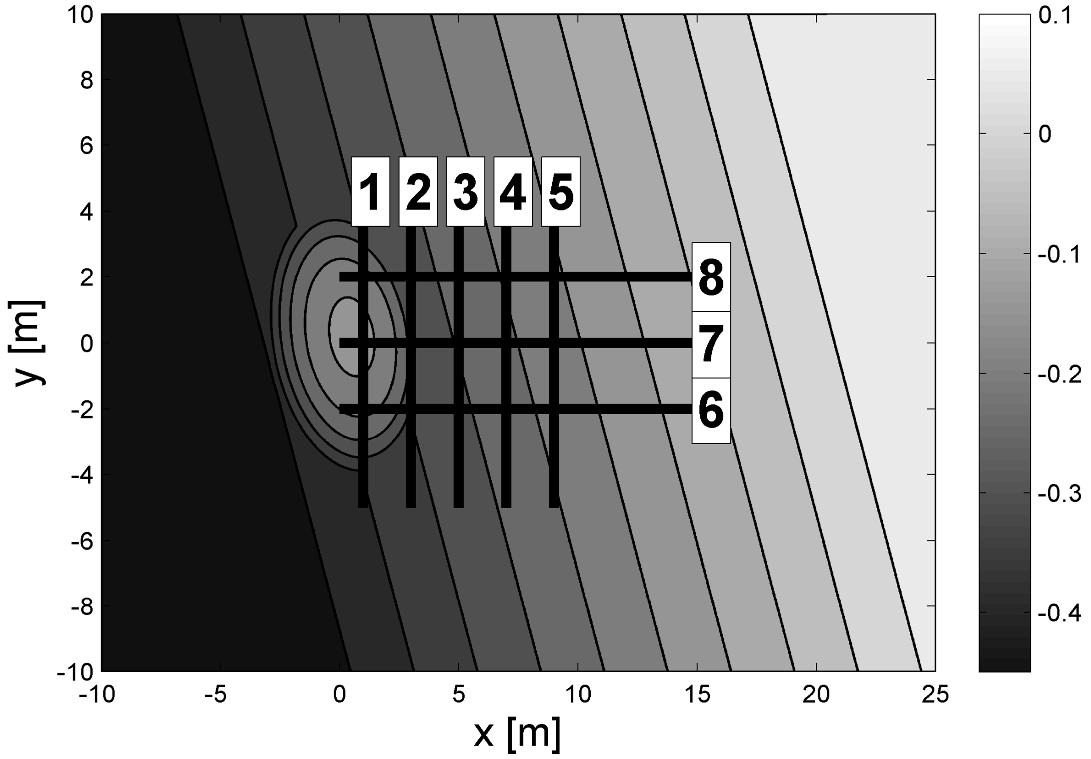


Figure 4-13 Bathymetry of the experiment carried out by Berkhoff et. al. and the location of the transects along which measurements were conducted.

The experiment was carried out in a rectangular wave tank measuring 35 by 20 metres. The bathymetry consists of an elliptic shoal located a gentle slope with straight depth contours (see Figure 4-13). A local coordinate system was used with the origin at the top of the shoal and the wavemaker located at x=-10 m. Monochromatic waves with a frequency of 1 Hz. and wave height of 4.64 cm. were generated and propagated into the domain from the wave maker to the beach. At the beach the waves broke and dissipate most of their energy.

Wave heights were measured along eight transects starting just behind the shoal. Three were placed in the x-direction and five in the y-direction. Along each transect ten wave gauges were deployed that were used to measure the wave height after a steady state was reached. The locations of each transect are shown in Figure 4-13 where they are numbered from one to eight.

In the region in front of the shoal the dominant wave processes are refraction and shoaling. These generally lead to a small increase of wave height on top of the shoal, and a small decrease in the regions at the sides of the shoal. Behind the shoal refraction focuses the waves into a narrow region and cross seas are generated. Locally the wave field has a strong variation in wave height along the crests. In the linear theory caustics are formed and this is therefore a region where the combined effect of refraction and diffraction should be taken into account.

Setup

Monochromatic waves with a frequency of 1 Hz. and wave height of 4.64 cm are generated at the boundary (*x=-10)* using depth average velocities obtained from a third order stokes solution. To avoid initial disturbances the wave height is slowly increased using a ramp function identical to the one used in Stelling and Zijlema (2003). Furthermore to allow the reflected long waves to leave the domain use is made of the non-reflective boundary condition as described in section 3.3. Both of the lateral boundaries (at *y=*-10 and *y=*10) are assumed to be fully reflective closed boundaries (*v=0*).

Use is made of a variable mesh size to allow for high detail in the region near the shoal. On top of the shoal $\Delta x = \Delta y = 0.025m$ which means that there are locally roughly sixty points per wavelength. In the regions in front and to the sides of the shoal the mesh size was increased to $\Delta x = \Delta y = 0.004m$ while at the back of the shoal $\Delta x = 0.06m$. To avoid sudden variations in the mesh size a transitional region was used where the grid smoothly varied in size using an expansion parameter of $r = 1.05$. (See section 3.1). This resulted in a computational grid of 990×650 cells.

The simulation time was set to sixty seconds and after roughly thirty seconds the steady state was reached. The free surface elevation was retrieved along the same transects as in the original experiment for the final thirty seconds. Average wave heights where obtained from these free surface time series using a zero crossing analysis and averaging over the number of waves in the record.

Results

In Figure 4-14 the results from the model along transects indicated in Figure 4-13 are shown and compared to the measurements. Along all transects the model results compare very favourably with the measurements. Only at transect 5 (*x=9*) the large peak due to the focussing of the waves is significantly under predicted. The present results for the depth averaged model are even better than those found in Smit (2008). This is most likely due to the higher resolution used in the present setup and the use of the more accurate McCormack scheme. The most impressive improvement in comparison to those results is found in section eight (*y=2)*. The minimum around *x=8 m* is now reproduced while this was absent in the previous study.

