Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

A momentum conservative finite-volume scheme on a staggered z-layer grid for three-dimensional non-hydrostatic flow

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

for the degree

MASTER OF SCIENCE
in
APPLIED MATHEMATICS

by PETER VAN DER PLAS

Delft, the Netherlands
December 2009

MSc THESIS APPLIED MATHEMATICS

A momentum conservative finite-volume scheme on a staggered z-layer grid for three-dimensional non-hydrostatic flow

PETER VAN DER PLAS

Delft University of Technology

Daily supervisor	**Responsible professor**
Dr.ir. P. Wilders	Prof.dr.ir. A.W. Heemink

Other thesis committee members

Prof.dr.ir. G.S. Stelling	Ir. J.A.Th.M. van Kester
Dr.ir. M.J.A. Borsboom	

December 2009 Delft, the Netherlands

Contents

Acknowledgements

This thesis is the result of my studies at the Delft Institute of Applied Mathematics and an internship at the Deltares Institute. This project could not be realized without the help of a number of people. Therefore I am grateful to the following persons

- Mart Borsboom for sharing his insight and knowledge on the subject, for inspiring discussions on the subject of non-hydrostatic flow simulation and for always having time to answer questions
- Jan van Kester for giving helpful suggestions, for providing literature from the field and keeping an eye on the organizational aspect of the project.
- Peter Wilders for his useful suggestions with respect to this thesis
- Arnold Heemink for being my responsible professor
- Jan van Kester, Frank Platzek and Mart Borsboom for extensive debugging sessions of the existing Delft3D-FLOW code which were necessary to enable implementation of the method presented in this thesis

and all others that should be in this list. Further I like to thank all collegae at Deltares for their help and interest and making the stay at the company a pleasant one. A special thanks goes out to my family for their support during this project, to my future wife Giorgia for her patience and support, for cooking delicious meals at the busy moments of the project.

Summary

In the early 90' the research institute Deltares began the development of a non-hydrostatic extension for the existing three-dimensional shallow-water simulation software Delft3D-FLOW a modeling package which was originally intended for hydrostatic shallow-water flow.

The original non-hydrostatic flow model is based on a non-conservative finite-difference discretization of the momentum equations. A model study demonstrated improved performance of conservative discretization schemes over non-conservative scheme for strongly varying and convection-dominated flow. For non-hydrostatic flow a momentum conservative finite-volume discretization is more appropriate. A three-dimensional finite-volume scheme on staggered z-layer grids is presented with improved momentum conservation properties.

For simulations of dynamic flow the conservation properties in time of the numerical scheme become more important. Since many typical non-hydrostatic flow situations do not concern steady state calculations, also attention is paid to the momentum conservation properties of the model in time.

For strongly non-hydrostatic flow simulations the treatment of the bottom and free-surface becomes of large importance. The design of the finite-volume discretization gave the opportunity to have a renewed look on of the existing Delft3D z-layer model.

Simulations have been carried out of short waves in a basin. The results showed an improvement compared with the existing finite-difference method and brought to the attention other aspects of the existing discretization that demand improvement in order to make the model suitable for strongly non-hydrostatic, convection-dominated flow.

Chapter 1

Introduction

Developments in the last decades have led to an increased interest in non-hydrostatic flow modeling. Due to limited computer power in the past flow simulations were mainly based on hydrostatic flow models. In the last two decades computer power drastically increased, making non-hydrostatic modeling a realistic option.

The Delft3D-FLOW modeling package was originally designed for problems in which the physical setting is dominated by horizontal flow and with differences in the vertical velocity that are of minor importance. For this type of flow it is possible to assume a hydrostatic pressure distribution leading to simplifications in the flow equations to be solved. The original Delft3D-FLOW software package is based on the three-dimensional shallow-water equations using a hydrostatic pressure assumption.

In the original software the vertical direction is only incorporated for the description of the horizontal velocity profile which becomes important for transport modeling and stratified flow. Pressure variations due to vertical acceleration of the flow are not taken into account and as such the model is only suitable for hydrostatic flow.

In the nineties Deltares (formerly WL | Delft Hydraulics) started the development of a non-hydrostatic extension in order to make the model suitable also for simulations in which vertical accelerations can no longer be neglected and the hydrostatic pressure assumption becomes invalid. Research was initiated in the mid nineties, e.g. [7],[14], and focused on the implementation of non-hydrostatic extensions for the existing shallow-water modeling packages. A first fully functional non-hydrostatic extension for Delft3D-FLOW was presented by Bijvelds in 2001 [4]. The model is based on a pressure-correction method and uses an ADI time integration scheme. A σ-layer and z-layer grid variant exists, this thesis follows on the work already done for the z-layer variant. In later years several model improvements have been made of which the last was presented by Ullmann in 2008 [16]. More details about the history of non-hydrostatic modeling and the development of the non-hydrostatic module in Delft3D-FLOW can be found among others in [16], [4].

The discretization schemes used in models for shallow-water flow are primarily designed with the focus on mass-conservation. For this reason Delft3D-FLOW explores a finite-volume scheme for the discretization of the continuity equation. In regions of the flow with an almost hydrostatic pressure distribution the momentum equations can generally without any problems be discretized using non-momentum-conservative finite-difference schemes. Momentum-conservation properties become more important for convection-dominated and strongly non-hydrostatic flow.

In 2003 Stelling and Duinmeijer presented a spatially momentum-conservative finite-difference method was introduced for the shallow-water momentum equations [12] on staggered grids. This scheme was also implemented in the hydrostatic version of Delft3D-FLOW. During the present M.Sc. project Deltares initiated the implementation of this scheme for the non-hydrostatic z-layer model.

A short model study demonstrated improved performance of conservative discretization schemes over non-conservative scheme for strongly varying and convection dominated flow. For non-hydrostatic flow it seems therefore more appropriate to use a momentum conservative finite-volume discretization. Following the results of this study a three-dimensional finite-volume scheme on staggered z-layer grids is presented with improved momentum conservation properties. This does not only regard the conservation properties of the spatial discretization. For simulations of dynamic flow the conservation properties in time of the numerical scheme become more important. Since many typical non-hydrostatic flow situations do not concern steady state calculations special attention is paid to the momentum conservation properties of the model in time.

For strongly non-hydrostatic flow simulations the treatment of the bottom and free-surface becomes important. The design of the finite-volume discretization gave the opportunity to have a renewed look at the discretization of the boundary conditions in the existing Delft3D-FLOW z-layer model. The finite-volume scheme leads to an improved treatment of the boundary conditions. Secondly in order to make the non-hydrostatic model suitable for flow with strong surface dynamics technical difficulties have been resolved occuring in z-layers near the free-surface. Finding an appropriate treatment of these z-layer artifacts is crucial for obtaining a stable and accurate model for simulations with a strongly varying free-surface level.

In chapter 2 a model study is presented demonstrating the improved performance for conservative discretizations of the momentum equations and the convection term in particular. Following these observations in chapter 3 the three-dimensional finite-volume implementation is presented for the free-surface equations discretized on a staggered z-layer grid. Several ideas and observations described in chapter 2 are extended to a multi-dimensional context.

Simulations have been carried out of short waves in a basin and are presented in chapter 4. The results showed an improvement compared with the existing finite-difference method and brought to the attention other aspects of the model that need improvement in order to make it suitable for strongly non-hydrostatic, convection-dominated flow.

Several improvements needed to be made also to the existing non-hydrostatic version of Delft3D-FLOW with finite difference discretization of the momentum equations, in order to make it more suitable for non-hydrostatic flow simulations. A short analysis and description of the adaptations can be found in appendix B.

1.1 The free-surface problem

In this thesis we discuss the discretization of non-hydrostatic free-surface flow simulation problems in which the physical domain is bounded from above by the free-surface $z = \zeta(t, x, y)$ and bounded from below by the bottom $z = -d(x, y)$, for (x, y) in some horizontal domain Ω (see figure 1.1). Following the notation of [16], the location of the bottom and free-surface are measured with respect to a reference level $z = 0$ which is generally located at the mean water level. The total water depth of the solution is expressed by $h(t, x, y) = d(x, y) + \zeta(t, x, y)$.

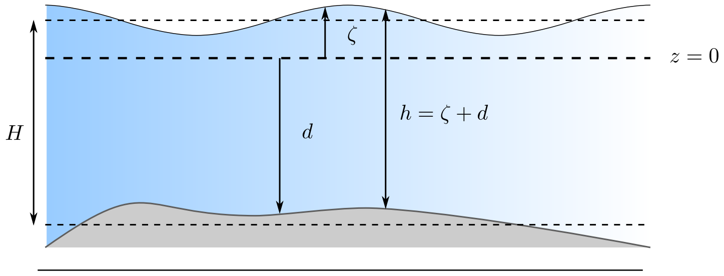


Figure 1.1: Visualization of the free-surface problem.

Later on we also use H, expressing the mean water depth.

The flow is described by the three-dimensional velocity field

$$\mathbf{u}(t, x, y, z) = \begin{pmatrix} u(t, x, y, z) \\ v(t, x, y, z) \\ w(t, x, y, z) \end{pmatrix} \tag{1.1}$$

and the pressure $p(t, x, y, z)$. The free-surface model is based on the three-dimensional Reynolds-averaged Navier-Stokes equations for incompressible flow, consisting of the continuity equation

$$\nabla\cdot \mathbf{u} = 0 \tag{1.2}$$

and the momentum equations

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla(\mathbf{u}\mathbf{u}^T) = -\frac{1}{\rho}\nabla p + \nu_h \left(\frac{\partial^2 \mathbf{u}}{\partial x^2} + \frac{\partial^2 \mathbf{u}}{\partial y^2} \right) + \nu_v \frac{\partial^2 \mathbf{u}}{\partial x^2} + \mathbf{g} \tag{1.3}$$

Here $\mathbf{g}$ describes the external body forces due to Earth's gravity, $\mathbf{g} = (0, 0, -g)$ and ν_h and ν_v denote the horizontal and vertical eddy viscosity coefficients.

The free-surface level is governed by the kinematic condition

$$\frac{\partial \zeta}{\partial t} - \mathbf{u}|_\zeta \cdot \mathbf{n}|_\zeta = 0 \tag{1.4}$$

where $\mathbf{u}|_\zeta$ and $\mathbf{n}|_\zeta$ are the velocity and normal vector at the bottom. Together the continuity equation (1.2), the momentum equations (1.3) and the kinematic surface condition (1.4) form a complete set of equations for the unknowns u, v, w, p, ζ. In this thesis these equations are referred to as the *free-surface equations*.

In order to obtain a well-posed problem the model description is completed with the necessary boundary conditions. Also at the bottom $z = -d(x, y)$ a kinematic condition is used

$$\mathbf{u}|_\zeta \cdot \mathbf{n}|_\zeta = 0 \tag{1.5a}$$

where $\mathbf{u}|_{-d}$ and $\mathbf{n}|_{-d}$ are the velocity and normal vector at the bottom. At the free-surface an atmospheric pressure condition is prescribed

$$p|_\zeta \equiv p_a \tag{1.6}$$

and at closed boundaries a Neumann condition for the pressure is applied

$$\frac{\partial p}{\partial n} = 0 \tag{1.7}$$

Sometimes it becomes usefull to distinguish between the hydrostatic and non-hydrostatic pressure components and therefore the pressure is written as

$$p \equiv p_a + \rho g \cdot (\zeta - z) + \delta p \tag{1.8}$$

where δp denotes the non-hydrostatic pressure component.

In the presentation of the equations several simplifications are made with respect to the full model equations as used by Delft3D-FLOW:

- the atmospheric pressure is set equal to zero, $p_a \equiv 0$
- the density is taken constant
- no-slip boundary conditions are used
- no turbulence model is used and the viscosity is constant in space and time
- other driving forces (e.g. bottom friction, wind stress and coriolis) are not included

1.2 Categories of flow simulation

When looking at the pressure distribution, flow situations occuring in the free-surface problem can be divided in two categories depending on the characteristic horizontal scale L and the characteristic vertical scale H (see e.g. [18]).

1.2.1 Hydrostatic flow

In case of flow situations with large horizontal length scales, L/H larger than approximately 100, horizontal flow typically dominates over vertical flow and the convection terms in the vertical momentum equation can be neglected. If furthermore it is assumed that all external body forces originate from earth's gravity the vertical momentum equation can be simplified to

$$\frac{\partial p}{\partial z} = -g$$

leading to the hydrostatic pressure assumption

$$p = p_a + \rho g(\zeta - z) \tag{1.9}$$

By depth integration of the continuity equation a condition for the evolution of the free-surface can be obtained.

$$\begin{aligned} 0 &= \int_{-d}^{\zeta} \left[\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right] dz \\ &= \frac{\partial}{\partial x} \int_{-d}^{\zeta} u dz + \frac{\partial}{\partial y} \int_{-d}^{\zeta} v dz + w|_{-d}^{\zeta} - u_\zeta \frac{\partial \zeta}{\partial x} - v_\zeta \frac{\partial \zeta}{\partial y} - u_{-d} \frac{\partial \zeta}{\partial x} - v_{-d} \frac{\partial \zeta}{\partial y} \end{aligned} \tag{1.10}$$

After substitution of the kinematic conditions (1.4) and (1.5a) at the bottom and free-surface, this leads to

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x} \int_{-d}^{\zeta} u dz + \frac{\partial}{\partial y} \int_{-d}^{\zeta} v dz = 0 \tag{1.11}$$

After substitution of the hydrostatic pressure relation, the horizontal momentum equations become

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial vu}{\partial y} + \frac{\partial wu}{\partial z} = -g \frac{\partial \zeta}{\partial x} + \nu_h \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \nu_v \frac{\partial^2 u}{\partial z^2}, \tag{1.12a}$$

$$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} + \frac{\partial wv}{\partial z} = -g \frac{\partial \zeta}{\partial x} + \nu_h \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + \nu_v \frac{\partial^2 v}{\partial z^2} \tag{1.12b}$$

Together the horizontal momentum equations (1.12a),(1.12b), the continuity equation (1.2) and the free-surface relation (1.11) form the three-dimensional shallow-water equations for the unknowns u, v, w and ζ. Originally Delft3D-FLOW was designed for simulation of hydrostatic flow based on these shallow-water equations.

By integration it is possible to fully eliminate the vertical velocity in order to obtain the two-dimensional depth-integrated shallow-water equations

$$\frac{\partial \zeta}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} = 0 \tag{1.13}$$

$$\frac{\partial hu}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial vu}{\partial y} + gh \frac{\partial \zeta}{\partial x} = 0 \tag{1.14}$$

$$\frac{\partial hv}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} + gh \frac{\partial \zeta}{\partial y} = 0 \tag{1.15}$$

For the derivation it is assumed that the vertical variation of the horizontal velocities is negligible (for background see e.g. [18],[19]). Inclusion of the vertical velocity is still essential if one is interested in vertical transport, for example in stratified flow.

1.2.2 Non-hydrostatic flow

As the difference between the characteristic horizontal and vertical length scales becomes smaller the hydrostatic pressure distribution provides a less accurate approximation. Horizontal and vertical velocities become of the same order of magnitude and due to vertical accelerations the vertical pressure distribution can no longer be assumed to be hydrostatic. For this reason the full vertical momentum equation needs to be modeled.

It is possible to distinguish between weakly non-hydrostatic and strongly non-hydrostatic flow. In case of weakly non-hydrostatic flow the pressure can be written as a hydrostatic part with a small correction δp containing the non-hydrostatic pressure component

$$p \approx g(\zeta - z) + \delta p$$

where one can still use the fact that $\delta p \ll g(\zeta - z)$. In case of strongly non-hydrostatic flow this is no longer the case and the non-hydrostatic pressure gradient can become much larger than the hydrostatic pressure gradient.

1.3 Discretization methods

1.3.1 Grid ordering

Several choices can be made for the discretization of the physical domain all with their own implications on the numerical properties and performance of the model. A distinction can be made between a staggered and collocated grid ordering. Delft3D-FLOW is discretized on a classical staggered C-grid (as used in [3]) in which the pressure and water levels are located halfway between the velocity variables. This provides numerical advantages since the velocity variables are now located in between the water levels at the location of the mass fluxes and secondly the pressure variables are centered around the velocity variables, so the pressure gradient is easily discretized using central differencing.

Finite volume methods are most often discretized on collocated grids (e.g. [10]). The use of grid staggering introduces special situations that need to be taken into account near the boundaries of the numerical domain. Different volumes are used for the different conservation laws.

The existing Delft3D-FLOW package includes the possibility of using a curvilinear grid transformation in the horizontal plane. In this way fluxes can still be discretized perpendicular to the cell faces and so-called staircase boundaries which occur in rectangular Cartesian grids. The use of a curvilinear grid transformation has however consequences for momentum conservation properties as fully described in section 3.1.

1.3.2 Z-layer versus sigma-coordinate models

For the discretization in the vertical plane, two commonly used vertical coordinate systems are the Cartesian and σ-coordinate systems. By applying a σ-coordinate transformation to the free-surface equations it is possible to discretize on a boundary fitted grid. An advantage of using this approach is that horizontal grid lines are aligned with the bottom and free-surface boundaries. Using a σ-coordinate transformation the domain is discretized on a constant number of layers of which the thickness depends on the local water level. In shallow regions the layer thickness can become very small. In the classical σ-layer grid the vertical grid lines are perpendicular to the bottom and free-surface facilitating the discretization of the boundary conditions. In this way the discretization in grid cells near the horizontal boundaries is significantly more easy. On the other hand the discretization is complicated due to the coordinate transformation of the equations and for the internal domain discretization on a Cartesian grid would be less complicated. A discussion is given in [4], chapter 5.

In z-layer grids, on the other hand, the number of layers depends on the local water depth of the solution and the layer thickness is fixed (except possibly at the free-surface. Using a Cartesian grid a larger number of grid layers is needed for obtaining the same accuracy as with σ-layer grids. In deep areas the z-layer approach is therefore computationally more expensive than the σ-layer approach, whereas in shallow areas the opposite is true.

The Delft3D-FLOW non-hydrostatic module which is under consideration in the present work is discretized on a z-layer grid. In the z-layer approach the physical domain is discretized on a Cartesian grid in which the bottom and free-surface boundaries do not coincide with the boundaries of the grid cells. A major drawback of using a discretization based on a z-layer grid is the more complicated treatment of the bottom and free-surface boundaries. For internal grid cells the z-layer approach is less complicated, however in boundary cells the σ-coordinate transformation approach has clear advantages over the z-layer approach in which accurate approximation of the boundary conditions quickly becomes more complicated and additional bookkeeping is necessary to cope with exceptions occuring when large free-surface and bottom gradients cross different z-layers. Special tricks need to be found in order to give reasonable approximations of flow around vertical sections in the bottom and in particular the free-surface. In these regions it is hardly possible to fully discretize the governing equations and necessary simplifications need to be made.