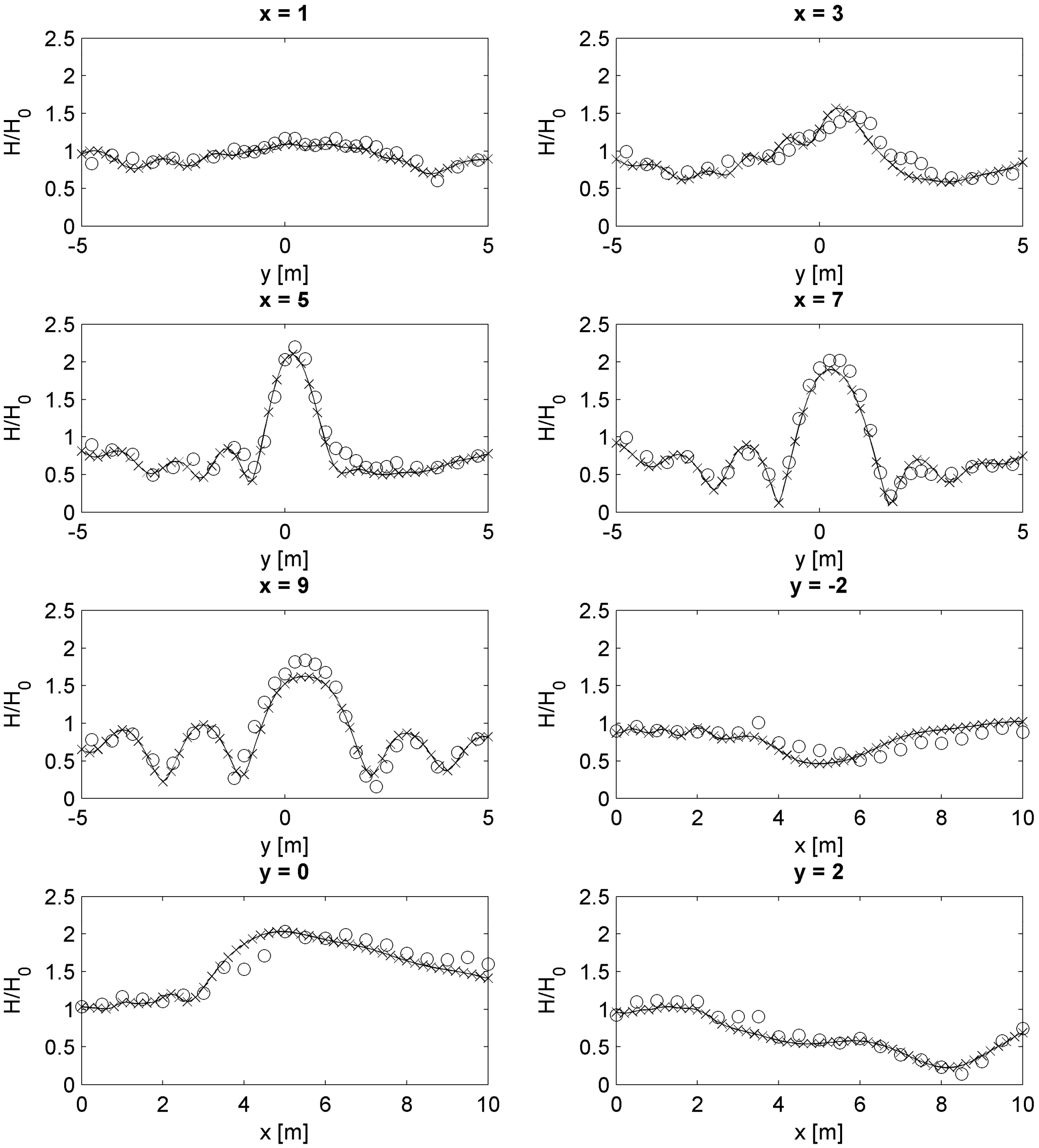


Figure 4-14 The relative wave height along the eight transects. Measurements are indicated with circles while .Comparison between measurements (symbols) and the results using a depth averaged approach. Shown are the relative wave heights along the eight transects (waves scaled with the incoming wave height).

4.5 Boers

not yet finished.

5 Implementation and usage

5.1 Introduction

Implementation approach

Xbeach has been set up in such a way that it is relatively easy to understand for those with a limited background in numeric's and programming. One of the goals in implementing the non-hydrostatic pressure correction was therefore to leave the original code base intact as much as possible. To accomplish this, the new subroutines are introduced in separate modules. The only modifications to the original code are then to introduce subroutine calls at the appropriate places that are only executed when requested by the user. Such an approach is possible because both the non-hydrostatic pressure correction and the McCormack scheme are formulated as a correction to the explicit first order equations.

To keep the code transparent new data structures, variables and procedure are kept hidden from any using units/procedures by private declaration. Only those variables/procedures that need to be accessed from other units/procedures are declared public. Unfortunately certain operations (such as output) require the introduction of new array types in the structures used for communications. These variables are included in the `spaceparams` file and the appropriate initializations statements have been added.

All of the new subroutines are disabled by default by conditional statements and can only be used by explicitly setting the correct parameter in the `params.txt` file. Therefore there is no need to adapt existing simulation setups.

Because of the used solution method the parallelisation of the non-hydrostatic code is not trivial and was not included in the current work. Thus when XBeach is compiled with MPI the subroutines handling the non-hydrostatic correction and the second order corrections are disabled using the pre-processor. This avoids potential problems when parallel jobs are run.