Making an appropriate choice between using σ and z-layer grids is problem dependent and knowledge of the flow field is necessary. For a detailed discussion see e.g. [4]. In the present work the equations are discretized using the existing z-layer model grid of Delft3D-FLOW.

Mixed forms In those areas where the bottom and/or free-surface remain within a single horizontal layer it is to a certain extent possible to use a mixed approach. By locally using a σ-type of transformation a more accurate treatment of the boundary conditions could be obtained. In section 3 this idea is applied at an analytical level where a simplified form of the classical sigma-transformation is used in order to derive a discretization of the boundary conditions at the bottom and free-surface while still discretizing on a z-layer grid. However it must be mentioned that this does not solve the problem of flow crossing different z-layers. For flow with strong surface dynamics the drawbacks of the z-layer model are still present.

Since the Delft3D-FLOW software was originally designed for shallow-water simulations the discretization of the bottom and free-surface was dealed with in a relatively simple way. It is interesting to search for a more accurate treatment of these boundaries and some attention will be paid to this. However, real gain can only be expected for variations staying within a single z-layer. In order to fully eliminate the problems occuring in case of bottom and free-surface gradients crossing different z-layers a logical solution would be to use a cut-cell technique.

1.3.3 Finite difference and finite-volume methods

Roughly speaking the commonly used discretization methods can be subdivided in *finite-difference methods*, *finite-volume methods* and *finite element methods*. In view of the existing non-hydrostatic model in Delft3D-FLOW we set aside the finite element approach and briefly discuss the finite-difference and finite-volume approach. Of course it is possible to think of many extended or mixed forms. In fact, the existing non-hydrostatic module of Delft3D-FLOW is itself a mixed form, in which the continuity and transport equations are discretized using a finite-volume approach and the momentum equations are discretized using finite-differences.

One of the advantages of using a finite-volume approach is the more direct approach to the discretization of the boundaries which generally fully coincide with the faces of the grid cells. This makes it possible to directly discretize the boundary conditions by means of boundary fluxes.

Secondly finite-volume methods are directly based on physical conservation properties and in this respect are superior to finite-difference methods. In regions of steep surface- or velocity-gradients and around hydraulic jumps the free-surface equations become less appropriate as approximation in continuous form. Across shocks and discontinuities the equations in fact become useless. The physical principles of mass and momentum conservation on the other hand still apply. When using a finite-volume discretization a discrete equivalent of these physical conservation laws is always fulfilled even if the approximation of the analytical conservation equations becomes less accurate (e.g. in case of coarse grids or steep gradients)

1.4 Existing numerical implementation

For reference purposes we now briefly describe the existing numerical implementation. The equations are taken from [16] which can be referred to for more details. For brevity the equations are presented in simplified form leaving out the spatial discretization and leaving out technical details. The overall solution procedure essentially remained unchanged.

The momentum equations (1.3) can be rewritten in non-conservative form by substituting $\mathbf{u}$ times the continuity equation (1.2), leading to

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \nu_h \left(\frac{\partial^2 \mathbf{u}}{\partial x^2} + \frac{\partial^2 \mathbf{u}}{\partial y^2} \right) + \nu_v \frac{\partial^2 \mathbf{u}}{\partial z^2} + \mathbf{g} \tag{1.16}$$

In the existing Delft3D-FLOW non-hydrostatic module the discretization of the momentum equations is based on this non-conservative form.

A semi-implicit time integration scheme is used for the momentum equations (1.16)

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \mathbf{u}^n\cdot \left(\frac{\partial \mathbf{u}^n}{\partial x} \quad \frac{\partial \mathbf{u}^n}{\partial y} \quad \frac{\partial \mathbf{u}^{n+1}}{\partial z} \right)^T = -\frac{1}{\rho}\nabla p^{n+\theta_p} + \nu_h \left(\frac{\partial^2 \mathbf{u}^n}{\partial x^2} + \frac{\partial^2 \mathbf{u}^n}{\partial y^2} \right) + \nu_v \frac{\partial^2 \mathbf{u}^{n+1}}{\partial z^2} + \mathbf{g} \tag{1.17}$$

and the continuity equation (1.2) is discretized at the new time level

$$\nabla\cdot \mathbf{u}^{n+1} = 0 \tag{1.18}$$

with the only exception of free-surface grid cells where a θ-scheme is used (of which the parameter is denoted by θ_ζ)

$$\nabla\cdot \mathbf{u}^{n+\theta_\zeta} = 0 \tag{1.19}$$

For the discretization of the pressure gradient a θ-scheme is used. A splitting method is applied to the momentum prediction equations. The split equations consist of the *momentum prediction equations*

$$\frac{\mathbf{u}^{n+1*} - \mathbf{u}^n}{\Delta t} + \mathbf{u}^n\cdot \left(\frac{\partial \mathbf{u}^n}{\partial x} \quad \frac{\partial \mathbf{u}^n}{\partial y} \quad \frac{\partial \mathbf{u}^{n+1*}}{\partial z} \right)^T = -\frac{1}{\rho}\nabla p^{n} + \nu_h \left(\frac{\partial^2 \mathbf{u}^n}{\partial x^2} + \frac{\partial^2 \mathbf{u}^n}{\partial y^2} \right) + \nu_v \frac{\partial^2 \mathbf{u}^{n+1*}}{\partial z^2} + \mathbf{g} \tag{1.20}$$

using time index $n + 1*$ for the predicted new velocity values, and the *momentum correction equations.*

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+1*}}{\Delta t} = -\frac{\theta_p}{\rho}\nabla(\Delta p^{n+1}) \tag{1.21}$$

with the pressure-correction variables Δp^{n+1} defined as

$$\Delta p^{n+1} = p^{n+1} - p^n \tag{1.22}$$

A splitting error is introduced due to the implicit treatment of the viscosity in the vertical direction (for a discussion see [16],pp. 22-24).

A pressure-correction method is used in which the momentum correction equations (1.21) are substituted in the continuity equation yielding the pressure-correction equations which are of the form

$$-\nabla\cdot\left[\frac{\Delta t\theta_p}{\rho}\nabla(\Delta p^{n+1})\right]=-\nabla\cdot\mathbf{u}^{n+1*}$$

After solving the pressure-correction equations the predicted solution $\mathbf{u}^{n+1*}$ is corrected to satisfy the continuity equation at the new time level.

After substitution of the discretized kinematic condition at the free-surface the continuity equation in the top grid layers contains the unknown ζ^{n+1}, which is eliminated by linearization of the pressure condition (1.6) at the new time level

$$\zeta^{n+1}-\zeta^{n}=\frac{\alpha^n_{i,j}}{g}\left(\Delta p_{i,j,k-1}+\delta p^n_{i,j,k-1}\right)+\frac{1-\alpha^n_{i,j}}{g}\left(\Delta p_{i,j,k}+\delta p^n_{i,j,k}\right) \tag{1.23}$$

In the present work we use the parameter

$$\alpha^n_{i,j}=\frac{z_k-\zeta^n_{i,j}}{z_k-z_{k-1}} \tag{1.24}$$

For further details about the spatial discretization, the linearized pressure condition and an overview of the exceptions near the boundaries we refer to [16], pp. 17-22.

The discretization of the pressure-correction equations leads to a three-dimensionally coupled system. The treatment at the free-surface introduces asymmetry in the discretized equations. The BICGSTAB method with MILU preconditioner is used for solving the system of equations[1] (see [16], pp. 25-42).

1.5 Choosing a convenient discretization scheme

The following aspects are important when making choices for the discretization of convection. The following overview is by no means exhaustive and depending on the type of application there can be other aspects that need consideration. In the rest of this work we will come back to some of these points while discussing several different numerical approaches to the discretization of the convection terms.

- *Accuracy* The scheme should not introduce too much numerical damping. In particular strongly varying, nonlinear flow problems ask for the implementation of higher-order schemes. Higher-order methods are generally less stable and an appropriate balance between robustness and accuracy needs to be found.

- *Time step restrictions etc.* Many numerical schemes are only stable under certain stability restrictions resulting in an upper bound on the time integration step. One can think of various scenarios in which these restrictions needlessly increase the amount of computational time. It is better to avoid as much as possible artifical mathematical constraints like the CFL condition. Ideally the numerical parameters of the model should

[1] In the present work the parameter for the MILU preconditioner is set to $\alpha_{\text{milu}}=0.95$ (see [16]).

primarily be restricted by physical constraints and modeling purposes (for example the desired accuracy).

Ideally if one does not want to be limited by a physically less interesting part of the model equations a solution would be to treat this part using an implicit discretization method. There are however practical reasons causing the need of numerical restrictions. For non-linear problems the design of implicit methods is more complicated and the computational load often much larger.

- *Boundary conditions*: When designing a numerical scheme it is relevant to consider the implementation of boundary conditions, both from the perspective of accuracy and stability and from the perspective of ease of implementation.

- *Stability*: It is important for a scheme to have some form of stability (e.g. linear stability, maximum principle, TVD). A numerical scheme must be consistent. In the ideal case there are stability and convergence theories available. In the non-linear and multi-dimensional case these properties can become very difficult or impossible to demonstrate theoretically.

- *Computational efficiency*: A numerical scheme needs to be computationally efficient. Parallel computation for example can highly increase the efficiency of the scheme. Conditional statements and complex problems (e.g. limiters and Riemann solvers) can severly increase the computational load. Many modern compilers provide the possibility of program optimization, for example by means of vectorization of iterations.

Reflection The primary objective of the current research was obtaining an improved discretization of the convection terms, in particular for weakly and strongly non-hydrostatic flow situations. First a choice needed to be made between using a conservative flux-formulation based on finite volumes and using a finite-difference convection scheme. A model study, set out in chapter 2, demonstrated the benefit of using a momentum-conservative discretization and therefore we have chosen to continue with the implementation of a finite-volume method. The first step towards obtaining an improved discretization of the convection terms (and for the momentum prediction equations in general) is to obtain improved momentum-conservation properties using a lower-order convection scheme.

Given the context of the existing non-hydrostatic model first an explicit discretization is used. For reasons to become clear later on, also the vertical convection, originally treated using an implicit scheme, is replaced by an explicit discretization. Explicit treatment of the vertical direction is possible if an appropriate lower bound is set on the vertical grid spacing. For the modeling purposes under consideration this does not yield strong restrictions.

In a later stage higher order accuracy as well as implicit schemes could be implemented. This is not subject of the present work.

Chapter 2

Momentum conservation; a model study

In the existing Delft3D non-hydrostatic z-layer model the momentum conservation equations are discretized using a finite-difference scheme. The importance of implementing a momentum conservative numerical scheme for convection is investigated in order to make a choice between using the existing finite-difference framework or designing a finite-volume scheme.

For accurate non-hydrostatic modeling conservation of momentum becomes an important physical aspect of the solution. Especially in convection dominated flow and solutions with steep velocity gradients the discretization of the convection can become crucial for obtaining accurate modeling results. In a non-hydrostatic, convection dominated context, the numerical scheme for the momentum equations preferably has all of the following properties:

- Robustness. The numerical scheme needs to remain stable in areas with steep velocity gradients and frequent changes in flow direction, both in the case of steady state solutions and time dependent simulations. Especially when strong non-linear effects start to be part of the solution the discretization of the convection term will have a more dominating effect on the accuracy of the model.

- Conservation properties. In convection dominated flow the physical aspect of momentum conservation becomes more important as well for robustness and for accuracy of the model. In particular for accurate simulation of surface dynamics and physically correct treatment of sudden changes in the bathymetry momentum conservation plays a crucial role.

- Boundary conditions. Since non-hydrostatic flow concerns physical settings with small length scales, the treatment of the boundary conditions will frequently have a large impact on the accuracy and correctness of the model results.

- Numerical damping. The numerical scheme should not show too much artificial dissipation, especially for wave modeling and transport simulations.

In the present chapter the focus will be on the first two properties. In chapter 3 attention is paid to the treatment of boundary conditions for the three-dimensional case. In the present work the aspect of numerical damping is of secondary importance and we will only reflect on it shortly during the discussion of the simulations in chapter 4.

In the present chapter the one-dimensional depth-integrated shallow-water equations will be used as basis for a test model. Although the model does not take into account non-hydrostatic effects, for our testing purposes it still provides a valid tool for comparison of numerical schemes. Firstly because the concept of momentum conservation is essentially not different for the hydrostatic shallow-water equations and the fully three-dimensional non-hydrostatic free-surface equations. Secondly, because the discretization of the convection term is expected to have a similar impact on the overall solution process, independent of the assumptions made about the distribution of the pressure field. We are primarily interested in the importance of the convection discretization in areas of strongly non-hydrostatic flow, this is where the convection can become of the same order as the hydrostatic pressure gradient. These situations are realized in a shallow-water context by introducing wave solutions characterized by large A/H, with A characterizing the amplitude of the wave and H (as introduced in chapter 1) characterizing the mean water depth.

A comparison of several upwind convection schemes is made, focusing on the behaviour in areas of steep velocity gradients and frequent sign changes. Most of the commonly used upwind convection schemes necessarily have a switching mechanism around points where the velocity becomes equal to zero. This can have a negative but also positive impact on the accuracy and stability properties of these methods.

As mentioned above the numerical schemes need to be applied to solutions with steep velocity gradients and frequent sign changes of the velocity. Furthermore we are interested primarily in the behaviour of the scheme for dynamic solutions in which flow not only strongly varies in space but also in time. Since no appropriate exact solution is available satisfying the research demands the comparison is performed by means of *prescribed solution forcing.*

2.1 shallow-water equations

In order to investigate the importance of momentum conservation we have a look at the non-linear one-dimensional shallow-water equations consisting of the continuity equation

$$\frac{\partial h}{\partial t} + \frac{\partial (H+\zeta)u}{\partial x} = 0 \tag{2.1}$$

and the momentum equation in conservative form given by

$$\frac{\partial q}{\partial t} + \frac{\partial q^2/h}{\partial x} + \frac{g}{2}\frac{\partial h^2}{\partial x} = gh\frac{\partial d}{\partial x} \tag{2.2}$$

where $q = hu$, or in energy-head conservative form[1] given by

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2 + g\zeta\right) = 0 \tag{2.3}$$

it is also possible to distinguish between the energy-head conservative form (2.3) and its non-conservative counterpart

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial \zeta}{\partial x} = 0 \tag{2.4}$$

The energy head is conserved along streamlines and the concept of energy head conservation is therefore not easily generalized to the three-dimensional case.

[1] It must be remarked that the equation is only energy-head conservative for steady-state solutions.

2.1.1 Dimensionless form

To assess the importance of the individual terms the equations can be written in dimensionless form which is done here for the energy-head conservative formulation

$$\frac{\partial h'}{\partial t'} + A\frac{\partial u'}{\partial x} + \frac{A}{H}\frac{\partial \zeta' u'}{\partial x'} = 0 \tag{2.5}$$

$$\frac{\partial u'}{\partial t'} + \frac{1}{2}\frac{A}{H}\frac{\partial u'^2}{\partial x'} + \frac{\partial \zeta'}{\partial x'} = 0 \tag{2.6}$$

where the dimensionless variables are based on typical dimensions of a linear wave

$$x' = kx,\ t' = c_0 kt,\ \zeta' = \zeta/A,\ h' = h/A,\ u' = u/(Ac_0/H),\ c_0^2 = gH$$

Under this wave assumption, the magnitude of the convection- and mass fluxes scales with the relative amplitude. In very shallow-water the nonlinear terms can be neglected. A useful quantity is the Froude number which indicates the importance of convection compared to the gravity-wave speed. Assuming typical dimensions as given above we see that it is proportional to the relative water depth

$$\max \mathrm{Fr} = \max \frac{|u|}{\sqrt{gh}} \sim \frac{A}{H}$$

Using the Froude number supercritical, critical and subcritical flow situations can be distinguished. In supercritical flow the velocity is larger than the propagation speed of gravity waves. The solution behaviour is dominated by convection and the flow is characterized by large velocities and low water depths. In subcritical flow the velocity is smaller than the propagation speed of gravity waves and the solution behaviour is dominated by gradients in the free-surface. At the transition between supercritical and subcritical flows a hydraulic jump occurs.

Comparison of the convection schemes will focus on the performance for large Froude numbers where both the sub- and supercritical case is considered. As soon as the relative wave amplitude approaches half the average depth flow starts to become supercritical (see figure 2.1).

2.1.2 Characteristics

Delft3D-FLOW uses the full three-dimensional Navier-Stokes equations and no characteristics decomposition is applied. Also in the numerical test model presented in this chapter, upwinding of the discretization will not be based on decomposition of the shallow-water characteristics. If true upwinding were to be achieved a characteristic decomposition of the full system of equations would be necessary, however a different approach is possible and is used in the current work. The terms in the shallow-water equations related to gravity waves are discretized using central schemes since the solution generally is a superposition of gravity waves propagating in both directions. The terms related to the flow velocity u are discretized using upwind schemes since flow velocity has only one direction.

To get an idea of correct upwinding it is useful to have a look at the characteristic decomposition of the equations at an analytical level can be useful. The equations in characteristic form are rewritten in order to distinguish between the part concerning the background velocity and the part concerning gravity waves. First this is done for the characteristic form of the momentum equation in non-conservative formulation (where $c = \sqrt{gh}$)

$$\begin{aligned}\frac{\partial}{\partial t}\begin{pmatrix}-c/h & 1\\ c/h & 1\end{pmatrix}\begin{pmatrix}h\\ u\end{pmatrix} &= -\begin{pmatrix}u-c & 0\\ 0 & u+c\end{pmatrix}\begin{pmatrix}-c/h & 1\\ c/h & 1\end{pmatrix}\frac{\partial}{\partial x}\begin{pmatrix}h\\ u\end{pmatrix}\\ &= -\left[\begin{pmatrix}u & 0\\ 0 & u\end{pmatrix}+\begin{pmatrix}-c & 0\\ 0 & +c\end{pmatrix}\right]\begin{pmatrix}-c/h & 1\\ c/h & 1\end{pmatrix}\frac{\partial}{\partial x}\begin{pmatrix}h\\ u\end{pmatrix}\end{aligned}$$

After row addition and division by 2, the original momentum equation is obtained again

$$\frac{\partial u}{\partial t} + \underbrace{u\frac{\partial u}{\partial x}}_{\text{background flow}} + \underbrace{g\frac{\partial h}{\partial x}}_{\text{gravity waves}} = 0$$

and it is seen that the convection term should be discretized using an upwind approach and the term involving the surface gradient should be discretized using a central approach.

For the conservative formulation of the momentum equation we would like to derive a similar discretization approach.

$$\begin{aligned}\begin{pmatrix}u^2-c^2 & c-u\\ c^2-u^2 & c+u\end{pmatrix}\frac{\partial}{\partial t}\begin{pmatrix}h\\ q\end{pmatrix} &= -\begin{pmatrix}u-c & 0\\ 0 & u+c\end{pmatrix}\begin{pmatrix}u^2-c^2 & c-u\\ c^2-u^2 & c+u\end{pmatrix}\frac{\partial}{\partial x}\begin{pmatrix}h\\ q\end{pmatrix}\\ &= -\left[\begin{pmatrix}u & 0\\ 0 & u\end{pmatrix}+\begin{pmatrix}-c & 0\\ 0 & c\end{pmatrix}\right]\begin{pmatrix}u^2-c^2 & c-u\\ c^2-u^2 & c+u\end{pmatrix}\frac{\partial}{\partial x}\begin{pmatrix}h\\ q\end{pmatrix}\end{aligned}$$

After row addition and division by $2c$ the original momentum equation is obtained, within which we distinguish between a background velocity part and a gravity-wave part

$$\frac{\partial q}{\partial t} + \underbrace{u\frac{\partial q}{\partial x}}_{\text{background flow}} \underbrace{+gh\frac{\partial h}{\partial x} + q\frac{\partial u}{\partial x}}_{\text{gravity waves}} = 0$$

When discretizing the momentum equation in conservative formulation preferably upwinding is used for the convection of the quantity q and a central approach for the remaining terms.

When introducing the convection schemes for this model study we will come back to the above discussion.

2.1.3 Relative importance

The relative importance of the terms in the momentum equation is shown in figure 2.1, where the comparison is made by assuming a simple wave solution and averaging across one wave length. It is seen that for waves with amplitudes smaller than approximately 50% of the water depth the momentum equation is dominated by the hydrostatic pressure gradient. For larger amplitudes the flow becomes supercritical and as the average Froude number approaches unity the convection term becomes of larger importance than the pressure gradient.

The primary focus in this chapter will be on the discretization of the convection term for flow where the relative wave amplitude is large (i.e. $A/H \gg 0$). It is in these situations that nonlinear effects in the convection term will start to play a more important role.

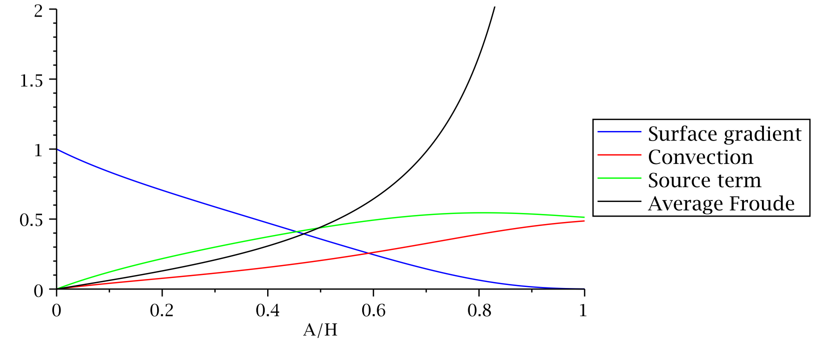


Figure 2.1: Relative importance of the convection term $u\frac{\partial u}{\partial x}$, the hydrostatic pressure gradient $g\frac{\partial \zeta}{\partial x}$ and the source term (2.15) introduced later on. The values are obtained by integrating the absolute value across one wave length. For comparison the values are normalized. The blue line describes the average Froude number.

The shallow-water equations only provide a realistic approximation for long waves ($L \gg H$). Independent of the relative wave length kH all waves propagate as shallow-water waves with a velocity of $\pm\sqrt{gh}$. If the number of grid points per wave length is kept constant, the wave length does not have effect on the relative importance of the different terms in the equations.

2.2 Conservative and non-conservative upwind schemes

Before proceeding to the comparison within the context of the shallow-water equations, in this section we discuss the differences between conservative and non-conservative discretization methods by looking at two first-order upwind schemes applied to the inviscous Burger's equation:

$$\frac{\partial u}{\partial t} = -\frac{\partial f}{\partial x}, \quad f = \frac{1}{2}u^2 \tag{2.7}$$

The Burger's equation can also be written in non-conservative form

$$\frac{\partial u}{\partial t} = -\frac{\partial f}{\partial x} = -\frac{df}{du}\frac{\partial u}{\partial x} = -u\frac{\partial u}{\partial x} \tag{2.8}$$

In the following it is assumed that the time-derivative is discretized using an Euler forward scheme. A discretization for the convection term in non-conservative form is given by the following finite-difference scheme

$$u_i\frac{u_i - u_{i-1}}{\Delta x}, \quad u_i > 0, \tag{2.9}$$

$$u_i\frac{u_{i+1} - u_i}{\Delta x}, \quad u_i < 0 \tag{2.10}$$

Taylor series analysis shows that the spatial truncation error of the scheme is then given by

$$-\frac{1}{4}\Delta x|u|\frac{\partial^2 u}{\partial x^2} + O(\Delta x^2)$$

A commonly used first-order upwind conservative discretization of the convection term is given by (see e.g. [17],[20])

$$\frac{f_{i+1/2} - f_{i-1/2}}{\Delta x} \tag{2.11}$$

where

$$f_{i+1/2} = \begin{cases} \frac{1}{2}\overline{u}_{i+1/2} u_i, & \overline{u} > 0, \\ \frac{1}{2}\overline{u}_{i+1/2} u_{i+1}, & \overline{u} < 0 \end{cases}$$

the spatial truncation error of this scheme is

$$\begin{array}{ll} -\frac{1}{4}\Delta x \frac{\partial}{\partial x}\left(u\frac{\partial u}{\partial x}\right) + O(\Delta x^2), & \overline{u}_{i-1/2} > 0, \overline{u}_{i+1/2} > 0 \\ \frac{1}{4}\Delta x \frac{\partial}{\partial x}\left(u\frac{\partial u}{\partial x}\right) + O(\Delta x^2), & \overline{u}_{i+1/2} < 0, \overline{u}_{i-1/2} < 0 \\ O(\Delta x^2), & \text{otherwise} \end{array}$$

Note that the conservation properties of the numerical schemes can also be observed in the truncation errors.

Around points where the velocity becomes equal to zero both the numerical schemes (2.9) and (2.11) have a switching mechanism. The non-conservative scheme switches when $u(x,t)$ crosses zero, whereas the conservative scheme switches when either $u(x + \Delta x/2, t)$ or $u(x - \Delta x/2, t)$ crosses zero. It is thought that the switching mechanisms around zero-points can negatively (or positively) influence the accuracy and stability of the numerical schemes.

Conservative discretizations are known to perform well around shocks and discontinuities in the flow. These properties are useful in the context of hyperbolic equations e.g. the Euler equations or also the Burger's equation. In the current work however this is less relevant since the convection scheme is to be embedded in a discretization of the three-dimensional Navier-Stokes equations and no full characteristic decomposition is used. The purpose of this chapter is therefore not to look at shock capturing techniques and Riemann problems. Especially when the solution contains steep velocity gradients it becomes important that the upwind direction is determined in a correct manner. This is best seen around points where the velocity is zero since there assuming a wrong upwind direction leads to sign changes in the numerical scheme. However it must be mentioned again that in the context of the shallow-water equations the true upwind direction of the characteristics does not directly depend on the direction of the velocity vector.

As will follow, several observations suggest that conservative schemes of the type (2.11) demonstrate a more stable upwinding behaviour.

In order to investigate the upwinding properties and overall performance the schemes (2.9) and (2.11) have been analyzed for a simple wave solution:

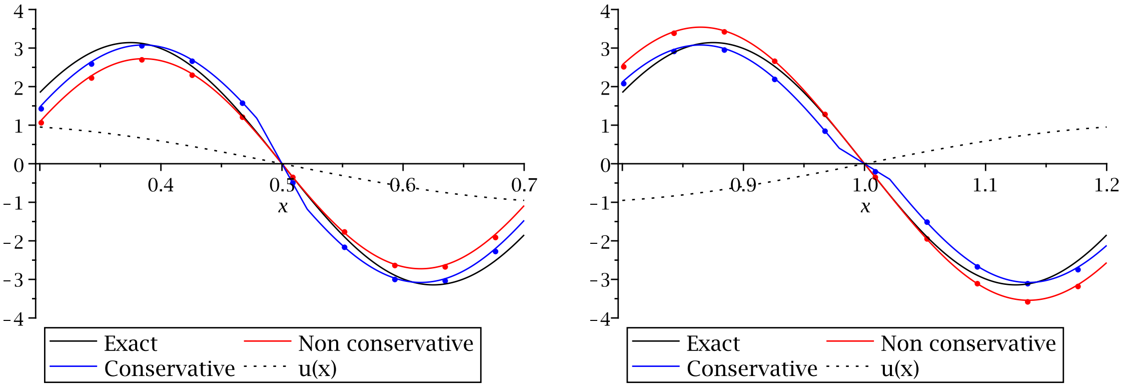
$$u(x) = \sin(2\pi x)$$

The properties of the convection schemes are compared in two respects:

(i) behaviour around points where the velocity equals zero

(ii) overall approximation of the convection term

First let us look in areas of the solution where the velocity becomes equal to zero. Situations around transonic points (i.e. where $u = 0$) can be divided in two categories: *converging flow* where in positive direction the velocity changes from $u > 0$ to $u < 0$ and *diverging flow* where in positive direction the velocity changes from $u < 0$ to $u > 0$. In fig. 2.2 the size of the convection term is plotted, showing the exact value and the approximation with the conservative and non-conservative schemes. The value of the convection term is multiplied with -1, making it equal to the increment $u^{n+1} - u^n$. In the figure it is seen that in regions of converging flow the conservative scheme gives a slight overestimation of the convection term and in regions of diverging flow an underestimation of the convection term (see also fig. 2.2). This under- and overestimation can be seen to have a sign-conserving effect on the approximation of the velocity at the new time level. Although the error of the conservative scheme is slightly larger than the error of the non-conservative scheme this has a small effect on overall accuracy since around points where $u = 0$ the solution itself is of size $O(\Delta x)$ hence both upwind schemes locally are $O(\Delta x^2)$ accurate. The slightly larger error of the conservative scheme can however have a stabilizing effect since it has a sign-conserving effect on the numerical solution.



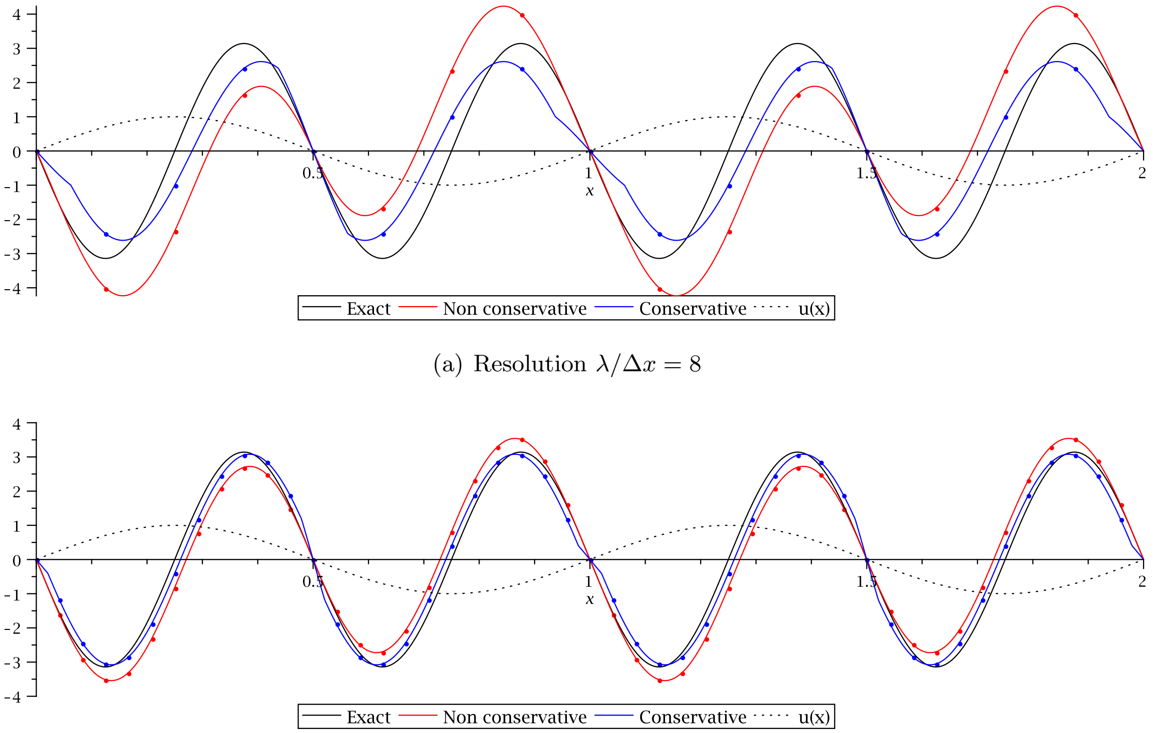
(a) Converging flow

(b) Diverging flow

Figure 2.2: The conservative scheme results in an overestimation of the convection in areas of converging flow and in underestimation in areas of diverging flow.

Now let us consider the overall performance of the convection schemes. Figure 2.3 shows the approximation of the convection scheme for a grid resolution of 8 and 24 points per wave length λ. The results indicate that a sufficiently large grid resolution is necessary to capture the second-order harmonics of the convection term. The conservative convection scheme gives a better approximation of the wave shape of the convection even for small grid resolutions (figure 2.2) whereas the non-conservative scheme over- and underestimates the convection in a much larger degree than the conservation scheme. Only in areas where the velocity approaches zero the non-conservative scheme gives a more accurate approximation than the conservative scheme.

Altogether it can be concluded that the conservative scheme exhibits a better overall accuracy and only in areas where the velocity approaches zero it becomes less accurate



(a) Resolution $\lambda/\Delta x = 8$

(b) Resolution $\lambda/\Delta x = 24$

Figure 2.3: The conservative and non-conservative scheme compared with the exact convection term for the solution $u(x) = \sin(2\pi x)$. The conservative scheme gives a better approximation of the wave shape of the convection term even for small grid resolutions. The conservative scheme clearly shows discontinuities located at $\pm\Delta x/2$ distance from the solutions of $u = 0$.

2.3 Prescribed solution forcing

To compare conservative and non-conservative convection schemes a solution is substituted in the 1D-nonlinear shallow-water equations. A wave solution above a flat bottom ($d \equiv H$) is considered, which fully satisfies the continuity equation:

$$\zeta(x,t) = A\cos(k(x - c_0 t)), \quad q(x,t) = q_0 + c_0\zeta(x,t), \quad q_0 = Hu_0, \quad c_0 = u_0 + \sqrt{gH} \tag{2.12}$$

Here H and u_0 are some reference values for depth and background velocity respectively. We have taken $H = 1$ and $u_0 = 0$. The solution approximates the exact solution $u(x,t) = u_0 + \frac{c_0}{H}\zeta$ of the linearized shallow-water equations. In fact we see that

$$u(x,t) = \frac{q(x,t)}{h(x,t)} = u_0 + \frac{c_0}{H}\zeta(x,t) - q(x,t)\left[\frac{\zeta(x,t)}{H^2} - \frac{\zeta^2(x,t)}{H^3} + \dots\right] \tag{2.13}$$

however for $A/H \gg 0$ the higher harmonic terms become larger and linear wave analysis no longer provides a good approximation of the solution behaviour.

Our primary interest is in solutions with $u_0 = 0$. For $u_0 = 0$ it contains points where the velocity crosses $u = 0$ both from above and from below, making it possible to investigate the effect of changing upwind directions.

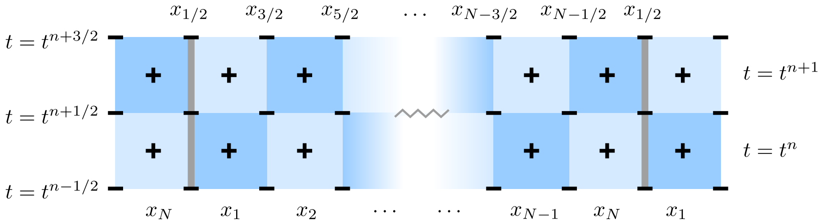


Figure 2.4: Grid staggering in space and time. The thick gray line indicates the boundaries of the computational domain. In this test case periodic boundary conditions are used.

By substitution of the above *prescribed* solution in the momentum equation it follows that it provides an analytical solution to the following *adapted* momentum equation in conservative form

$$\frac{\partial q}{\partial t} + \frac{\partial q^2/h}{\partial x} + \frac{g}{2}\frac{\partial h^2}{\partial x} = \frac{3gH^2\zeta + 3gH\zeta^2 + g\zeta^3}{h^2}\frac{\partial \zeta}{\partial x} := s_q(x,t) \tag{2.14}$$

and in non-conservative form

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial \zeta}{\partial x} = \frac{3gH^2\zeta + 3gH\zeta^2 + g\zeta^3}{h^3}\frac{\partial \zeta}{\partial x} := s_u(x,t) \tag{2.15}$$

It can be seen that the added source terms only contain $O(A^2)$ and higher terms. They have the same order of magnitude as the convection terms. When considering the momentum-conservative form the source term s_q can be interpreted as a bottom source term, reperesenting a moving bottom.

2.4 Numerical scheme of test model

The convection schemes are embedded in a two-step scheme for the nonlinear shallow-water equations. The solution is discretized on a space- and time-staggered grid

$$h_i^n \approx h(x_i, t^n) \qquad\qquad u_{i+1/2}^{n+1/2} \approx u(x_{i+1/2}, t^{n+1/2}) \tag{2.16}$$

where $x_i = x_0 + i\Delta x$, $t^n = n\Delta t$.

First the continuity equation is solved to obtain the water depths h_i^{n+1} at the new time level

$$\frac{h_i^{n+1} - h_i^n}{\Delta t} = -\frac{q_{i+1/2}^{n+1/2} - q_{i-1/2}^{n+1/2}}{\Delta x} \tag{2.17}$$

In the second step the new water depths are used to obtain the new discharge values $q_{i+1/2}^{n+3/2}$ using a discretization of the momentum equation, either in conservative form

$$\frac{q_{i+1/2}^{n+3/2} - q_{i+1/2}^{n+1/2}}{\Delta t} = -C_{i+1/2}^{n+1/2} - \frac{1}{2}g\frac{(h_{i+1}^{n+1})^2 - (h_i^{n+1})^2}{\Delta x} + (s_q)_{i+1/2}^{n+1} \tag{2.18a}$$

or non-conservative form

$$\frac{u_{i+1/2}^{n+3/2} - u_{i+1/2}^{n+1/2}}{\Delta t} = -C_{i+1/2}^{n+1/2} - g\frac{h_{i+1}^{n+1} - h_i^{n+1}}{\Delta x} + (s_u)_{i+1/2}^{n+1} \tag{2.18b}$$

Here $C_{i+1/2}^{n+1/2}$ denotes an explicit discretization of the convection term of which details vary from scheme to scheme and are presented later.