Usage

Running the non-hydrostatic model is very similar to running a default XBeach simulation. Therefore it is recommended that first the XBeach manual is studied before using the non-hydrostatic module. In section 5.3 only those steps that are specific for a non-hydrostatic simulation will be explained together with advice on recommended settings. A full description of all parameters can be found in appendix C

5.2 Program structure

Main program

Most of the modifications and additions are to the flow sub-module and this means that hardly any modifications are necessary to the main program structure. Thus the main time stepping loop, the number and order in which the various physical process (e.g. suspended transport, wave action balance) are computed is left unchanged. The only change is that the wave timestep subroutine is disabled for non-hydrostatic computations as the short wave field is now resolved in the flow module (see Figure 5-1). Note that the wave boundary condition subroutine is still used to provide the boundary conditions for the short waves and is therefore not disabled.

flow module

Broadly speaking the original flow module consisted of two different sub-processes: (i) updating the momentum equations and (ii) updating the mass balance. Because of the chosen formulation for the non-hydrostatic pressure and the second order corrections these main sub-processes are unmodified. Instead, both the updating of the momentum and mass equations serve as first order predictions which may optionally be corrected for second order accuracy and non-hydrostatic pressures. Thus the relative order is maintained but several other sub-processes are introduced.

In the modified flow module again first the flooding and drying and the updating of the momentum equations is performed which serves as a predictor of the depth averaged velocities. After this the pressure is explicitly included in the momentum equation to improve the predicted value when the non-hydrostatic pressure is enabled. The predicted value is subsequently corrected using an adapted McCormack scheme for the advection.

For non-hydrostatic computations velocity is corrected with the difference between the new value of the pressure and the old value used in the predictor. To do this first the linear system is constructed which is subsequently solved in an iterative manner. The resulting pressure differences are used to correct velocities.

Finally the mass balance is updated; this gives approximate values for the free surface at the new timestep. These are either accepted or, when higher order accuracy is required, corrected using again a variant of the McCormack scheme.

A flow chart that describes the modified flow module is given in Figure 5-1. Notice that the non-hydrostatic and second order corrections can be activated independently from each other. Furthermore, if both are disabled the original flow module structure is retrieved.

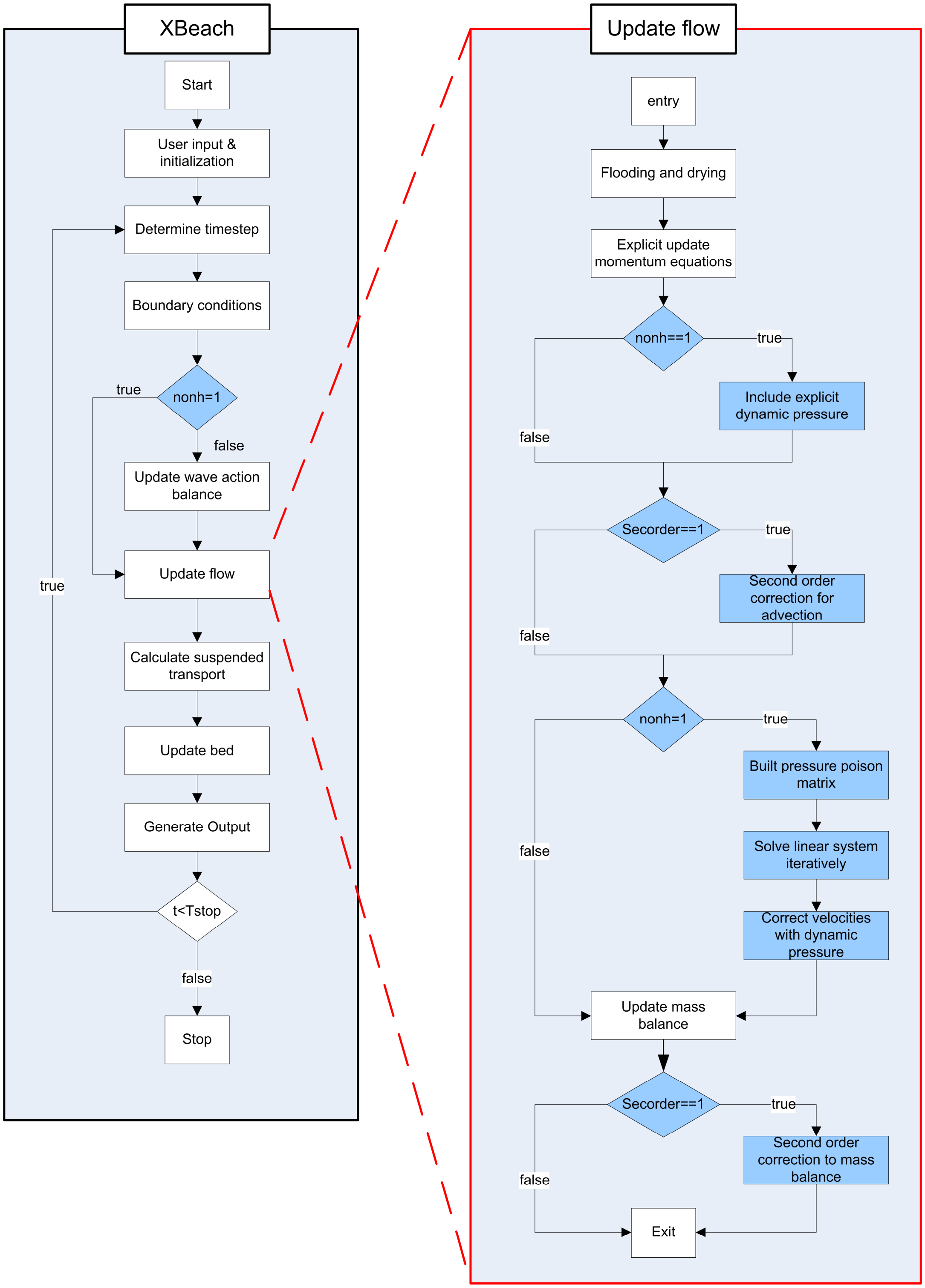

XBeach
Start
User input & initialization
Determine timestep
Boundary conditions
nonh=1
true
false
Update wave action balance
Update flow
true
Calculate suspended transport
Update bed
Generate Output
t<Tstop
false
Stop
Update flow
entry
Flooding and drying
Explicit update momentum equations
nonh==1
true
Include explicit dynamic pressure
false
Secorder==1
true
Second order correction for advection
false
nonh=1
true
Built pressure poison matrix
Solve linear system iteratively
Correct velocities with dynamic pressure
false
Update mass balance
Secorder==1
true
Second order correction to mass balance
false
Exit

Figure 5-1 Flowchart showing the structure of the main program on the left and a detailed overview of the flow module on the right. Note that processes already described in XBeach are denoted as white boxes while new processes are in blue.

5.3 Running the model

Enabling the non-hydrostatic module

To perform a non-hydrostatic calculation the non-hydrostatic sub-module needs to be enabled in the XBeach command file. This can simply be done by setting `nonh=1` in the `params.txt` file. This in principle is enough to change a default XBeach computation into a non-hydrostatic computation. There are however some extra limitations to the non-hydrostatic module.

Because of the conceptual difference between default XBeach (a long wave module forced by a short wave model) and the non-hydrostatic model (all scales of motion resolved in the flow module) it is recommended not to use the short wave action balance in combination with the non-hydrostatic module. This can either be achieved by sending in zero short wave energy at the boundaries or by manually disabling the `wave_timestep` subroutine in the main program source code.

Furthermore it is important to remember that if the program has been compiled for parallel usage the non-hydrostatic module is always disabled. Thus it is not possible to run parallel jobs with the non-hydrostatic sub-module enabled. This is mainly due to the iterative solver used and this might change in the future.

Boundary conditions

Presently the non-hydrostatic module can only be forced using either free surface or velocity time series at the boundary. Thus it is not possible to provide a (parameterised) short wave spectrum at the boundary. Therefore if the non-hydrostatic module is used either `instat=3` or `instat=8` has to be used.

To force the module using water levels the user should use `instat=3` and provide a long wave time series. It is important to remember that the short wave energy should be set to zero while the short wave motion is included into the long wave signal. This wave signal is then used to estimate the depth averaged velocity at the boundary. Because the depth averaged velocity is estimated using long wave theory this approach is only recommended for fairly long waves for which $c \approx \sqrt{gH}$ is a good approximation. For shorter waves this assumption is no longer valid and this generally results in a local distortion of the wave profile (see Van Reeuwijk, 2002).

The second method to force the model is to directly prescribe the depth averaged velocity at the boundary. This is the recommended procedure as this generally leads to the least amount of distortions. In this case the user should use `instat=8` and provide a file `boun_U.bcf` containing the velocity time series (see appendix E for a description of the file format).

In principle any of the non-reflective formulations (e.g set `front` to 0,1 or 4) can be used with the non-hydrostatic model, but experience has shown that some perform better than others. If both the incident and reflected waves are relatively long at the boundary ($kH \ll 1$) then all of the formulations perform similarly as they would in default XBeach. In this case the non-reflective boundary condition by Dongeren and Svendsen (1997) is the best choice.

However, for a typical non-hydrostatic computation the incident wave field will also contain short wave components and the assumption that both the incident and reflected waves are shallow water waves is no longer valid. In this case non-reflective formulation based on the radiating condition (`front = 0`) and the formulation based on Dongeren and Svendsen (1997) (`front = 1`) will again introduce distortions for the incident wave signal. Therefore, when performing non-hydrostatic computations, the use of the boundary condition as described in section 3.3 is recommended.

To enable this formulation the user should first set `front = 4` and activate the reflection compensation by setting `arc = 1` (note that if arc = 0 the boundary will again be fully reflective). Furthermore the user needs to provide both the time evolution of the velocity and the free surface at the boundary in the file `boun_U.bcf`. For this reason this non-reflective boundary can only be used in combination with a velocity forcing at the boundary `(instat = 8).`

Numerical options

When using the non-hydrostatic module the short wave motion is included in the flow module. The consequence of this is that the mesh sizes needs to be a fraction of the shortest wave length that is to be modelled. In default XBeach the number of meshes per wave length is generally large and this justifies the use of simple FOU approximations (generally the length scales of the other physical processes are much smaller than those of the long wave motion). However for non-hydrostatic computations the mesh size is limited by the short wave motion and experience has shown that, especially for non-linear motion, the FOU approximations introduce excessive numerical damping. This was the main motivation for the introduction of the second order McCormack scheme and it is therefore always recommended to use the McCormack scheme for non-hydrostatic computations (set `secorder = 1`). Usually when using the second order scheme a resolution of around thirty meshes per wave length suffices for accurate computations.

For one dimensional simulations (`ny=2`) the use of the tri-diagonal solver is recommended `(set solver=2)`. This solver is substantially faster than the SIP solver and more accurate as it is a direct solver. For two dimensional simulations (`ny>2`) the SIP solver has to be used `(set solver=1)`.When using the iterative solver initially the default settings are recommended as these generally work for most situation (see also section 3.5).