The discrete velocity and momentum variables satisfy the discrete momentum relation

$$q_{i+1/2}^{n+1/2} := \overset{*}{h}{}_{i+1/2}^{n} u_{i+1/2}^{n+1/2} \tag{2.19a}$$

where the missing water levels at the velocity points are approximated using for example an upwind approach

$$\overset{*}{h}{}_{i+1/2}^{n} = \begin{cases} h_i^n, & q_{i+1/2}^{n+1/2} > 0, \text{ or: } u_{i+1/2}^{n+1/2} > 0 \\ h_{i+1}^n, & q_{i+1/2}^{n+1/2} < 0, \text{ or: } u_{i+1/2}^{n+1/2} < 0 \\ \frac{1}{2}\left(h_i^n + h_{i+1}^n\right), & \text{otherwise} \end{cases} \tag{2.19b}$$

The approximation could be improved by using a higher order (limited) approximation for the water levels. If a discretization is used in terms of h and q variables the velocities follow from relation (2.19a) and this can be done similarly if the equations are solved in terms of h and u.

Solving the momentum equation in terms of $q^{n+1/2}$ as one variable it is seen that time integration of the continuity equation is $O(\Delta t^2)$ accurate. If the momentum is expressed in terms of h and u this is no longer the case. Time integration of the momentum equation is $O(\Delta t^2)$ accurate with respect to the hydrostatic pressure gradient but only $O(\Delta t)$ with respect to the explicit convection scheme.

Source terms The source terms given by (2.14) and (2.15) are discretized at the midpoints in time and space in order to obtain an $O(\Delta x^2, \Delta t^2)$ accurate implementation.

Boundary and initial conditions For testing purposes periodic boundary conditions are used. In this way the comparison is not influenced by spurious reflections from the boundaries. The initial condition at $t = 0$ is provided by the prescribed solution.

Stability The local Courant number is given by

$$\max |u \pm \sqrt{gh}| \frac{\Delta t}{\Delta x} \tag{2.20}$$

Due to nonlinear effects it can be difficult to get an a a priori estimate of the maximum Courant number. During the numerical experiments the local Courant number was kept below $\frac{1}{2}$.

Remark The chosen two-step scheme in explicit formulation is more comparable with the scheme used in the non-hydrostatic z-layer module of Delft3D-FLOW. The momentum equation is solved explicitly and the obtained velocity (or momentum) variables are then substituted in an implicit discretization of the continuity equation. The momentum equation can therefore be seen as a prediction step followed by a correction step in order to satisfy the continuity equation at the new time level. However there is one fundamental difference with the solution procedure in Delft3D-FLOW. In the present shallow-water model the time integration scheme is fully explicit and due to time-staggering a quasi-implicit discretization of the continuity equation is obtained. An advantage is that the water levels at the new time level are already available while solving the momentum equation. This property of the scheme will become useful later on.

2.5 Three basic convection schemes

In this section we present several discretization schemes for the convection term in the momentum equation, focusing on conservation properties. A performance comparison of the convection schemes is presented in section 2.9.

First three basic schemes are presented which can be seen as natural discretizations for the momentum-conservative, energy-head conservative and non-conservative formulation of the momentum equation respectively. Performance comparison of these basic schemes needs to give insight in the importance of conservation properties at the discrete level.

First order momentum conservative At the discrete level the shallow-water equations are expressed in h and q variables, any u variables need to be reconstructed. Scheme (2.18a) is used for the discretization of the momentum equation in which the convection term is discretized using a conservative flux formulation:

$$C_{i+1/2}^{n+1/2} := \frac{f_{i+1}^{n+1/2} - f_i^{n+1/2}}{\Delta x} \qquad f_i^{n+1/2} = \begin{cases} \overline{u}_i^{n+1/2} q_{i-1/2}^{n+1/2}, & \overline{u}_i^{n+1/2} \geqslant 0 \\ \overline{u}_i^{n+1/2} q_{i+1/2}^{n+1/2}, & \overline{u}_i^{n+1/2} < 0 \end{cases} \tag{2.21}$$

where the discrete momentum variables are defined by (2.19a) and the transport velocities $\overline{u}_i^{n+1/2}$ are defined as $(u_{i-1/2}^{n+1/2} + u_{i+1/2}^{n+1/2})/2$. The latter can be approximated in two ways. By using the reconstruction

$$u_{i+1/2} := \frac{q_{i+1/2}}{\overset{*}{h}_{i+1/2}} \tag{2.22}$$

the water level is effectively reconstructed using a first-order upwind approach. Writing out the discretized mass fluxes in the continuity equation it is seen that upwind water levels are used

$$\frac{q_{i+1/2} - q_{i-1/2}}{\Delta x} = \frac{u_{i+1/2}\overset{*}{h}_{i+1/2} - u_{i-1/2}\overset{*}{h}_{i-1/2}}{\Delta x} \tag{2.23}$$

On the other hand if a central reconstruction is used

$$u_{i+1/2} := \frac{2q_{i+1/2}}{h_i + h_{i+1}} \tag{2.24}$$

the discretization of the mass fluxes becomes central as well, with the reconstruction of the water levels expressed by

$$\overset{*}{h}_{i+1/2} = \frac{1}{2}\left(h_i + h_{i+1}\right) \tag{2.25}$$

Numerical results confirm the expectation that the second option is more accurate but less robust than the first option.

The momentum-conservative scheme can be rewritten to see that it uses an upwind approach for the background velocity part and a central approach for the gravity-wave part in which the transport velocities are biased towards the upwind side (here for $u > 0$):

$$\underbrace{u_{i+1/2}\frac{q_{i+1/2}-q_{i-1/2}}{\Delta x}}_{\text{background flow}} + \underbrace{\left[q_{i-1/2}\frac{u_{i+1/2}-u_{i-1/2}}{\Delta x} + q_{i+1/2}\frac{u_{i+3/2}-u_{i+1/2}}{\Delta x}\right] \Big/ 2}_{\text{gravity waves}}$$

First-order energy-head conservative For the one-dimensional shallow-water equations it is also possible to obtain an energy-head conservative discretization. By including this scheme in the numerical experiments it is possible to determine the importance of conservation of momentum or conservation in general. Secondly the scheme provides a conservative discretization of the momentum equation in u-formulation so it can be used as fair comparison for the non-conservative scheme. At the discrete level the equations are expressed in h and u variables. Scheme (2.18b) is used for the discretization of the momentum equation in which the convection term is discretized using a conservative flux formulation. A flux vector splitting is applied (see e.g. [10]):

$$f^+_{i+1/2} = \max(0, u_{i+1/2})u_{i+1/2}/2 \qquad f^-_{i+1/2} = \min(0, u_{i+1/2})u_{i+1/2}/2 \tag{2.26}$$

and a first-order energy-head conservative scheme is then given by

$$C^{n+1/2}_{i+1/2} := \frac{f^{n+1/2}_{i+1} - f^{n+1/2}_{i}}{\Delta x} \qquad f_i = f^+_{i-1/2} + f^-_{i+1/2} \tag{2.27}$$

First order non-conservative The convection term is discretized using the finite-difference scheme

$$C^{n+1/2}_{i+1/2} := \begin{array}{ll} u^{n+1/2}_{i+1/2}\dfrac{u^{n+1/2}_{i+1/2} - u^{n+1/2}_{i-1/2}}{\Delta x}, & u^{n+1/2}_{i+1/2} \geqslant 0, \\ u^{n+1/2}_{i+1/2}\dfrac{u^{n+1/2}_{i+3/2} - u^{n+1/2}_{i+1/2}}{\Delta x}, & u^{n+1/2}_{i+1/2} < 0 \end{array} \tag{2.28}$$

2.6 Spatial momentum conservation in non-conservative formulation

In Delft3D-FLOW the equations are expressed in terms of water levels h and velocity variables u, v, w. This framework is not suitable for a momentum conservative discretization in terms of momentum variables (e.g. q in the current shallow-water test case).

When expressing the equations in h and u variables it is however still possible to obtain a scheme which is fully or partially momentum conservative. In 2003 Stelling and Duinmeijer [12] presented a spatially momentum conservative scheme for the shallow-water momentum equations on staggered grids. In this approach, which is used among others in [13],[11],[15] a mommentum-conservative discretization is obtained by using

$$\frac{d}{dt}(\overline{h}_{i+1/2}u_{i+1/2}) = \overline{h}_{i+1/2}\frac{du_{i+1/2}}{dt} + u_{i+1/2}\frac{d\overline{h}_{i+1/2}}{dt} \tag{2.29}$$

The second term at the right hand side is then replaced by substituting an average of the discretized continuity equation:

$$\frac{d\overline{h}_{i+1/2}}{dt} \approx -\frac{\overline{q}^n_{i+1} - \overline{q}^n_i}{\Delta x}$$

where

$$\overline{q}_i = \frac{q_{i+1/2} + q_{i-1/2}}{2} \qquad \overline{h}_{i+1/2} = \frac{h_{i+1} + h_i}{2} \tag{2.30}$$

The time derivative of the conservative discretization is then written as

$$\frac{d}{dt}(\overline{h}_{i+1/2}u_{i+1/2}) = \overline{h}_{i+1/2}\frac{du_{i+1/2}}{dt} - u_{i+1/2}\frac{\overline{q}^n_{i+1} - \overline{q}^n_i}{\Delta x}$$

In this way a momentum-conservative semi-discrete form of the momentum equation is obtained

$$\overline{h}_{i+1/2}\frac{du_{i+1/2}}{dt} \approx -C^{n+1/2}_{i+1/2} - \frac{1}{2}g\frac{(h^{n+1}_{i+1})^2 - (h^{n+1}_i)^2}{\Delta x} + u^{n+1/2}_{i+1/2}\frac{\overline{q}^{n+1/2}_{i+1} - \overline{q}^{n+1/2}_i}{\Delta x} + (s_q)^{n+1}_{i+1/2}$$

The discretization of the hydrostatic pressure gradient can be rewritten to obtain a semi-discrete equation with the time derivative only involving the discrete velocity variable $u_{i+1/2}$:

$$\frac{du_{i+1/2}}{dt} \approx \frac{1}{\overline{h}_{i+1/2}}\left[-C^{n+1/2}_{i+1/2} + u^{n+1/2}_{i+1/2}\frac{\overline{q}^{n+1/2}_{i+1} - \overline{q}^{n+1/2}_i}{\Delta x} + (s_q)^{n+1}_{i+1/2}\right] - g\frac{\zeta^{n+1}_{i+1} - \zeta^{n+1}_i}{\Delta x} \tag{2.31a}$$

In the rest of this thesis, scheme (2.31) is referred to as the *SD-scheme*[2]. Multidimensional variations of this scheme are implemented in the hydrostatic shallow-water model of Delft3D-FLOW [1] but also used in other simulation packages as e.g. SIMONA [2].

Following [12], the discretization of the convection term is chosen as

$$C^{n+1/2}_{i+1/2} := \frac{\overline{q}_{i+1}\overset{*}{u}_{i+1} - \overline{q}_i\overset{*}{u}_i}{\Delta x} \tag{2.31b}$$

using an upwind reconstruction for the missing velocities in the water level points

$$\overset{*}{u}_i = \begin{cases} u_{i-1/2}, & \overline{q}_i > 0, \\ u_{i+1/2}, & \overline{q}_i < 0 \end{cases} \tag{2.32}$$

In this way the momentum variables $\overline{q}_i$ used in the convection scheme are fully consistent with the mass fluxes in the averaged continuity equation. The variables $\overline{q}_i$ act as transport velocity in the convection term and $\overset{*}{u}_i$ is the transported scalar. This is fundamentally different from the upwind procedure used in the basic momentum-conservative scheme. In order to further interpret this difference we write out scheme (2.31b) for positive flow, and rearrange terms to obtain

$$\underbrace{q_{i+1/2}\frac{u_{i+1/2} - u_{i-1/2}}{\Delta x}}_{\text{gravity waves}} + \underbrace{\left[u_{i-1/2}\frac{q_{i+1/2} - q_{i-1/2}}{\Delta x} + u_{i+1/2}\frac{q_{i+3/2} - q_{i+1/2}}{\Delta x}\right] \Big/ 2}_{\text{background flow}}$$

[2] Where the name refers to the authors *S*telling and *D*uinmeijer

From this form it is seen that the scheme adopts an upwind approach for the gravity-wave part and a central approach for the background velocity part. The prescribed solution (2.12) concerns a single wave propagating in one direction only. This implies that half of the time wave information is taken from the upwind side and half of the time from the downwind side, depending on the sign of the velocity.

The difference with the basic momentum conservative upwind scheme is reflected in the truncation error. For the convection term in the SD-scheme the truncation error introduces numerical diffusion of the form

$$\frac{\partial}{\partial x}\left(q\frac{\partial u}{\partial x}\right) \tag{2.33}$$

whereas the basic momentum-conservative scheme introduces numerical diffusion of the form

$$\frac{\partial}{\partial x}\left(u\frac{\partial q}{\partial x}\right) \tag{2.34}$$

For steady state solutions it follows from the continuity equation that q is constant across the domain. From this it follows that in steady state situations the truncation error of the second type vanishes. This is not true for the first type of truncation error, since for steady state solutions the velocity does not need to be constant, most notably in case of steep bathymetry. Numerical damping is desirable for obtaining a stable time integration scheme, however in the steady state case its purpose becomes obsolete and only reduces accuracy. Therefore, for steady state calculations upwinding of the momentum seems to be preferred above upwinding of the velocity.

2.7 Momentum conservative time integration of the SD-scheme

Equation (2.31) is still in semi-discrete form and a choice needs to be made for the method of time integration. An appropriate discretization needs to be chosen in order to obtain full momentum conservation.

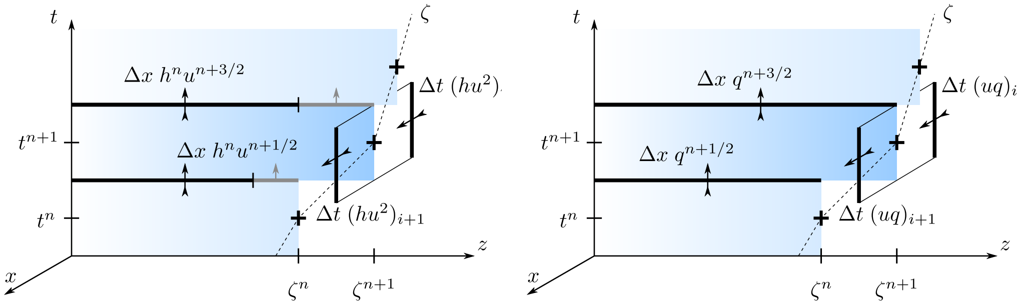
Full momentum conservation is only obtained if also the discretization of the time derivative is conservative. Discretization (2.18a) for the momentum equation can be written in flux formulation

$$\Delta x\left[q^{n+3/2} - q^{n+1/2}\right] + \Delta t\left[(uq)_{i+1} - (uq)_i + \frac{g}{2}\left(h_{i+1}^2 - h_i^2\right)\right] = 0 \tag{2.35}$$

Using a single variable q for the description of momentum the discretization in time automatically becomes conservative.

In case the momentum is expressed in velocity and water level variables the conservation properties depend on the momentum definition at the discrete level. One would like to obtain a discretization in flux form with the momentum variable q replaced by a discrete expression involving h and u. The momentum definition should be consistent across the entire numerical scheme for the shallow-water equations. In the previous sections it has already been seen that in spatial direction several choices can be made for the reconstruction of the momentum $q_{i+1/2}$. Now let us have a look at the discrete momentum definition in time. A logical starting point is to look at the discretization of the time derivative.

For the discretization of the time derivative in scheme (2.31) we present two options:



(a) (h,u)-formulation with frozen water levels

(b) (h,q)-formulation

Figure 2.5: Flux discretization in time. Conservative fluxes are shown as two-sided arrows. When using frozen water levels the grey parts of the fluxes are only discretized at the new time level. Progressing to the next time level the definition of momentum changes from $h^n u^{n+3/2}$ to $h^{n+1} u^{n+3/2}$ causing conservation loss.

- *Spatially conservative form.* As done similarly by [12],[13],[11],[15] the time derivatives can be discretized as

$$
\begin{aligned}
u_{i+1/2}\frac{d\overline{h}_{i+1/2}}{dt} &\approx u^{n+1/2}\frac{\overline{h}^{n+1}-\overline{h}^{n}}{\Delta t}\\
\overline{h}_{i+1/2}\frac{du_{i+1/2}}{dt} &\approx \overline{h}^{n+1/2}\frac{u^{n+3/2}-u^{n+1/2}}{\Delta t}
\end{aligned}
$$

This shows that the time-derivative with respect to the discrete momentum is given by

$$
\begin{aligned}
\Delta t\frac{d\overline{h}_{i+1/2}u_{i+1/2}}{dt} &\approx \overline{h}^{n}\left(u^{n+3/2}-u^{n+1/2}\right)+u^{n+1/2}\left(\overline{h}^{n+1}-\overline{h}^{n}\right)\\
&= \overline{h}^{n+1}u^{n+3/2}-\overline{h}^{n}u^{n+1/2}-\left(u^{n+3/2}-u^{n+1/2}\right)\left(\overline{h}^{n+1}-\overline{h}^{n}\right)\\
&= q^{n+3/2}-q^{n+1/2}-\left(u^{n+3/2}-u^{n+1/2}\right)\left(\overline{h}^{n+1}-\overline{h}^{n}\right) \qquad (2.36)
\end{aligned}
$$

With the discrete momentum defined as

$$
q_{i+1/2}^{n+1/2}=\overline{h}_{i+1/2}^{n}u_{i+1/2}^{n+1/2} \qquad (2.37)
$$

the discretization of the time-derivative can be seen as a linearization of the momentum at the new time level. It follows that momentum conservation is obtained up to $O(\Delta t^2)$ since a conservation error is made proportional to $\Delta t^2\frac{\partial u}{\partial t}\frac{\partial \zeta}{\partial t}$. In applications where the solution does not change much in time or in steady state calculations this error could be neglected. However in simulations with strong surface dynamics and strongly varying flow velocity the error term can become significantly large. The above analysis shows that the difference with genuinely conservative methods should become less as the time step is taken smaller. Moreover, if the Courant number is near unity, as frequently happens in practice, the discretization errors in time can become just as large as for example $O(\Delta x^2)$ conservation errors in space.

One solution for obtaining full conservation in time would be to use the velocity at the new time level

$$\begin{aligned} q^{n+3/2} - q^{n+1/2} &\approx \overline{h}^{n} \left(u^{n+3/2} - u^{n+1/2} \right) + u^{n+3/2} \left(\overline{h}^{n+1} - \overline{h}^{n} \right) \\ &= \overline{h}^{n+1} u^{n+3/2} - \overline{h}^{n} u^{n+1/2} \end{aligned}$$

and substitute the right hand side of the discretized continuity equation. However, since frequently an implicit scheme for the discretization of the continuity equation is used this would introduce implicit coupling.