Appendices

A. Linear dispersion

In order to derive the dispersion relation it is assumed that the initial condition is a small disturbance in which case the non-linear terms can be neglected. Furthermore the bottom is assumed to be flat. In this case the Euler equations are simplified to:

$$\frac{\partial \zeta}{\partial t} + H\frac{\partial u}{\partial x} + U\frac{\partial H}{\partial x} = 0 \tag{C.1}$$

$$\frac{\partial u}{\partial t} + g\frac{\partial \zeta}{\partial x} + \frac{\partial p}{\partial x} = 0 \tag{C.2}$$

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \tag{C.3}$$

$$\frac{\partial w}{\partial t} + \frac{\partial p}{\partial z} = 0 \tag{C.4}$$

Notice that all advective terms have dropped out. Now a semi-discretisation is performed in the vertical only. It is assumed that there is an equidistant layer distribution with layer thickness $h = \frac{H}{k}$. In this case the equations become:

$$\frac{\partial \zeta}{\partial t} + \sum_{k=1}^{k_{\max}} h\frac{\partial u_k}{\partial x} = 0 \tag{C.5}$$

$$\left.\frac{\partial u}{\partial t}\right|_k + g\frac{\partial \zeta}{\partial x} + \tfrac{1}{2}\left.\frac{\partial p}{\partial x}\right|_{k+\frac{1}{2}} + \tfrac{1}{2}\left.\frac{\partial p}{\partial x}\right|_{k-\frac{1}{2}} = 0 \tag{C.6}$$

$$\left.\frac{\partial u}{\partial x}\right|_k + \frac{w_{k+\frac{1}{2}} - w_{k-\frac{1}{2}}}{H} = 0 \tag{C.7}$$

$$\left.\frac{\partial w}{\partial t}\right|_{k+\frac{1}{2}} + \left.\frac{\partial w}{\partial t}\right|_{k-\frac{1}{2}} + 2\frac{p_{k+\frac{1}{2}} - p_{k-\frac{1}{2}}}{h} = 0 \tag{C.8}$$

For the vertical momentum equation the Keller box has been employed instead of the compact scheme. But as mentioned before, these are essentially equivalent when advection terms are ignored. As there are k momentum and local continuity equations the total number of equations becomes $n = 3k + 1$.
Substituting for each of the variables a single fourier mode with different amplitudes and phases. As a linear superposition different fourier modes will also be a solution this does not mean a loss of generality.

$$\zeta(x,t) = \hat{\zeta}e^{i(kx-\omega t)} \qquad u_k(x,t) = \hat{u}e^{i(kx-\omega t)} \qquad w_k(x,t) = \hat{w}e^{i(kx-\omega t)} \qquad p_k(x,t) = \hat{p}e^{i(kx-\omega t)} \tag{C.9}$$

In order to simplify the expression the amplitudes are taken to be complex and therefore include the phase differences. Using the relations in (C.9) the system of equations (C.5)-(C.8) result into:

$$
\begin{aligned}
&-i\omega\zeta + \sum_{k=1}^{k_{\max}} ihku_k = 0 \\
&-i\omega u_k + igk\zeta + \tfrac{1}{2} ikp_{k+\frac{1}{2}} + \tfrac{1}{2} ikp_{k-\frac{1}{2}} = 0 \\
&iku_k + \frac{w_{k+\frac{1}{2}} - w_{k-\frac{1}{2}}}{H} = 0 \\
&-i\omega w_{k+\frac{1}{2}} + -i\omega w_{k-\frac{1}{2}} + 2\frac{p_{k+\frac{1}{2}} - p_{k-\frac{1}{2}}}{h} = 0
\end{aligned}
\tag{C.10}
$$

With $p_{k_{\max}+\frac{1}{2}} = 0$ and $w_{\frac{1}{2}} = 0$ by virtue of the kinematic boundary condition at the bottom. The equations are written in matrix notation as:

$$\mathbf{Ax} = \mathbf{0} \tag{C.11}$$

Where

$$
\mathbf{A} = \begin{pmatrix} -i\omega & ikH & & \\ ikg & -i\omega & & \tfrac{1}{2} ik \\ & ik & H^{-1} & \\ & & -i\omega & -2H^{-2} \end{pmatrix} \quad \mathbf{x} = e^{i(kx-\omega t)} \begin{pmatrix} \hat{\zeta} \\ \hat{u}_1 \\ \hat{w}_{1\frac{1}{2}} \\ \hat{p}_{\frac{1}{2}} \end{pmatrix}
\tag{C.12}
$$

In order for the system of equations in (C.11) to have more than the trivial solution $\mathbf{x} = \mathbf{0}$ the determinant of the matrix $\mathbf{A}$ has to be equal to zero thus:

$$
Det\left(\mathbf{A}\right) = \begin{vmatrix} -i\omega & ikH & & \\ ikg & -i\omega & & \tfrac{1}{2} ik \\ & ik & H^{-1} & \\ & & -i\omega & -2H^{-2} \end{vmatrix} = 0
\tag{C.13}
$$

A cofactor expansion of the determinant of the matrix yields:

$$Det\left(\mathbf{A}\right) = 2\frac{\omega^2}{H^2} - \frac{k^2\omega^2}{2} - 2g\frac{k^2}{H} = 0 \tag{C.14}$$

This expression is precisely zero if the following equation holds:

$$\omega = k\sqrt{\frac{gH}{1 + \frac{1}{4}\left(kH\right)^2}} \tag{C.15}$$

This relation between ω and k is the dispersion relation present in the case of a single computational layer. The group velocity in this case is:

$$c_g \equiv \frac{d\omega}{dk} = \sqrt{\frac{gH}{1 + \frac{1}{4}\left(kH\right)^2}}\left[1 - \frac{1}{1 + 4\left(kH\right)^{-2}}\right] \tag{C.16}$$

B. Procedure description

flow_secondorder_init

Called by	:	flow_secorder_con (only at start)	*File*	:	flow_secorder.F90
Calls	:	none	*Module*	:	flow_secorder_module

Purpose Allocates and initializes the necessary variable for the second order corrections. Is only called at the start of the simulation.

flow_secondorder_con

Called by	:	flow_timestep	*File*	:	flow_secorder.F90
Calls	:	minmod	*Module*	:	flow_secorder_module

Purpose Performs the second order correction for the continuity equation. Not called when sec_order=1.

flow_secondorder_advUV

Called by	:	flow_timestep	*File*	:	flow_secorder.F90
Calls	:	minmod	*Module*	:	flow_secorder_module

Purpose Performs the second order correction for the advection in the U- and V- momentum equations. Not called when sec_order=1

minmod

Called by	:	flow_secondorder_con flow_secondorder_UV flow_secondorder_W	*File*	:	nonh.F90
Calls	:	none	*Module*	:	nonh_module

Purpose Implementation of the minmod limiter. Responsible for alternating between the second and first order accurate schemes.

flow_secondorder_advW

Called by	:	nonh_explicit	*File*	:	flow_secorder.F90
Calls	:	minmod	*Module*	:	flow_secorder_module

Purpose Performs the second order correction for the advection in the W- momentum equation. Not called when sec_order=1

nonh_cor

Called by	:	flow_timestep	*File*	:	nonh.F90
Calls	:	solver_solvemat	*Module*	:	nonh_module

Purpose This is the main procedure that builds the discrete pressure poisson matrix. After the matrix is built the solver routine is called which returns the hydrodynamic pressures. With the now known hydrodynamic pressures the velocities are corrected. Only called when nonh=1

nonh_explicit

Called by	:	flow_timestep	*File*	:	nonh.F90
Calls	:	flow_secondorder_W	*Module*	:	nonh_module

Purpose This routine incorporates the hydrodynamic pressure in the predicted value of the flow field using the old pressure. Furthermore it solves the explicit part of the depth averaged vertical momentum equation. Only called when nonh=1

Nonh_init

Called by	:	nonh_explicit	*File*	:	nonh.F90
Calls	:	none	*Module*	:	nonh_module

Purpose Allocates and initializes the necessary variable for the non-hydrostatic correction. Is only called at the start of the simulation when nonh=1.

solver_init

Called by	:	solver_solvemat	*File*	:	solver.F90
Calls	:	none	*Module*	:	solver_module

Purpose Initializes the resources needed for the various matrix solvers.

solver_solvemat

Called by	:	nonh_cor	*File*	:	solver.F90
Calls	:	solver_tridiag solver_init solver_sip	*Module*	:	solver_module

Purpose Subroutine that calls the correct solver based on the user input (solver =1/2).

solver_tridiag

Called by	:	solver_solvemat	*File*	:	solver.F90
Calls	:	none	*Module*	:	solver_module

Purpose Solves the linear system using the Thomas algorithm. Only apllicble for 1d simulations. Onlycalled when solver =2

solver_sip

Called by	:	solver_solvemat	*File*	:	solver.F90
Calls	:	none	*Module*	:	solver_module

zurpose Solves the linear system using Stone's Implicit procedure. Only called when solver=1

velocity_Boundary

Called by	:	flow_bc	*File*	:	boundaryconditions.F90

Calls : velocity_boundary_read *Module* : boundaryconditions

Purpose Returns the current value of the velocity/surface elevation at the boundary. The values are read from a text file using velocity_boundary_read. Procedure is only called when instat=8.

velocity_Boundary_read

Called by : velocity_boundary *File* : boundaryconditions.F90

Calls : none *Module* : boundaryconditions

Purpose Reads a new line from the boundarycondition file and returns the appropriate variables.

visc_smagorinsky

Called by : flow_timestep *File* : flow_timestep.F90

Calls : none *Module* : flow_timestep_module

Purpose Calculates the value of the eddy viscosity based on a Smagorinsky-type subgrid model.

C. Input parameters

This section briefly describes the additional/modified parameters that where included in the course of this study. For a full description of all parameters and how to set them the reader is referred to the XBeach manual (Roelvink et al., 2009).

Boundary conditions

Front	Controls the type of boundary condition used in the flow module. **0-3** *See the XBeach manual.* **4** *Radiating boundary condition as described in 3.3. The user needs to provide a file containing time series for the velocity at the boundary. If [Arc=0] the boundary reduces to a von Neumann boundary.*
instat	Controls the way the model is forced. Reference is made to the XBeach manual when the non-hydrostatic module is not used. **0–7** *See the XBeach manual* **8** *The model is forced using a time series for the velocity on the boundary. See appendix C for a description.*

non-hydrostatic model

nonh	Controls whether or not the non-hydrostatic corrections are enabled. When MPI is used the non-hydrostatic corrections are always disabled irrespective of the value of nonh. The default value is zero (disabled). **0** *Non-hydrostatic corrections disabled* **1** *Non-hydrostatic corrections enabled.*
dispc	Coefficient to optimize the dispersion relation (see 4.1). Only used when the non-hydrostatic options are enabled. The default value is one. **> 0.0** *Constant value over the entire domain.* **< 0.0** *Value is optimized for the local depth and a user defined (peak) period (see Topt).*
Topt	(Peak) period in seconds. Used to locally optimize the numerical dispersion relation. Only used when dispc<0. The default value is 10 s.

sub-grid model

smag	Controls if the Smagorinsky sub-grid model (see section 2.3). is used for the calculation of the eddy viscosity. When MPI is used the Smagorinsky sub-grid model is always disabled irrespective of the value of smag. The default value is zero (disabled). **0** *Smagorinsky sub-grid model disabled* **1** *Smagorinsky sub-grid model enabled.*
nuh	If the Smagorinsky sub-grid model is enabled this sets the value of the Smagorinsky constant C_s (see section 2.3). If the Smagorinsky sub-grid model is disabled this is the value for the constant background viscosity.