- *Fully conservative form.* Within the two-step scheme (2.17), (2.18a)/(2.18b), the new water levels h^{n+1} are already available before solving the momentum equation and therefore a fully conservative time integration scheme can also be obtained by multiplying the time derivative for u with h at the new time level

$$\begin{aligned} q^{n+3/2} - q^{n+1/2} &\approx \overline{h}^{n+1} \left(u^{n+3/2} - u^{n+1/2} \right) + u^{n+1/2} \left(\overline{h}^{n+1} - \overline{h}^{n} \right) \\ &= \overline{h}^{n+1} u^{n+3/2} - \overline{h}^{n} u^{n+1/2} \end{aligned} \tag{2.38}$$

  Within the two-step scheme currently used this provides a fully conservative adaptation of the SD-scheme (2.31) with time integration (2.36).

Remark Within the current numerical scheme it is possible to obtain a fully conservative discretization in terms of water level variables h and velocity variables u without the need of substituting the discretized continuity equation. A momentum conservative flux formulation in time is for example provided by

$$\begin{aligned} q^{n+3/2} - q^{n+1/2} &\approx \overline{h}^{n} \left(u^{n+3/2} - u^{n+1/2} \right) + u^{n+1/2} \left(\overline{h}^{n} - \overline{h}^{n-1} \right) \\ &= \overline{h}^{n} u^{n+3/2} - \overline{h}^{n-1} u^{n+1/2} \end{aligned} \tag{2.39}$$

Defining the momentum at the discrete level as $q^{n+1/2} \equiv \overline{h}^{n-1} u^{n+1/2}$ the scheme can be seen to be conservative. Since the continuity equation at the old time level is already solved, $\overline{h}^{n} - \overline{h}^{n-1}$ can be substituted directly. This approach is used in chapter 3 to obtain a conservative discretization of convection at the free-surface.

The accuracy can be improved by using linearization instead of constant extrapolation. By using the continuity equation at the present as well as the old time level, the following discretization is obtained

$$\begin{aligned} & q^{n+3/2} - q^{n+1/2} \\ &\quad \approx \overline{h}^{n} \left(u^{n+3/2} - u^{n+1/2} \right) + u^{n+1/2} \left(\overline{h}^{n+1} - \overline{h}^{n} \right) + \left(u^{n+1/2} - u^{n-1/2} \right) \left(\overline{h}^{n} - \overline{h}^{n-1} \right) \\ &\quad = \left(\overline{h}^{n} u^{n+3/2} + u^{n+1/2} (\overline{h}^{n+1} - \overline{h}^{n}) \right) - \left(\overline{h}^{n-1} u^{n+1/2} + u^{n-1/2} (\overline{h}^{n} - \overline{h}^{n-1}) \right) \end{aligned} \tag{2.40}$$

and full conservation is realized with respect to the linearized discrete momentum

$$q^{n+1/2} \equiv \overline{h}^{n-1} u^{n+1/2} + u^{n-1/2} \left(\overline{h}^{n} - \overline{h}^{n-1} \right)$$

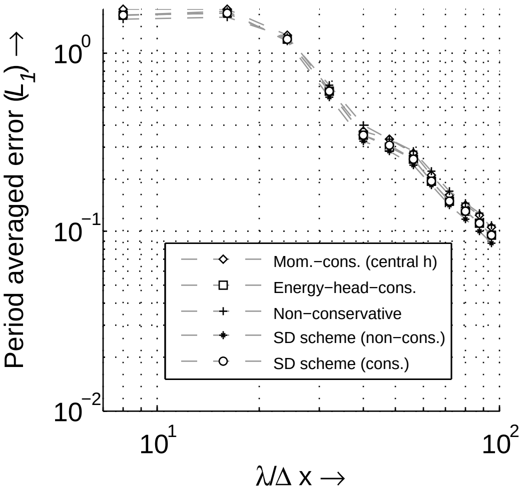


Figure 2.6: Numerical error for a linear wave solution with dimensionless quantity $A/H = 0.005$.

In this way a linearized momentum-conservative scheme is obtained in which the discrete momentum definition is $O(\Delta t^2)$ accurate. For strongly nonlinear problems it seems however more appropriate to avoid linearization and to use a discretization in terms of h and q variables.

2.8 Test case description

For comparison the following schemes have been considered

- *Conservative (upwind h)*: (2.21) with upwind water level reconstruction (2.19b)
- *Energy-head conservative*: (2.27)
- *Non-conservative*: (2.28)
- *SD-scheme (spatially conservative)*: SD-scheme with time discretization (2.36)
- *SD-scheme (conservative)*: SD-scheme with time discretization (2.38)

In order to investigate the behaviour of the different convection schemes in combination with the source term, the simulations are based on periodic boundary conditions. In this way the results are not be disturbed by spurious reflections introduced at the boundaries. The physical domain is set equal to three wave lengths and the results are observed in an arbitrary grid cell as soon as the period-averaged error has converged. In all experiments the period-averaged error stabilized before the end of the simulation. A simulation time of fifty wave periods was sufficient to obtain a stabilized solution.

By using period averaged norms we get a more reliable view on error behaviour than by looking at a single time level. A normalized 1-norm of the approximation errors in the

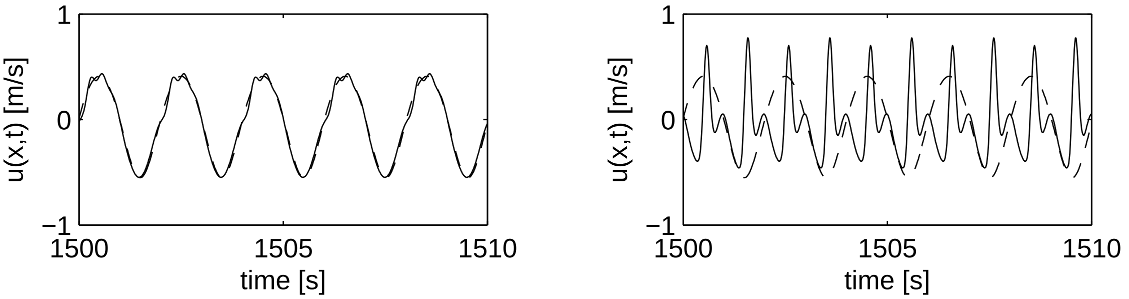


Figure 2.7: Checkpoint analysis for the velocity at $x = 6.15$ after stabilization of the period-averaged error, with the exact solution prescribed as initial condition (left) and with a still water surface ($\zeta = u \equiv 0$) as initial condition (right). The results have been obtained using the basic momentum conservative scheme, with $A/H = 0.15$ and $\lambda/\Delta x = 24$.

observation point across the last wave period is used to compare the numerical results.

$$\epsilon_{1,\mathrm{rel}} = \frac{\sum_{k=n}^{n+N-1} \left| u_{obs+1/2}^{k+1/2} - u(x_{obs}, t^{k+1/2}) \right|}{\sum_{k=n}^{n+N-1} \left| u(x_{obs}, t^{k+1/2}) \right|} \tag{2.41}$$

where n is some time level and N is the number of time steps per wave period. During the numerical experiments it was observed that using the 2-norm or infinity-norm instead, qualitatively similar results are obtained.

The performance of the convection schemes is analyzed by varying the relative wave amplitude A/H and the number of grid points per wave length $\lambda/\Delta x$. The wave length of the exact solution is kept constant at $\lambda = 2\pi$ and the influence of the grid resolution is analyzed by varying Δx. The water depth is kept equal to $H = 1$ hence the wave period is given by $T = 2\pi/\sqrt{g}$. While varying the grid spacing or the relative amplitude, the ratio $\Delta t/\Delta x$ is kept constant at $\Delta t/\Delta x = T/8L$.

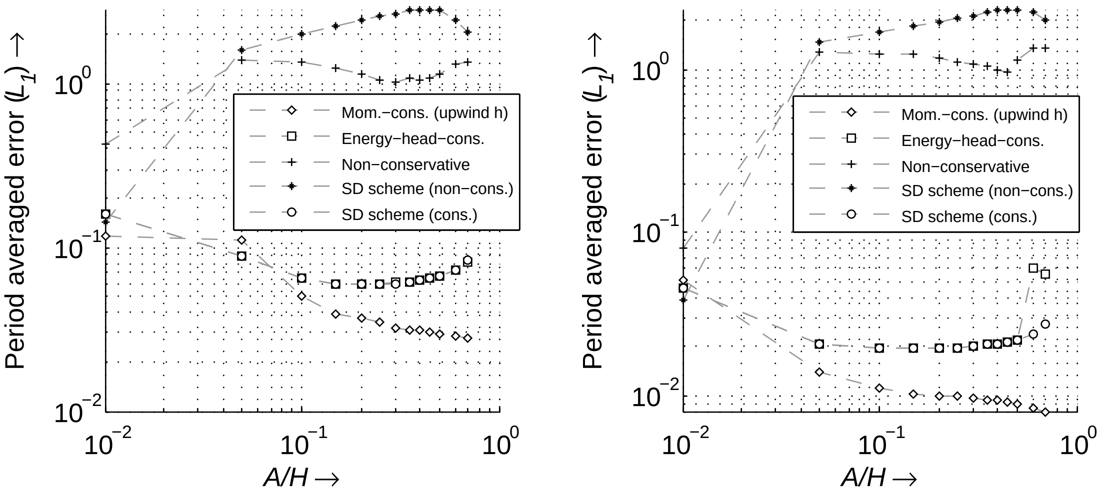
Unless mentioned otherwise a solution without background discharge is considered (i.e. $q_0 = 0$). For simulations with the same wave amplitude and variable grid resolution the Courant number is kept constant which is in correspondence with common practice. For simulations with constant grid resolution and variable wave amplitude the time step is chosen in such a way that the Courant number remains below 1/2 for all simulated wave amplitudes.

Apart from the convection scheme there are other important influences which must be taken into account. The solution is also infuenced by interaction with the source term. As soon as the numerical solution becomes different from the prescribed solution the source term will force it back to the prescribed solution. Since the shallow-water problem becomes strongly non-linear for large amplitude waves, different solutions are possible depending on the numerical parameters used. In order to let the numerical solution converge to the prescribed solution the exact solution is prescribed as initial condition. If a cold start is used the method can converge to a different solution as illustrated in figure 2.7.

The size of the source term is of the same order as the size of the convection term. Nevertheless the source term is exact and at the discrete level is implemented with $O(\Delta t^2)$, so errors made in the convection discretization can still have a large impact on the solution.

2.9 Observations

First the numerical schemes have been tested for a linear wave solution $A/H = 0.005$ in order to investigate convergence behaviour. Figure 2.6 clearly demonstrates the theoretical convergence behaviour of the schemes, which is $O(\Delta x)$. For almost linear solutions the damping effect of the upwind water levels in the continuity equation and the upwind convection term is almost zero (we refer back to figure 2.1 in section 2.1.3). For large amplitude waves the theoretical order of convergence may be more difficult to observe due to the introduction of strongly nonlinear effects. For small amplitude waves all methods show more or less the same relative error since the influence of the convection term is negligible. If the amplitude becomes small compared with the average depth the influence of the convection and source term become negligible and the solution is dominated by the surface gradient (see also figure 2.1 in section 2.1.3). For large amplitude waves nonlinear effects start to play a role and the methods demonstrate different convergence behaviour. The relative error of the conservative scheme remains of the same order whereas the relative error of the non-conservative scheme (2.28) and the SD-scheme with non-conservative time integration (2.36) quickly becomes larger. Observing figure 2.8(a) it is seen that all conservative methods (both momentum and energy-head conservative) give good results, with improvement as the grid resolution is increased. On the other hand all non-conservative schemes start to deviate from the prescribed solution due to interaction with the source term. Similar performance differences are still observed when running simulations with a higher grid resolution. This is seen in figure 2.8(b) where the results are shown for a grid resolution of 108 points per wave length.



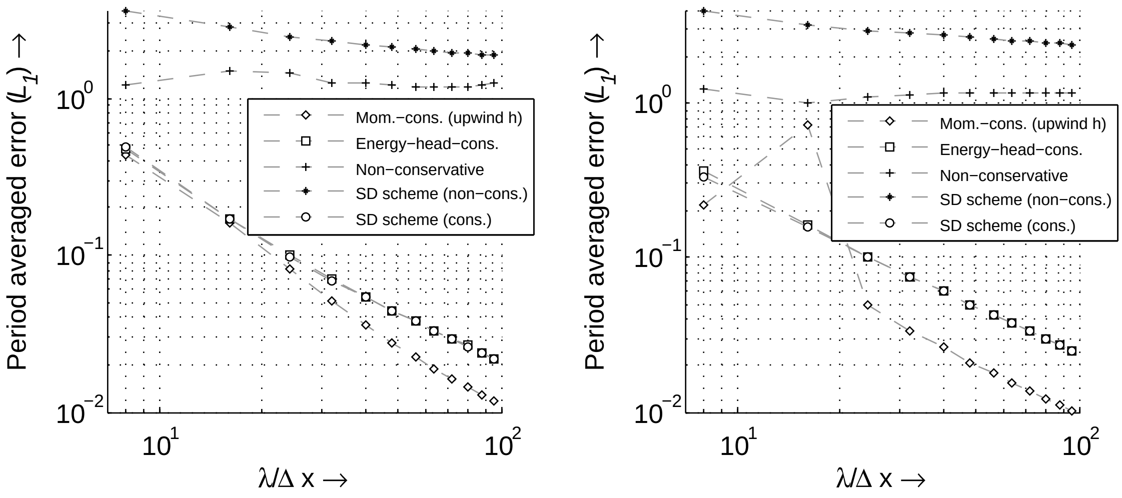
(a) Intermediate grid resolution ($\lambda/\Delta x = 36$)

(b) High grid resolution ($\lambda/\Delta x = 108$)

Figure 2.8: Period averaged error for simulations with a grid resolution of 36 and 108 points per wave length.

Following the observations made in section 2.2 the grid resolution should be taken sufficiently large (say $\lambda/\Delta x > 16$) in order to capture the overall shape of the convection term. However for performance comparison also lower grid resolutions are included.

The conservative methods demonstrate similar convergence behaviour. However especially



(a) Small amplitude ($A/H = 0.15$)

(b) Large amplitude ($A/H = 0.5$)

Figure 2.9: Period averaged error for simulations of waves with small and large amplitudes. The Courant number is kept approximately equal.

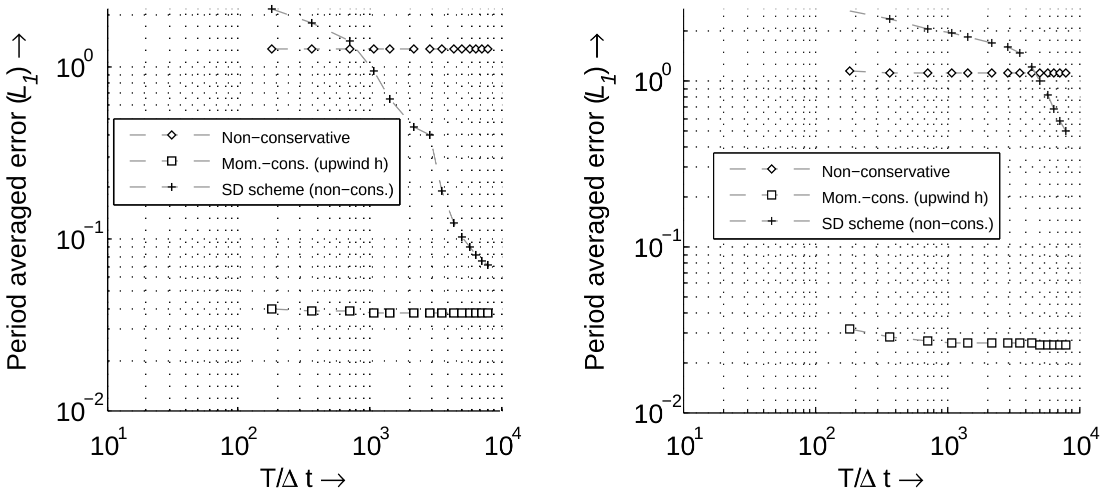
as the grid resolution becomes higher the conservative scheme (2.21) starts to show slightly better convergence behaviour than the energy-head conservative scheme and the fully momentum conservative version of the SD-scheme. In figures 2.9(a) and 2.9(b) it can be seen that the convergence of the conservative scheme is slightly faster and that for intermediate and large amplitudes the error is almost a factor three smaller.

The energy-head conservative scheme and the fully momentum conservative version of the SD-scheme show similar results which only become visibly different for large amplitude waves (see 2.8(a) and 2.8(b)). However this concerns strongly supercritical flow which does not form part of the model purposes of Delft3D-FLOW.

Momentum conservation in time The conservation error made in the spatially conservation version of the SD-scheme is of size $O(\Delta t^2)$ and therefore the performance differences with the fully conservative schemes is expected to become smaller as the time step is increased. The basic momentum-conservative scheme (2.21) (using (2.19b)) and the spatially conservative version of the SD-scheme have been compared for different time step sizes (see figures 2.10(a) and 2.10(b)). It was found that only after strong reduction of the time step the two schemes start to show similar performance. As the relative wave amplitude increases, a larger reduction of the time step size becomes necessary. For small amplitude waves fig. 2.10(b) shows that the time step needs to be reduced by a factor of over 10^3 in order to obtain similar results as with the conservative convection scheme. The CFL number has then already reached below 0.01, which is unacceptable for practical purposes.

The numerical results confirm the ideas set out in the previous sections. Although the convection scheme is spatially conservative, for solutions with time-varying dynamics it is less suitable.

Finally the results in figures 2.10(a) and 2.10(b) demonstrate the improved behaviour of the SD-scheme compared with the non-conservative convection scheme (2.28). Decreasing



(a) Medium amplitude wave

(b) Large amplitude wave

Figure 2.10: Numerical results for the simulation of two different wave amplitudes.

the time step does not result in significant improvement of the non-conservative convection scheme since it still suffers from conservation errors in the spatial discretization.

The convergence to an equilibrium solution can also be observed by inspecting the total momentum. For conservative methods (of any type) the total momentum remained around the initial value where only negligible oscillations are introduced due to the source term which, being a source term, is non-conservative part. For methods that are not fully conservative, on the other hand, a significant production or destruction of momentum is observed.

Remark

During the numerical experiments it was observed that for very large amplitudes or on very coarse grids the conservative first-order upwind scheme becomes unstable. Several other momentum-conservative first-order upwind discretizations could be used instead. In order to make a correct choice for implementation in the three-dimensional momentum equations of Delft3D-FLOW a short comparison has been made.

Using the flux vector splitting (e.g. [9] ch. 13, [10], ch. 15)

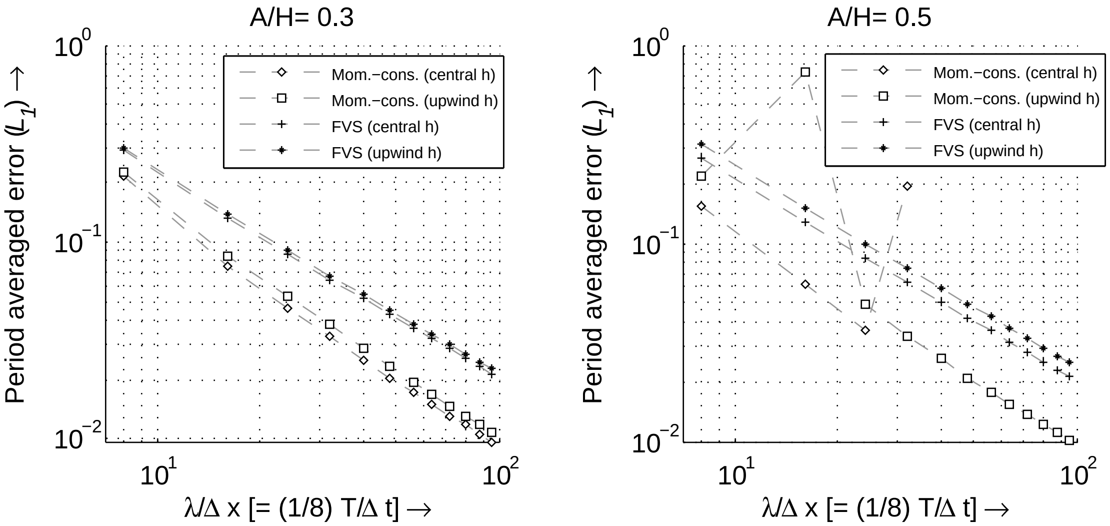
$$f^+_{i+1/2} = \max(0, \overline{u}_{i+1/2}) q_{i+1/2} \qquad f^-_{i+1/2} = \min(0, \overline{u}_{i+1/2}) q_{i+1/2} \tag{2.42}$$

a first-order conservative scheme is given by

$$C^{n+1/2}_{i+1/2} := \frac{f^{n+1/2}_{i+1} - f^{n+1/2}_{i}}{\Delta x} \qquad f_i = f^+_{i-1/2} + f^-_{i+1/2} \tag{2.43}$$

Figure 2.11 shows that the conservative scheme (2.21) performs better than scheme (2.43) based on flux vector splitting. Besides this, figure 2.11 shows that reconstructing the transport velocities with an upwind relation for the water levels leads to a more stable scheme.

Therefore for implementation within Delft3D-FLOW the conservative scheme (2.21) with the transport velocities obtained using a first-order upwind relation for the water levels (2.19b)



(a) Small amplitude ($A/H = 0.15$)

(b) Large amplitude ($A/H = 0.5$)

Figure 2.11: Period averaged error for simulations with four different conservative discretizations.

provides a robust scheme which is more accurate than the conservative convection schemes based on flux vector splitting. It must be mentioned that although there are clear differences between the just presented schemes, all of them are still only $O(\Delta x)$ accurate and significant accuracy improvement is to be obtained by means of implementing higher order (limiter) schemes.

Second order convection schemes

The so far discussed numerical schemes all suffer from some form of artificial dissipation. Firstly in all discretizations expressed in terms of h, u variables a first-order upwind reconstruction is used for the water levels in the velocity points. Secondly in all cases the convection schemes are merely first-order accurate.

When using a discretization for the equation expressed in h and u variables both the velocities and the water levels need to be reconstructed with limiters. The SD-scheme (2.36) can be made second-order accurate in space by using a second-order limited reconstruction of the water levels and velocities as described in [12].

If a discretization is used for the shallow-water equations expressed in h and q variables all terms except the convection term are already discretized with second-order accuracy. Only for the convection scheme a limiter scheme is needed and this makes the discretization more straightforward.

2.10 Conclusions

The numerical results presented in the previous sections clearly demonstrate the better performance of conservative discretizations. Both energy-head and momentum-conservative schemes showed clear superiority above non-conservative schemes, with the basic momentum-conservative scheme being somewhat superior to the others.

Within the present work the results of this research have led to the choice of implementing a momentum-conservative finite-volume method for the momentum prediction equations in Delft3D-FLOW, as to be described in the next chapter. The analysis and test results also clearly indicated the importance of using a scheme which is fully conservative, i.e. both in space and time. The numerical experiments show that the second-order conservation errors in time of the spatially conservative SD-scheme still have a large impact on the approximation of the solution.

The numerical results show that the basic momentum-conservative scheme in terms of mass-flux variables gives the best results and therefore seems the best choice for implementation in Delft3D-FLOW. However, the scheme does not fit in well with the discretization in terms of water levels and velocity variables. Rewriting the existing numerical scheme of Delft3D-FLOW in terms of momentum variables q asks for extensive changes in the discretization of the pressure-correction equations as well as the momentum-prediction equations at the free-surface (see chapter A). For this reason it was chosen to implement a momentum-conservative scheme in terms of water level- and velocity variables. The results of the present chapter show that special attention needs to be paid to the discretization around the free-surface, because this is where the conservation properties of the time-integration scheme come into play.

Further research is possible. In the present model study only a comparison is made of first-order upwind schemes and it might be interesting to investigate the influence of conservation properties of higher order (limiter) schemes. Furthermore it must be noted that the influence of the source term could be reduced by introducing a prescribed solution with higher harmonics.

Chapter 3

A finite-volume discretization

As has been mentioned our main goal is to study and improve the implementation of the momentum equations in the existing non-hydrostatic version of Delft3D-FLOW. This implementation is based on the non-conservative form of the momentum equations. In chapter 2 a model study has been performed showing that momentum conservative schemes, to be formulated starting from the conservative form of the momentum equations, yield better results for convection dominated flow. Therefore our interest is primarily directed towards momentum conservation properties.

With respect to the momentum equations the existing version of Delft3D-FLOW explores a finite-difference scheme. Such an approach is not attractive when starting from the equations in conservative form. In this case the finite-volume method is more suitable. Therefore research focused on the implementation of a conservative convection scheme. A finite-volume discretization is designed for the momentum prediction equations (1.20).

Adopting a finite-volume formulation leads to the necessity of studying boundary conditions in more detail. In fact, a more natural or direct implementation results because the volumes will be such that volume interfaces coincide with boundaries. Special attention is paid to the implementation of the boundary conditions because it forms an important aspect for achieving correct conservation of momentum.

The finite-volume implementation presented in this chapter is intended primarily for strongly non-hydrostatic flow in which case horizontal and vertical flow have the same order of magnitude and the grid spacings are chosen accordingly. For this reason a fully explicit convection scheme can be used without this leading to severe stability constraints. Originally the Delft3D-FLOW software the was designed for the simulation of hydrostatic and weakly non-hydrostatic flow situations leading to the need of implicit time integration of the vertical convection terms in order to avoid too strong stability restrictions in the vertical direction. In contrast to the horizontal convection terms, which are discretized using a multi-dimensional firts order upwind scheme (e.g. [4], p. 33), the vertical convection terms are discretized using a second-order central scheme. In typical applications of Delft3D-FLOW the vertical velocity is very small and the central scheme does not make the model unstable and in this way can be used for accurate transport modeling.

In the current work the already existing Delft3D-FLOW finite-difference implementation of the viscosity terms will be retained. It is easy to see that the difference between a finite-volume and finite-difference formulation is marginal and it was thought that changing the code in this aspect as well, would lead to marginal differences.

The test cases in chapter 2 showed that momentum conservative numerical schemes yield better results for convection dominated flow. In the present chapter we therefore have a look at the three-dimensional implementation of conservative schemes. Initially the convection terms will be implemented using a first-order upwind scheme since for now we are primarily interested in the momentum-conservation properties and not yet in the reduction of artificial damping.

A solution procedure is used as described in section 1.4 where the original finite-difference discretization of the momentum prediction equations (1.20) is replaced by a finite-volume discretization of the momentum equations in conservative formulation (equations (1.3)) leading to

$$\frac{\mathbf{u}^{n+1*}-\mathbf{u}^n}{\Delta t}+\nabla(\mathbf{u}^n\mathbf{u}^{nT})=-\frac{1}{\rho}\nabla p^n+\nu_h\left(\frac{\partial^2\mathbf{u}^n}{\partial x^2}+\frac{\partial^2\mathbf{u}^n}{\partial y^2}\right)+\nu_v\frac{\partial^2\mathbf{u}^{n+1*}}{\partial z^2}+\mathbf{g} \tag{3.1}$$

3.1 Coordinate transformed equations

In the existing Delft3D-FLOW model a horizontal coordinate transformation is applied in order to support discretization on curvilinear grids. This transformation has consequences for the momentum conservation properties of the numerical scheme as described further on in this section. For this reason it has been chosen to incorporate the existing coordinate transformation in the description of the finite-volume method presented in this chapter.

The physical domain is discretized on a z-layer grid having strictly vertical grid lines and strictly horizontal grid planes. In physical space the grid is described by the cell vertices

$$(x,y,z)_{i+1/2,j+1/2,k+1/2},\quad i=0,\ldots,I,\ j=0,\ldots,J,\ k=0,\ldots,K \tag{3.2}$$

In horizontal direction a curvilinear coordinate transformation is applied as commonly done in practice, e.g. [2],[1],[21]. Although the equations are discretized on a z-layer grid with strictly vertical grid lines, also the z and time dimensions are included in the transformation because this will provide a useful tool for the derivation of discretization methods near the moving free-surface boundaries. The rectangular computational grid is described by the vertices

$$(\xi,\eta,\sigma)_{i+1/2,j+1/2,k+1/2}=\left((i+1/2)\Delta\xi,(j+1/2)\Delta\eta,\sum_{\kappa=1}^{k}\Delta\sigma_\kappa\right),$$
$$i=0,\ldots,I,\ j=0,\ldots,J,\ k=0,\ldots,K \tag{3.3}$$

The coordinate transformation is described by

$$t=t(\tau)\qquad z=z(\tau,\xi,\eta,\sigma)\qquad x=x(\tau,\xi,\eta,\sigma)\qquad y=y(\tau,\xi,\eta,\sigma) \tag{3.4a}$$

the coordinate transformation provides a mapping between the computational and physical grid points

$$(x,y,z)_{i+1/2,j+1/2,k}=(x(\xi,\eta,\sigma),y(\xi,\eta,\sigma),z(\xi,\eta,\sigma))_{i+1/2,j+1/2,k} \tag{3.5}$$

where $x(\xi,\eta,\sigma),y(\xi,\eta,\sigma),z(\xi,\eta,\sigma)$ are coordinate mappings from the dimensionless computational space (ξ,η,σ) to the physical grid space (x,y,z).

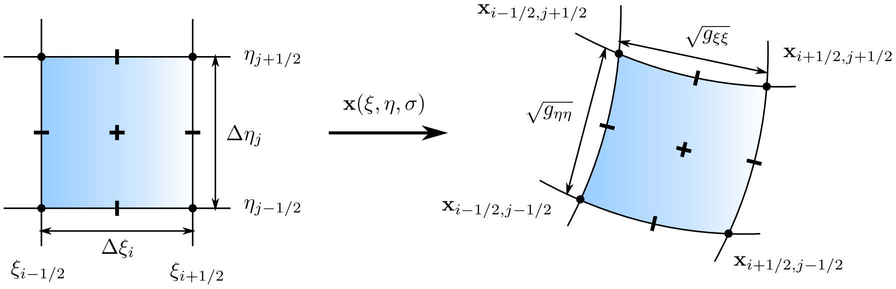

$\eta_{j+1/2}$
$\Delta\eta_j$
$\eta_{j-1/2}$
$\Delta\xi_i$
$\xi_{i-1/2}$
$\xi_{i+1/2}$
$\mathbf{x}(\xi,\eta,\sigma)$
$\mathbf{x}_{i-1/2,j+1/2}$
$\sqrt{g_{\xi\xi}}$
$\mathbf{x}_{i+1/2,j+1/2}$
$\sqrt{g_{\eta\eta}}$
$\mathbf{x}_{i-1/2,j-1/2}$
$\mathbf{x}_{i+1/2,j-1/2}$

Figure 3.1: A computational grid cell and its image in physical space.

In the rest of the document it is assumed that $\Delta\xi = \Delta\eta = \Delta\sigma = 1$. Since the variables ξ, η and σ are dimensionless they can be left out of the difference schemes.

Generally the exact coordinate mapping is not available. Therefore it is approximated numerically based on the physical grid coordinates (as discussed in the next section). The z-layer grid contains strictly vertical grid lines which do not move in time, so $x_\tau = y_\tau = 0$ and $x_\sigma = y_\sigma = 0$ and it follows that the derivatives are transformed according to

$$\begin{pmatrix} \frac{\partial}{\partial\tau} \\ \frac{\partial}{\partial\xi} \\ \frac{\partial}{\partial\eta} \\ \frac{\partial}{\partial\sigma} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & z_\tau \\ 0 & x_\xi & y_\xi & z_\xi \\ 0 & x_\eta & y_\eta & z_\eta \\ 0 & 0 & 0 & z_\sigma \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial t} \\ \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} \tag{3.6}$$

with the corresponding inverse relation given by

$$\begin{pmatrix} \frac{\partial}{\partial t} \\ \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} = \frac{1}{J} \begin{pmatrix} J & 0 & 0 & -z_\tau\left(x_\xi y_\eta - x_\eta y_\xi\right) \\ 0 & z_\sigma y_\eta & -z_\sigma y_\xi & y_\xi z_\eta - z_\xi y_\eta \\ 0 & -z_\sigma x_\eta & z_\sigma x_\xi & x_\eta z_\xi - x_\xi z_\eta \\ 0 & 0 & 0 & x_\xi y_\eta - x_\eta y_\xi \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial\tau} \\ \frac{\partial}{\partial\xi} \\ \frac{\partial}{\partial\eta} \\ \frac{\partial}{\partial\sigma} \end{pmatrix} \tag{3.7}$$

where the Jacobian

$$J = z_\sigma(x_\xi y_\eta - x_\eta y_\xi) \tag{3.8}$$

describes the infinitesimal volume size. The coordinate transformation is assumed to be orthogonal, hence

$$x_\xi x_\eta + y_\xi y_\eta = 0 \tag{3.9}$$

From this orthogonality property it follows that the grid cell surface equals the product of the lengths of the grid lines

$$\sqrt{g^*} := \sqrt{g_{\xi\xi}}\sqrt{g_{\eta\eta}} = x_\xi y_\eta - x_\eta y_\xi \tag{3.10}$$

where

$$\sqrt{g_{\xi\xi}} = \sqrt{x_\xi^2 + y_\xi^2} \qquad \sqrt{g_{\eta\eta}} = \sqrt{x_\eta^2 + y_\eta^2} \tag{3.11}$$

The infinitesimal volume size is then simply expressed by

$$J = z_\sigma \sqrt{g^*} \tag{3.12}$$

The free-surface equations will be expressed in velocities perpendicular to the cell interface. For this purpose we introduce the contravariant velocity variables are introduced.

$$U = \frac{y_\eta u - x_\eta v}{\sqrt{g_{\eta\eta}}} \qquad\qquad V = \frac{x_\xi v - y_\xi u}{\sqrt{g_{\xi\xi}}} \tag{3.13}$$

> *Note: For ease of notation in the remainder of this thesis the velocity variables u and v will refer to the contravariant velocity components:*

$$U \to u \qquad\qquad V \to v$$

3.1.1 Transformed free-surface equations

In this section a brief overview is given of the transformed free-surface equations. Deriving the transformed equations asks for some lengthy computations, but is straightforward. Here we only present the final transformed equations.

It can be shown that the transformed continuity equation can be expressed in contravariant velocity components

$$\frac{\partial z_\sigma \sqrt{g_{\eta\eta}} u}{\partial \xi} + \frac{\partial z_\sigma \sqrt{g_{\xi\xi}} v}{\partial \eta} + \frac{\partial}{\partial \sigma}\left(-z_\xi \sqrt{g_{\eta\eta}} u - z_\eta \sqrt{g_{\xi\xi}} v + \sqrt{g^*} w\right) = 0 \tag{3.14}$$

In terms of contravariant velocity components the momentum equations are

$$\begin{aligned}\frac{\partial J u}{\partial \tau} &+ \frac{\partial z_\sigma \sqrt{g_{\eta\eta}} u^2}{\partial \xi} + \frac{\partial z_\sigma \sqrt{g_{\xi\xi}} uv}{\partial \eta} \\ &+ \frac{\partial}{\partial \sigma}\left(-z_\tau \sqrt{g^*} u - z_\xi \sqrt{g_{\eta\eta}} u^2 - z_\eta \sqrt{g_{\xi\xi}} vu + \sqrt{g^*} wu\right) \\ &- z_\sigma v^2 \frac{\partial \sqrt{g_{\eta\eta}}}{\partial \xi} + z_\sigma uv \frac{\partial \sqrt{g_{\xi\xi}}}{\partial \eta} = -\frac{\partial \sqrt{g_{\eta\eta}} z_\sigma p}{\partial \xi} + z_\sigma p \frac{\partial \sqrt{g_{\eta\eta}}}{\partial \xi} + \frac{\partial \sqrt{g_{\eta\eta}} z_\xi p}{\partial \sigma}\end{aligned} \tag{3.15a}$$

$$\begin{aligned}\frac{\partial J v}{\partial \tau} &+ \frac{\partial}{\partial \xi}\left(z_\sigma \sqrt{g_{\eta\eta}} uv\right) + \frac{\partial}{\partial \eta}\left(z_\sigma \sqrt{g_{\xi\xi}} v^2\right) \\ &+ \frac{\partial}{\partial \sigma}\left(-z_\tau \sqrt{g^*} v - z_\xi \sqrt{g_{\eta\eta}} uv - z_\eta \sqrt{g_{\xi\xi}} v^2 + \sqrt{g^*} wv\right) \\ &+ z_\sigma uv \frac{\partial \sqrt{g_{\eta\eta}}}{\partial \xi} - z_\sigma u^2 \frac{\partial \sqrt{g_{\xi\xi}}}{\partial \eta} = -\frac{\partial \sqrt{g_{\xi\xi}} z_\sigma p}{\partial \eta} + z_\sigma p \frac{\partial \sqrt{g_{\xi\xi}}}{\partial \eta} + \frac{\partial \sqrt{g_{\xi\xi}} z_\eta p}{\partial \sigma}\end{aligned} \tag{3.15b}$$

$$\begin{aligned}\frac{\partial J w}{\partial \tau} &+ \frac{\partial}{\partial \xi}\left(z_\sigma \sqrt{g_{\eta\eta}} uw\right) - \frac{\partial}{\partial \eta}\left(z_\sigma \sqrt{g_{\xi\xi}} vw\right) \\ &+ \frac{\partial}{\partial \sigma}\left(-z_\tau \sqrt{g^*} w - z_\xi \sqrt{g_{\eta\eta}} uw - z_\eta \sqrt{g_{\xi\xi}} vw + \sqrt{g^*} w^2\right) = -\frac{\partial \sqrt{g^*} p}{\partial \sigma}\end{aligned} \tag{3.15c}$$

Due to the use of contravariant velocity components some non-conservative terms appear in the equations. Firstly the convection is no longer conservative due to the presence of the centrifugal terms

$$z_\sigma v^2 \frac{\partial}{\partial \xi}\left(\sqrt{g_{\eta\eta}}\right) \qquad z_\sigma uv \frac{\partial}{\partial \eta}\left(\sqrt{g_{\xi\xi}}\right) \qquad z_\sigma uv \frac{\partial}{\partial \xi}\left(\sqrt{g_{\eta\eta}}\right) \qquad z_\sigma uu \frac{\partial}{\partial \eta}\left(\sqrt{g_{\xi\xi}}\right)$$

and secondly the pressure derivative is no longer conservative. It is possible to rewrite the pressure terms in conservative form with a non-conservative correction

$$\begin{aligned}\frac{\sqrt{g_{\eta\eta}}}{\rho}\frac{\partial z_\sigma p}{\partial \xi} &= \frac{1}{\rho}\frac{\partial \sqrt{g_{\eta\eta}} z_\sigma p}{\partial \xi} - \frac{z_\sigma p}{\rho}\frac{\partial \sqrt{g_{\eta\eta}}}{\partial \xi},\\ \frac{\sqrt{g_{\xi\xi}}}{\rho}\frac{\partial z_\sigma p}{\partial \eta} &= \frac{1}{\rho}\frac{\partial \sqrt{g_{\xi\xi}} z_\sigma p}{\partial \eta} - \frac{z_\sigma p}{\rho}\frac{\partial \sqrt{g_{\xi\xi}}}{\partial \eta}\end{aligned}$$

At the discrete level the same can be done in order to obtain maximum conservation properties (see section 3.8). In computational space the kinematic conditions are given by

$$J z_\tau + z_\sigma \sqrt{g_{\eta\eta}} u z_\xi + z_\sigma \sqrt{g_{\xi\xi}} v z_\eta - J w = 0 \tag{3.16}$$

and at the bottom

$$z_\sigma \sqrt{g_{\eta\eta}} u z_\xi + z_\sigma \sqrt{g_{\xi\xi}} v z_\eta - J w = 0 \tag{3.17}$$

In the remainder of this document the non-conservative parts of the momentum equations are left out of the discussion. These terms are simply implemented using central difference schemes.