Numerics

secorder	Controls whether or not the higher order scheme for the advection is used. When MPI is used the second order corrections are always disabled irrespective of the value of secorder. The default value is zero (disabled). **0** *Higher order corrections turned off.* **1** *Higher order corrections enabled.*
kdmin	Removes points from the pressure matrix if the shortest wave length ($L = 2\Delta x$) has a *kH* wave shortness smaller than *kdmin*. See also section 3.4. The default value is 0.0 (no points are removed).
solver_urelax	Under-relaxation parameter α in the SIP solver. From experience it is shown that the SIP solver generally converges for $\alpha \leq 0.92$. Generally, the higher the value of α the less iterations are needed. (section 3.5.) The default value is α is 0.92
solver	Chooses the matrix solver used to solve the linear system. For one dimensional simulations (*ny=2)* the tri-diagonal solver is recommended. The default value is 1 (Sip Solver). **1** *Sip solver* **2** *Tri-diagonal solver (only applicable for one dimensional simulations)*
solver_acc	Stop criterion for the SIP solver. Used to terminate the iteration process if the requested accuracy has been reached (section 3.5.). The default value is 0.005 [-].
solver_maxit	Maximum number of iterations performed in the SIP solver. If the maximum number of iterations is reached the iteration process is stopped, irrespective whether or not the requested accuracy has been reached (section 3.5.). The default value is 20 [-]

D. strongly implicit procedure

The basis for the SIP method lies in the observation that an LU decomposition is an excellent general purpose solver, which unfortunately cannot take advantage of the sparseness of a matrix. Secondly, in an iterative method, if the matrix **M** is a good approximation to the pressure coefficient matrix **A**, rapid convergence results. These observations lead to the idea of using an approximate LU factorization of **A** as the iteration matrix **M**. i.e.:

$$\mathbf{M} = \mathbf{LU} = \mathbf{A} + \mathbf{N} \tag{3.17}$$

Where **L** and **U** are both sparse and **N** is small. For asymmetric matrices the incomplete LU (ILU) factorisation gives such an decomposition but unfortunately converges rather slowly. In the ILU method one proceeds as in a standard LU decomposition. However, for every element of the original matrix **A** that is zero the corresponding elements in **L** or **U** is set to zero. This means that the product of **LU** will contain more nonzero diagonals that the original matrix **A.** Therefore the matrix **N** must contain these extra diagonals as well if (3.17) is to hold.

Stone reasoned that if the equations approximate an elliptic partial differential equation the solution can be expected to be smooth. This means that the pressure points corresponding to the extra diagonals can be approximated by interpolation of the surrounding points. By allowing **N** to have more non zero entries on all seven diagonals and using the interpolation mentioned above the SIP method constructs an **LU** factorization with the property that for a given approximate solution ϕ the product $\mathbf{N}\phi \approx 0$ and thus the iteration matrix **M** is close to **A** by relation (3.17). To solve the system of equations the following iterations is performed, starting with an initial guess for the pressure vector $\mathbf{p}^s$ iteration is performed solving:

$$\mathbf{U}\mathbf{p}^{s+1} = \mathbf{L}^{-1}\mathbf{N}\mathbf{p}^{s} + \mathbf{L}^{-1}\mathbf{Q} \tag{3.18}$$

Since the matrix **U** is upper triangular this equation is efficiently solved by back substitution. An essential property which makes the method feasible is that the matrix **L** is easily invertible. This iterative process is repeated until convergence is reached. Note that when convergence is reached the solution is identical to the original problem.

E. Boundary file description

When specifying an incoming short wave signal using `par%instat = 8` a text file specifying the variation in time of the orthogonal velocity at the boundary has to be provided. Furthermore if `par%arc = 1` and `par%front = 4` also the free surface elevation needs to be specified. To do this a text file named `Boun_U.bcf` containing this information needs to be placed in the same directory as the `params.txt` file.

The structure of the file is shown below. Note that terms between brackets indicate options.

File: Boun_U.bcf

[scalar/vector]
[number of variables]
[variables: t,U,Zs,W]

$$\left.\begin{array}{cccccccccc} t^0 & U_1^0 & \cdots & U_J^0 & \eta_1^0 & \cdots & \eta_J^0 & W_1^0 & \cdots & W_J^0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots & & \vdots \\ t^n & U_1^N & \cdots & U_J^N & \eta_1^N & \cdots & \eta_J^N & W_1^N & \cdots & W_J^N \end{array}\right\}\text{[Data]}$$

[scalar/vector]
This option indicates if the specified data is constant along the boundary, or if a separate value for all grid points along the boundary is given.

[Number of variables]
Specifies the number of variables (including time) specified. (between 2 and 4)

[Variables]
List of the variables specified. The horizontal U-velocity should always be included but specifying the free surface elevation and vertical velocity due to the short wave signal is optional. The first variable specified is always time.

[Data]
After the variables have been specified the actual data needs to be provided. This is done in a table like fashion using a space, tab or comma as a delimiter. The first column indicates the time (in seconds) at which data is provided. Note that if no data is available at a particular moment it is interpolated from the two nearest values that are known. Furthermore, if the simulation time exceeds the last time in the table, the last row is repeated indefinitely.
The second and following columns contain the value of U (m/s)/Zs (m) /W(m/s) at the specified moment. The order in which the variables are given is identical to the order they are listed under in variables.

Example 1: Scalar data

There is no variation in the lateral direction if we are trying to specify a wave that is propagating in the direction of the x-axis. In this case it suffices to specify a single value for U and Zs at a certain moment. Thus for the boundary file we would specify:

Example 1: Scalar

```
scalar
3
t Z U
0.0     0.0     0.000
1.0     0.1     0.005
2.0     0.2     0.010
3.0     0.1     0.005
4.0     0.0     0.000
```

Therefore, at time t=2, the free surface is set to $\eta_{1,j} = 0.2$ m for all j. Similarly the horizontal velocity is set to $U_{1,j} = 0.020$ m/s for all j.