3.2 The staggered z-layer grid

The computational grid is defined by (3.3) and corresponds to the curvilinear z-layer grid (3.2) in physical space. The horizontal spacings of the physical grid are defined at the location of the velocity points and follow from the coordinate transformation.

$$\sqrt{g_{\xi\xi}}\big|_{i,j+1/2} = \sqrt{(x_{i+1/2,j+1/2} - x_{i-1/2,j+1/2})^2 + (y_{i+1/2,j+1/2} - y_{i-1/2,j+1/2})^2} \tag{3.18a}$$

$$\sqrt{g_{\eta\eta}}\big|_{i+1/2,j} = \sqrt{(x_{i+1/2,j-1/2} - x_{i+1/2,j-1/2})^2 + (y_{i+1/2,j+1/2} - y_{i+1/2,j-1/2})^2} \tag{3.18b}$$

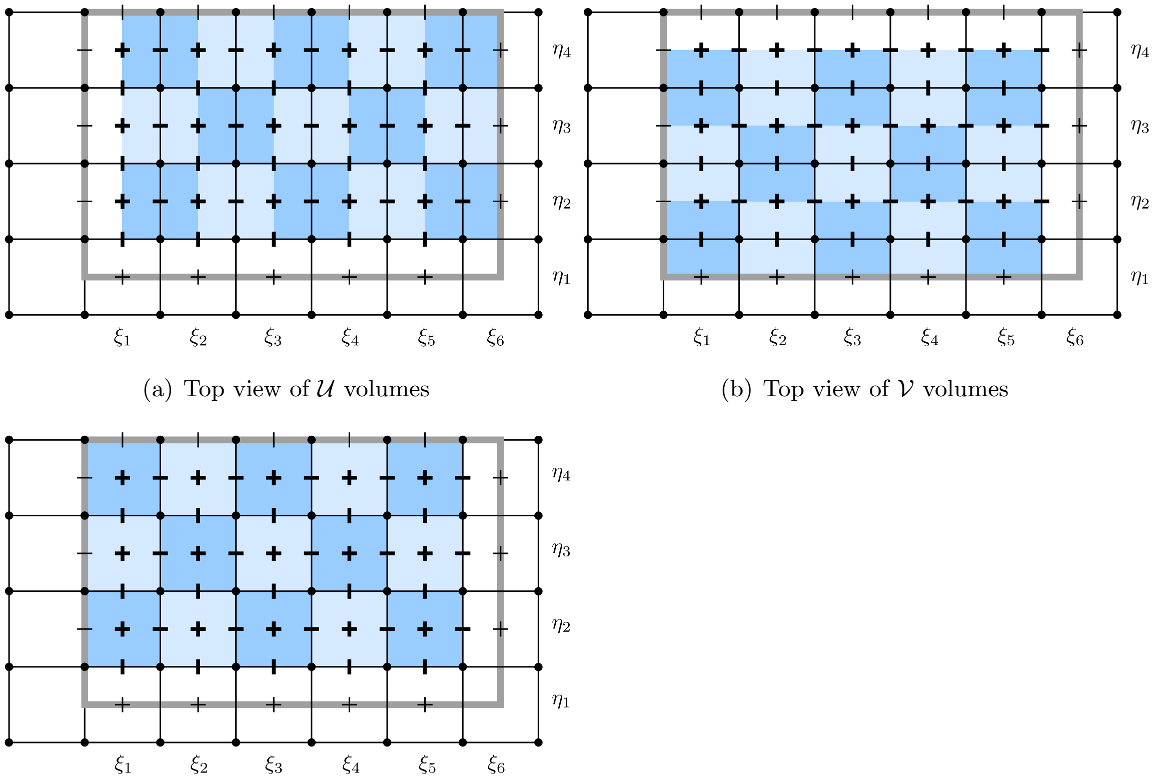
These variables describe the length of the cell interfaces in physical space along the ξ and η directions and if a Cartesian grid is used simply equal Δx_i and Δy_j.

Grid staggering is used for the discretization of the different variables. The non-hydrostatic pressure values and water levels are defined in the grid cell centers (i.e. in (ξ, η, σ) grid space), and are denoted by $d_{i,j}, \zeta_{i,j}$ respectively (plus symbols in figure 3.2).

In computational grid space the continuity equation (3.14) is discretized on the grid cells

$$\mathcal{M}_{i,j,k} = \left\{\xi_{i-1/2} \leqslant \xi \leqslant \xi_{i+1/2}, \eta_{j-1/2} \leqslant \eta \leqslant \eta_{j+1/2}, \sigma_{k-1/2} \leqslant \sigma \leqslant \sigma_{k+1/2}\right\} \tag{3.19}$$

In the following these will be referred to as *mass volumes.* The velocity points are centered on the cell faces and denoted by $u_{i+1/2,j,k}, v_{i,j+1/2,k}, w_{i,j,k}$ respectively (dashes in figure 3.2).



(a) Top view of $\mathcal{U}$ volumes

(b) Top view of $\mathcal{V}$ volumes

(c) Top view of $\mathcal{W}$ and $\mathcal{M}$ volumes

Figure 3.2: Top view of the locations of the momentum volumes in computational space. The equations are discretized on a staggered grid using an Arakawa C-grid ordering (see also [18]). Near velocity and water level openings the volumes do not always reach up to the boundaries. Open boundaries are located at the thick gray line, closed boundaries are always located in velocity points.

The momentum equations (3.15a),(3.15b) and (3.15c) are discretized on grid cells centered around these velocity variables (see figure 3.2).

$$\mathcal{U}_{i+1/2,j,k} = \left\{\xi_i \leqslant \xi \leqslant \xi_{i+1}, \eta_{j-1/2} \leqslant \eta \leqslant \eta_{j+1/2}, \sigma_{k-1/2} \leqslant \sigma \leqslant \sigma_{k+1/2}\right\} \tag{3.20a}$$

$$\mathcal{V}_{i,j+1/2,k} = \left\{\xi_{i-1/2} \leqslant \xi \leqslant \xi_{i+1/2}, \eta_j \leqslant \eta \leqslant \eta_{j+1}, \sigma_{k-1/2} \leqslant \sigma \leqslant \sigma_{k+1/2}\right\} \tag{3.20b}$$

$$\mathcal{W}_{i,j,k+1/2} = \left\{\xi_{i-1/2} \leqslant \xi \leqslant \xi_{i+1/2}, \eta_{j-1/2} \leqslant \eta \leqslant \eta_{j+1/2}, \sigma_k \leqslant \sigma \leqslant \sigma_{k+1}\right\} \tag{3.20c}$$

In the following these will be referred to as *momentum volumes*.

The curvilinear grid spacings $\sqrt{g_{\xi\xi}}$ and $\sqrt{g_{\eta\eta}}$ are calculated around the v and u-velocity points respectively. Around the water level points they are obtained using averaging of the nearest two grid spacings[1]

$$\sqrt{g_{\xi\xi}}\big|_{i,j} = \frac{1}{2}\left(\sqrt{g_{\xi\xi}}\big|_{i,j+1/2} + \sqrt{g_{\xi\xi}}\big|_{i,j-1/2}\right) \tag{3.21a}$$

$$\sqrt{g_{\eta\eta}}\big|_{i,j} = \frac{1}{2}\left(\sqrt{g_{\eta\eta}}\big|_{i+1/2,j} + \sqrt{g_{\eta\eta}}\big|_{i-1/2,j}\right) \tag{3.21b}$$

and around the other velocity points

$$\sqrt{g_{\xi\xi}}\big|_{i+1/2,j} = \frac{1}{4}\left(\sqrt{g_{\xi\xi}}\big|_{i,j+1/2} + \sqrt{g_{\xi\xi}}\big|_{i,j-1/2} + \sqrt{g_{\xi\xi}}\big|_{i+1,j+1/2} + \sqrt{g_{\xi\xi}}\big|_{i+1,j-1/2}\right) \tag{3.21c}$$

$$\sqrt{g_{\eta\eta}}\big|_{i,j+1/2} = \frac{1}{4}\left(\sqrt{g_{\eta\eta}}\big|_{i+1/2,j} + \sqrt{g_{\eta\eta}}\big|_{i-1/2,j} + \sqrt{g_{\eta\eta}}\big|_{i+1/2,j+1} + \sqrt{g_{\eta\eta}}\big|_{i-1/2,j+1}\right) \tag{3.21d}$$

The averaging of the horizontal grid spacing introduces numerical errors which become larger as the curvilinear grids starts to deviate from orthogonality or variation in the grid spacing becomes larger. All kinds of conditions can be derived in order to limit this error (see e.g. [2]). A better approximation of the grid spacing is possible, this is however not subject of the current research. For now all model experiments are based on rectangular grids.

For each grid index i, j the values $k_{\max}^{i,j;n}$ and $k_{\min}^{i,j}$ are defined to indicate the local index of the layer containing the bottom and free-surface

$$\begin{aligned} k_{\max}^{i,j;n} &= \max\left\{k \mid z_{k-1/2} < \zeta_{i,j}^{n}, 1 \leqslant k \leqslant K\right\}, \\ k_{\min}^{i,j} &= \min\left\{k \mid z_{k+1/2} > -d_{i,j}, 1 \leqslant k \leqslant K\right\} \end{aligned} \tag{3.22}$$

The pressure is defined for all (partially) filled mass volumes

$$p_{i,j,k},\ k_{\min}^{i,j} \leqslant k \leqslant k_{\max}^{i,j;n} \tag{3.23}$$

A mass control volume is called *active* if its corresponding pressure point is defined. The velocity points are active if and only if they are located between two active pressure points (see figure 3.3) which can also include water level boundaries. Only exception to this rule concerns the velocity variables in vertical free-surface sections, a z-layer artifact which is discussed in section 3.10. A momentum control volume is called *active* if the corresponding velocity point is active which is the case if and only if the two neighbouring pressure points are defined (including water level boundaries). Grid staggering results in the existence of half volumes near the boundaries which are not included in all momentum equations (consult also figure 3.2).

In order to describe the indices of the lowest and highest active horizontal momentum volumes we define (see also section 3.6)

$$k_{\min}^{i+1/2,j} = \max\{k_{\min}^{i,j}, k_{\min}^{i+1,j}\}, \tag{3.24a}$$

$$k_{\min}^{i,j+1/2} = \max\{k_{\min}^{i,j}, k_{\min}^{i,j+1}\}, \tag{3.24b}$$

$$k_{\max}^{i+1/2,j;n} = \begin{cases} k_{\max}^{i,j;n}, & u_{i+1/2,j} > 0, \\ k_{\max}^{i+1,j;n}, & u_{i+1/2,j} < 0 \end{cases} \tag{3.24c}$$

$$k_{\max}^{i,j+1/2;n} = \begin{cases} k_{\max}^{i,j;n}, & v_{i,j+1/2} > 0, \\ k_{\max}^{i,j+1;n}, & v_{i,j+1/2} < 0 \end{cases} \tag{3.24d}$$

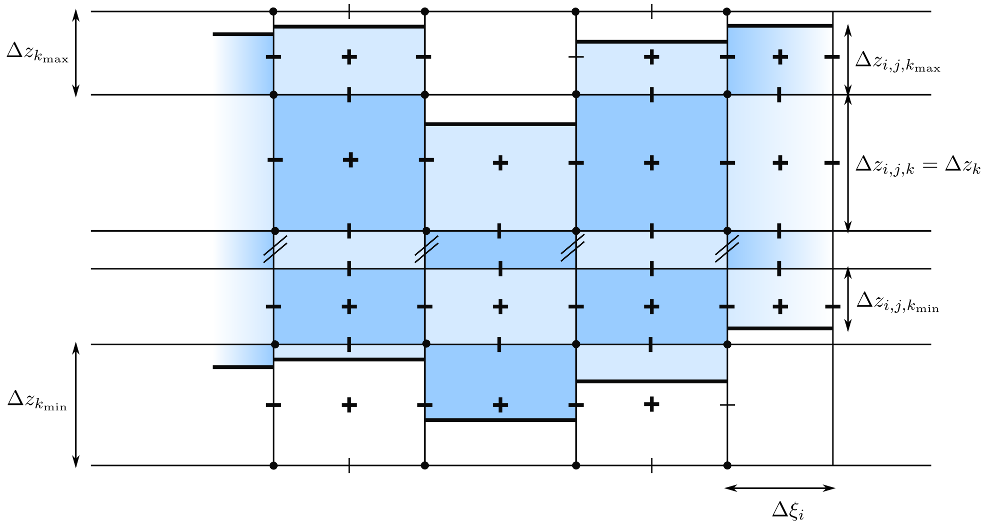

$\Delta z_{k_{\max}}$
$\Delta z_{i,j,k_{\max}}$
$\Delta z_{i,j,k} = \Delta z_k$
$\Delta z_{i,j,k_{\min}}$
$\Delta z_{k_{\min}}$
$\Delta \xi_i$

Figure 3.3: 2DV view of the z-layer grid as currently used in the Delft3D non-hydrostatic module. The horizontal lines are only to indicate the bottom level in the water level points, in the new model the bottom is discretized as piecewise linear (see section 3.6).

The indices of the vertical momentum volumes range from $k_{\text{min}}^{i,j} + 1/2$ up to $k_{\text{max}}^{i,j;n} - 1/2$. The variables Δz_k denote the *fixed* spacing of the z-layer grid in vertical direction. In the top and bottom grid layer the location of the horizontal grid lines depends on the local solution. For (partially) filled grid cells the locations are defined as

$$z_{i,j,k+1/2} = \min(z_{k+1/2}, \zeta_{i,j}^n) \tag{3.25}$$

$$z_{i,j,k-1/2} = \max(z_{k-1/2}, -d_{i,j}) \tag{3.26}$$

$$z_{i,j,k} = \frac{1}{2}\left(z_{i,j,k-1/2} + z_{i,j,k-1/2}\right) \tag{3.27}$$

the indices i, j have been added to distinguish the values from the fixed grid. The vertical grid spacing is defined as

$$\Delta z_{i,j,k} = z_{i,j,k+1/2} - z_{i,j,k-1/2} \qquad k_{\text{min}}^{i,j} \leqslant k \leqslant k_{\text{max}}^{i,j;n} \tag{3.28}$$

$$\Delta z_{i,j,k+1/2} = z_{i,j,k+1} - z_{i,j,k} \qquad k_{\text{min}}^{i,j} \leqslant k < k_{\text{max}}^{i,j;n} \tag{3.29}$$

In this staggered arrangement of momentum volumes the lower and upper faces of vertical momentum volumes are located at the center of the neighbouring mass volumes as is also the case in ξ-direction for $\mathcal{U}$ volumes and in η-direction for $\mathcal{V}$ volumes. The vertical spacing at the non-integer indices demands special attention and will be discussed in sections 3.6 and 3.7.

The accuracy of the discretization of the finite-volume geometry is determined by the following aspects:

- definition of the volume sizes

[1] Given a generated curvilinear orthogonal grid instead of using averages the grid spacing in these points could also be calculated directly.