Example 2: Vector data

If the velocity and/or free surface signal varies in both space and time along the boundary (for instance for obliquely incident waves) the velocity/surface elevation along the entire boundary needs to be specified at every moment in time. If we would only have 3 grid points along the boundary in the lateral y-direction the specified boundary file could read:

Example 1: Vector

```
vector
3
t U Z
0.0     0.000     0.000     0.000     0.000     0.000     0.000
1.0     0.005     0.004     0.003     0.100     0.090     0.080
2.0     0.010     0.009     0.008     0.200     0.190     0.180
3.0     0.005     0.004     0.003     0.100     0.090     0.090
4.0     0.000     0.000     0.000     0.000     0.000     0.000
```

Here we first specify the velocity as it is listed first under the variables. Furthermore a value for each point along the boundary is specified. At t=2 the velocity $U_{1,2}$ is set to 0.009 m/s and the surface elevation at the same position to $\eta_{1,2} = 0.19$ m

References

Andrews, D. G. and M. E. Mcintyre (1978). "An exact theory of nonlinear waves on a Lagrangian-mean flow." Journal of Fluid Mechanics Digital Archive **89**(04): 609-646.

Batchelor, G. K. (1967). An introduction to Fluid Dynamics. New York, Cambridge University Press.

Berkhoff, J. C. W., et al. (1982). "Verification of numerical wave propagation models for simple harmonic linear water waves." Coastal Engineering **6**: 255-279.

Boers, M. (2005). Surf zone turbulence. Delft, Delft university of technology. **Doct. Thesis**.

Casulli, V. and G. S. Stelling (1998). "Numerical Simulation of 3D Quasi-Hydrostatic, Free-Surface Flows." Journal of Hydraulic Engineering **124**(7): 678-686.

Chen, Q., et al. (1999). "Boussinesq modeling of a rip current system." J. Geophys. Res. **104**.

Chen, Q., et al. (2000). "Boussinesq Modeling of Wave Transformation, Breaking, and Runup. II: 2D." Journal of Waterway, Port, Coastal, and Ocean Engineering **126**(1): 48-56.

Dingemans, M. W. (1997). water wave propagation over uneven bottoms. Singapore, World Scientific Publishing Co. Pte. Ltd.

Dongeren, A. R. V. and I. A. Svendsen (1997). "Absorbing-Generating Boundary Condition for Shallow Water Models." Journal of Waterway, Port, Coastal, and Ocean Engineering **123**(6): 303-313.

Dressler, R. F. (1952). "Hydraulic resistence effect on the Dam-Break functions." Journal of Research of the National Bureau of Standards **49**(3): 217-225.

Fenton, J. (1972). "A ninth-order solution for the solitary wave." Journal of Fluid Mechanics Digital Archive **53**(02): 257-271.

Grimshaw, R. (1971). "The solitary wave in water of variable depth. Part 2." Journal of Fluid Mechanics Digital Archive **46**(03): 611-622.

Hibberd, S. and D. H. Peregrine (1979). "Surf and run-up on a beach: a uniform bore." Journal of Fluid Mechanics Digital Archive **95**(02): 323-345.

Hirsch, C. (2007). Numerical computation of internal & external flows. New York, John Wiley & Sons.

Holthuijsen, L. H., et al. (1989). "A prediction model for stationary, short-crested waves in shallow water with ambient currents." Coastal Engineering **13**(1): 23-54.

MacCormack, R. W. (1969). The Effect of viscosity in hypervelocity impact cratering. AIAA Hyper velocity Impact Conference. **Paper 69-354**.

Peregrine, D. H. and I. A. Svendsen (1978). Spilling breakers, bores and hydraulic jumps. ASCE Proc. 16th Int. Conf. Coast. Engineering. **Chapter 30** 540-550.

Ritter, A. (1892). "Die Fortpflanzung der Wasserwellen." Vereine Deutscher Ingenieure Zeitschrift **36**(2): 947-954.

Roelvink, D., et al. (2009). XBeach Model Description and Manual Delft, Unesco-IHE Institute for Water Education, Deltares and Delft University of Technology.

Roelvink, D., et al. (2009). "Modelling storm impacts on beaches, dunes and barrier islands." Coastal Engineering **56**(11-12): 1133-1152.

Smagorinsky, J. (1963). "General Circulation experiments with the primitive equations." Monthly Weather Review **91**(3): 99-164.

Smit, P. B. (2008). Non-hydrostatic modelling of large scale tsunamis. Delft, Delft University of technology. **M.Sc. Thesis**.

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

Stelling, G. S. and S. P. A. Duinmeijer (2003). "A staggered conservative scheme for every Froude number in rapidly varied shallow water flows." International Journal for Numerical Methods in Fluids **43**(12): 1329-1354.

Stoker, J. J. (1957). Water Waves : The Mathematical Theory with Applications. New York, Interscience Publishers.

Stone, H. L. (1968). "Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations." SIAM Journal on Numerical Analysis **5**(3): 530-558.

Svendsen, I. A. (2006). Introduction to Nearshore Hydrodynamics. Singapore, World Scientific Publishing Co Pte. Ltd.

Van Reeuwijk, M. (2002). Efficient simulation of non-hydrostatic free surface flows. Delft, Delft University of Technology. **Msc-Thesis**.

Whitham, G. B. (1955). "The Effects of Hydraulic Resistance in the Dam-Break Problem." Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences **227**(1170): 399-407.

Zijlema, M. and G. S. Stelling (2005). "Further experiences with computing non-hydrostatic free-surface flows involving water waves." INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS(48): 169–197.

Zijlema, M. and G. S. Stelling (2008). "Efficient computation of surf zone waves using the nonlinear shallow water equations with non-hydrostatic pressure." Coastal Engineering **55**(10): 780-790.