- size of the flux faces

At the discrete level the size of the mass and momentum volumes in physical space (the Jacobian of the coordinate transformation) is approximated by

$$V_{i,j,k} = \sqrt{g^*}_{i,j} \Delta z_{i,j,k} \tag{3.30a}$$

$$V_{i+1/2,j,k} = \frac{1}{2}\left(\sqrt{g^*}_{i,j} \Delta z_{i,j,k} + \sqrt{g^*}_{i+1,j} \Delta z_{i+1,j,k}\right) \tag{3.30b}$$

$$V_{i,j+1/2,k} = \frac{1}{2}\left(\sqrt{g^*}_{i,j} \Delta z_{i,j,k} + \sqrt{g^*}_{i,j+1} \Delta z_{i,j+1,k}\right) \tag{3.30c}$$

$$V_{i,j,k+1/2} = \frac{1}{2}\sqrt{g^*}_{i,j} \left(\Delta z_{i,j,k} + \Delta z_{i,j,k+1}\right) \tag{3.30d}$$

respectively. The discretization of the volume sizes is second-order accurate because for the size of the mass volumes the midpoint rule is used and the size of the other volumes are obtained with central averaging. Defining the volume sizes in this way in the bottom and top grid cells results in second-order truncation errors depending on the curvature of the bottom and free-surface. Setting aside open boundaries in the horizontal plane at the discrete level the total size of the horizontal momentum volumes equals that of the mass volumes. Another motivation of these choices can be found in section 3.4. The size of the mass-flux interfaces in ξ and η-direction is given by

$$\sqrt{g_{\eta\eta}}_{i+1/2,j} \Delta z_{i+1/2,j,k} \qquad\qquad \sqrt{g_{\xi\xi}}_{i,j+1/2} \Delta z_{i,j+1/2,k}$$

respectively. The accuracy of this discretization depends on the approximation of the vertical spacings Δz at the non-integer indices (see section 3.6) which in the current work is first-order accurate. The flux faces of the momentum volumes in parallel direction (i.e. for the fluxes u^2, v^2, w^2) are located at the centers of the mass volumes and discretized with second-order accuracy. For the flux interfaces of the horizontal momentum volumes in the sidewards directions an upwind approach is used. For $\mathcal{U}$-volumes the area of the flux interface equals the average of the neighbouring mass flux interfaces in η-direction:

$$\overline{\sqrt{g_{\eta\eta}}\Delta z}^{(\eta)}_{i+1/2,j+1/2,k} = \frac{1}{2}\left(\sqrt{g_{\xi\xi}}_{i,j+1/2} \Delta z_{i,j+1/2,k} + \sqrt{g_{\xi\xi}}_{i+1,j+1/2} \Delta z_{i+1,j+1/2,k}\right)$$

For $\mathcal{V}$-volumes the same approach is used, leading to

$$\overline{\sqrt{g_{\xi\xi}}\Delta z}^{(\xi)}_{i+1/2,j+1/2,k} = \frac{1}{2}\left(\sqrt{g_{\xi\xi}}_{i+1/2,j} \Delta z_{i+1/2,j,k} + \sqrt{g_{\xi\xi}}_{i+1/2,j+1} \Delta z_{i+1/2,j+1,k}\right)$$

In this approach the total area of the momentum flux interfaces equals the total size of the mass flux interfaces. It is seen that the accuracy of the approximation of the flux interfaces in the sidewards directions depends on the accuracy of the discretization of the mass-flux interfaces.

3.2.1 Vertical grid transformation

In the vertical direction the grid transformation follows from the actual solution. In the computational domain all horizontal grid layers have thickness $\Delta\sigma = 1$ and σ simply equals the layer index k

$$\sigma_{k+1/2} = k + 1/2, \qquad\qquad \sigma_k = k \tag{3.31a}$$

For all active layers the vertical part of the transformation is given by

$$z(\tau^n, \xi_i, \eta_j, \sigma) = z_{i,j,k-1/2} + (\sigma - \sigma_{k-1/2})\Delta z^n_{i,j,k} \qquad \sigma_{k-1/2} \leqslant \sigma \leqslant \sigma_{k+1/2} \tag{3.31b}$$

$$z_\sigma \equiv \Delta z^n_{i,j,k} \tag{3.31c}$$

and follows directly from the vertical spacing of the local solution. At the non-integer indices the transformation is performed similarly. When solving the continuity equation the grid is made boundary fitted in time. For mass volumes the grid transformation satisfies

$$z_\tau(\tau, \xi, \eta, \sigma_{k+1/2}) = \frac{\partial \zeta}{\partial t} \qquad k = k^{i,j;n}_{\max}, \tau^n \leqslant \tau < \tau^{n+1} \tag{3.32}$$

whereas for the horizontal as well as the vertical momentum volumes a frozen grid geometry is used, satisfying

$$z_\tau \equiv 0 \tag{3.33}$$

Details are presented in the corresponding sections.

Remark Note that the volumes in computational space are simply rectangular and the discretization of the boundaries will follow in a natural way from the discretization of the coordinate transformation and therefore we omit further details here. Further details on the discretization of the bathymetry and the free-surface level can be found in sections 3.6

3.3 Discretization of the continuity equation

The continuity equation is discretized using mass volumes defined around the pressure points (see section 3.2). Integration of the continuity equation (3.14) over the finite-volumes $\mathcal{M}_{i,j,k}$ and application of the divergence theorem in $(\tau, \xi, \eta, \sigma)$-space leads to

$$\begin{aligned} F^{\mathcal{M}}(\xi_{i-1/2}, \eta_j, \sigma_k) - F^{\mathcal{M}}(\xi_{i-1/2}, \eta_j, \sigma_k) + G^{\mathcal{M}}(\xi_i, \eta_{j+1/2}, \sigma_k) - G^{\mathcal{M}}(\xi_i, \eta_{j-1/2}, \sigma_k) \\ + H^{\mathcal{M}}(\xi_i, \eta_j, \sigma_{k+1/2}) - H^{\mathcal{M}}(\xi_i, \eta_j, \sigma_{k-1/2}) = 0 \end{aligned}$$

where the flux functions are given by

$$\begin{aligned} F^{\mathcal{M}}(\xi, \eta, \sigma) &= \int_{\eta_{j-1/2}}^{\eta_{j+1/2}} \int_{\sigma_{k-1/2}}^{\sigma_{k+1/2}} z_\sigma \sqrt{g_{\eta\eta}} u d\eta d\sigma \\ G^{\mathcal{M}}(\xi, \eta, \sigma) &= \int_{\xi_{i-1/2}}^{\xi_{i+1/2}} \int_{\eta_{j-1/2}}^{\eta_{j+1/2}} z_\sigma \sqrt{g_{\xi\xi}} v d\xi d\sigma \\ H^{\mathcal{M}}(\xi, \eta, \sigma) &= \int_{\xi_{i-1/2}}^{\xi_{i+1/2}} \int_{\eta_{j-1/2}}^{\eta_{j+1/2}} \left[-z_\xi \sqrt{g_{\eta\eta}} u - z_\eta \sqrt{g_{\xi\xi}} v + \sqrt{g^*} w \right] d\xi d\eta \end{aligned}$$

The midpoint values at the interfaces are used as approximation for the surface integrals. In this way, by discretizing the transformed equations on the computational grid automatically a conservative flux formulation is obtained (recall that $\Delta\xi = \Delta\eta = \Delta\sigma = 1$).

All vertical boundaries are included in the derivatives with respect to ξ and η, discretization of these boundaries is straightforward. Any oblique or moving boundaries (i.e. bottom and free-surface) are included in the derivatives with respect to σ. For the fluxes in σ-direction several simplifications can be made. In the internal of the domain upper and lower cell faces are strictly horizontal, i.e. $z_\xi = z_\eta = 0$ and at the bottom and free-surface the respective kinematic conditions (3.17) and (3.16) can be substituted, altogether leading to

$$-z_\xi\sqrt{g_{\eta\eta}}u - z_\eta\sqrt{g_{\xi\xi}}v + \sqrt{g^*}w \equiv \sqrt{g^*}z_\tau, \qquad \sigma = \sigma_{k+1/2},\ k = k_{\max}^{i,j;n} \tag{3.34a}$$

$$-z_\xi\sqrt{g_{\eta\eta}}u - z_\eta\sqrt{g_{\xi\xi}}v + \sqrt{g^*}w \equiv \sqrt{g^*}w, \qquad \sigma = \sigma_{k+1/2},\ k_{\min}^{i,j} \leqslant k < k_{\max}^{i,j;n} \tag{3.34b}$$

$$-z_\xi\sqrt{g_{\eta\eta}}u - z_\eta\sqrt{g_{\xi\xi}}v + \sqrt{g^*}w \equiv 0, \qquad \sigma = \sigma_{k-1/2},\ k = k_{\min}^{i,j} \tag{3.34c}$$

The mass volumes are boundary-fitted in space and time hence it is possible to substitute the time derivative of the free-surface in the surface flux (3.34a), because

$$z_\tau(\tau,\xi,\eta,\sigma_{k+1/2}) = \frac{\partial\zeta}{\partial t}, \quad k = k_{\max}^{i,j;n}$$

In non-simplified form the discretization of the continuity equation is given by

$$D_\xi\left[\Delta z\sqrt{g_{\eta\eta}}u\right] + D_\eta\left[\Delta z\sqrt{g_{\xi\xi}}v\right] + D_\sigma\left[-z_\xi\sqrt{g_{\eta\eta}}u - z_\eta\sqrt{g_{\xi\xi}}v + \sqrt{g^*}w\right] = 0 \tag{3.35}$$

where we introduce the mass-flux difference operators

$$D_\xi\left[.\right] = \left[.\right]_{i+1/2,j,k} - \left[.\right]_{i-1/2,j,k} \tag{3.36a}$$

$$D_\eta\left[.\right] = \left[.\right]_{i,j+1/2,k} - \left[.\right]_{i,j-1/2,k} \tag{3.36b}$$

$$D_\sigma\left[.\right] = \left[.\right]_{i,j,k+1/2} - \left[.\right]_{i,j,k-1/2} \tag{3.36c}$$

As mentioned, at the free-surface a θ-scheme is used for the time integration of the continuity equation (denoted by θ_ζ) and for internal grid cells using a fully implicit scheme. In view of the above, for internal grid cells the discretization of the continuity equation is given by

$$D_\xi\left[\Delta z^n\sqrt{g_{\eta\eta}}u^{n+1}\right] + D_\eta\left[\Delta z^n\sqrt{g_{\xi\xi}}v^{n+1}\right] + D_\sigma\left[\sqrt{g^*}w^{n+1}\right] = 0 \tag{3.37a}$$

and for continuity cells at the free-surface the discretization becomes

$$\sqrt{g^*}_{i,j}\frac{\zeta_{i,j}^{n+1} - \zeta_{i,j}^n}{\Delta t} + D_\xi\left[\Delta z^n\sqrt{g_{\eta\eta}}u^{n+\theta_\zeta}\right] + D_\eta\left[\Delta z^n\sqrt{g_{\xi\xi}}v^{n+\theta_\zeta}\right] - \sqrt{g^*}_{i,j}w_{i,j,k-1/2}^{n+\theta_\zeta} = 0 \tag{3.37b}$$

for cells including the bottom the discretization becomes

$$D_\xi\left[\Delta z^n\sqrt{g_{\eta\eta}}u^{n+1}\right] + D_\eta\left[\Delta z^n\sqrt{g_{\xi\xi}}v^{n+1}\right] + \sqrt{g^*}_{i,j}w_{i,j,k+1/2}^{n+1} = 0 \tag{3.37c}$$

and for cells including both the bottom and free-surface the discretization becomes

$$\sqrt{g^*}_{i,j}\frac{\zeta_{i,j}^{n+1} - \zeta_{i,j}^n}{\Delta t} + D_\xi\left[\Delta z^n\sqrt{g_{\eta\eta}}u^{n+\theta_\zeta}\right] + D_\eta\left[\Delta z^n\sqrt{g_{\xi\xi}}v^{n+\theta_\zeta}\right] = 0 \tag{3.37d}$$

Note that in grid cells at the free-surface the vertical grid spacings are discretized at the old time level.

3.4 Discretization of the horizontal momentum equations

Only the implementation of the momentum equation in ξ-direction is presented. Due to grid-symmetry the momentum equations in η-direction can be treated similarly. For the horizontal momentum in ξ-direction the conservation law (3.15a) is integrated over the finite-volumes in computational space. The derivative with respect to τ can be put outside the integral since the region of integration is independent of τ. Application of the divergence theorem in $(\tau, \xi, \eta, \sigma)$-space then results in

$$\begin{aligned} \frac{\partial}{\partial\tau} \iiint\limits_{\mathcal{U}_{i+1/2,j,k}} J u d\xi d\eta d\sigma \\ + F^{\mathcal{U}}(\xi_{i+1}, \eta_j, \sigma_k) - F^{\mathcal{U}}(\xi_i, \eta_j, \sigma_k) + G^{\mathcal{U}}(\xi_{i+1/2}, \eta_{j+1/2}, \sigma_k) - G^{\mathcal{U}}(\xi_{i+1/2}, \eta_{j-1/2}, \sigma_k) \\ + H^{\mathcal{U}}(\xi_{i+1/2}, \eta_j, \sigma_{k+1/2}) - H^{\mathcal{U}}(\xi_{i+1/2}, \eta_j, \sigma_{k-1/2}) = RHS \end{aligned} \tag{3.38}$$

where

$$\begin{aligned} F^{\mathcal{U}}(\xi, \eta, \sigma) &= \int_{\eta_{j-1/2}}^{\eta_{j+1/2}} \int_{\sigma_{k-1/2}}^{\sigma_{k+1/2}} z_\sigma \sqrt{g_{\eta\eta}} (u^2 + p) d\eta d\sigma \\ G^{\mathcal{U}}(\xi, \eta, \sigma) &= \int_{\xi_i}^{\xi_{i+1}} \int_{\sigma_{k-1/2}}^{\sigma_{k+1/2}} z_\sigma \sqrt{g_{\xi\xi}} uv d\xi d\sigma \\ H^{\mathcal{U}}(\xi, \eta, \sigma) &= \int_{\xi_i}^{\xi_{i+1}} \int_{\eta_{j-1/2}}^{\eta_{j+1/2}} \left[-z_\tau \sqrt{g^*} u - z_\xi \sqrt{g_{\eta\eta}} (u^2 + p) - z_\eta \sqrt{g_{\xi\xi}} vu + \sqrt{g^*} wu \right] d\xi d\eta \end{aligned}$$

The integrals are approximated using the midpoint rule (recall that). Similarly, the volume integral in the time derivative is approximated by taking the midpoint of the volume. Because $\Delta\xi = \Delta\eta = \Delta\sigma = 1$ a finite-volume discretization is obtained by simply using a finite-difference discretization of the transformed momentum equation on the computational grid.

Several simplifications can be made for the fluxes in σ-direction. At the bottom the kinematic condition (3.17) applies and all internal interfaces are strictly horizontal and fixed in time (i.e. $z_\xi = z_\eta = z_\tau = 0$) hence the fluxes in σ-direction at the bottom reduce to

$$\begin{aligned} - z_\tau \sqrt{g^*} u - z_\xi \sqrt{g_{\eta\eta}} (u^2 + p) - z_\eta \sqrt{g_{\xi\xi}} vu + \sqrt{g^*} wu \equiv -z_\xi \sqrt{g_{\eta\eta}} p, \\ \sigma = \sigma_{k-1/2}, k = k_{\min}^{i+1/2,j} \end{aligned} \tag{3.39a}$$

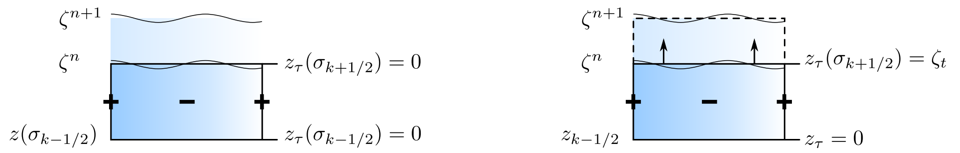
and at internal interfaces reduce to

$$\begin{aligned} - z_\tau \sqrt{g^*} u - z_\xi \sqrt{g_{\eta\eta}} (u^2 + p) - z_\eta \sqrt{g_{\xi\xi}} vu + \sqrt{g^*} wu \equiv \sqrt{g^*} wu, \\ \sigma = \sigma_{k+1/2}, k_{\min}^{i+1/2,j} \leqslant k < k_{\max}^{i+1/2,j;n} \end{aligned} \tag{3.39b}$$

The flux in σ-direction in the free-surface layer is directly related to the method of time integration and is discussed below.

3.4.1 Time evolution of the horizontal momentum volumes

A choice needs to be made for the time discretization of the volume sizes $V_{i+1/2,j}$ and $V_{i,j+1/2}$ used in the momentum equations. Choices made for the discretization of the time derivative are related to the discretization of the convection and the pressure gradient at the free-surface and have consequences for momentum-conservation properties.



(a) Frozen grid-geometry

(b) Locally boundary-fitted grid-geometry

Figure 3.4: A visualization of the grid transformation in the vertical direction for a frozen momentum volume and a locally boundary-fitted momentum volume.

Frozen grid geometry

It is possible to use a frozen grid geometry in which the volume sizes are only updated after full completion of each time step while during the time step $z_\tau = 0$ everywhere (figure 3.4(a)). It follows that the volume size does not change in time, so the time derivative can be discretized as

$$\frac{\partial Ju}{\partial \tau} \approx V^n \frac{u^{n+1} - u^n}{\Delta t} \tag{3.40}$$

In the surface layer the vertical component of the grid transformation is given by

$$z(\tau, \xi_i, \eta_j, \sigma) = z^n_{i+1/2,j,k+1/2} + (\sigma - \sigma_{k-1/2})\Delta z^n_{i+1/2,j,k} \qquad k = k^{i,j;n}_{\max}, \tau^n \leqslant \tau \leqslant \tau^{n+1} \tag{3.41}$$

It follows that the upper faces of surface cells ($\sigma = \sigma_{k+1/2}$) only coincide with the free-surface at the beginning of the time step. Since $z_\tau = 0$ the flux in σ-direction through the top interfaces of grid cells in the surface layer reduces to

$$\begin{aligned} &- z_\tau \sqrt{g^*}u - z_\xi \sqrt{g_{\eta\eta}}(u^2 + p) - z_\eta \sqrt{g_{\xi\xi}} vu + \sqrt{g^*}wu \equiv \\ &\qquad\qquad - z_\xi \sqrt{g_{\eta\eta}}(u^2 + p) - z_\eta \sqrt{g_{\xi\xi}} vu + \sqrt{g^*}wu, \\ &\qquad\qquad\qquad\qquad \sigma = \sigma_{k+1/2},\ k = k^{i+1/2,j;n}_{\max} \end{aligned} \tag{3.42}$$

Furthermore it is seen that it is not possible to substitute the kinematic condition. For surface cells a discretization needs to be found for the *grid*-surface fluxes (3.42) in σ-direction. Note that these can no longer truly be called free-surface fluxes. It has also to be remarked that the atmospheric pressure condition (1.6) can no longer be applied and some approximation for the pressure at the upper volume faces is needed, details are presented in section 3.8. Approximation of the convective part of the grid-surface flux is not straightforward and will be discussed in section 3.9.

Locally boundary-fitted grid

It is also possible to use a locally boundary-fitted grid in which the volumes in the top layer change along with the moving free-surface (figure 3.4(b)). In that case the grid transformation is assumed to satisfy

$$\begin{aligned} z(\tau, \xi_i, \eta_j, \sigma) = z_{i+1/2,j,k-1/2} + (\sigma - \sigma_{k-1/2})(\zeta(\tau, x_{i+1/2,j}, y_{i+1/2,j}) - z_{i+1/2,j,k-1/2}), \\ \sigma_{k-1/2} \leqslant \sigma \leqslant \sigma_{k+1/2},\ k = k^{i+1/2,j;n}_{\max}, \tau^n \leqslant \tau \leqslant \tau^{n+1} \end{aligned} \tag{3.43}$$

If the transformation is made boundary-fitted at the free-surface, the kinematic condition (3.16) and the atmospheric pressure condition (1.6) can be applied at the upper faces of the surface cells, leading to the simplified *free-surface*-fluxes

$$-z_\tau\sqrt{g^*}u - z_\xi\sqrt{g_{\eta\eta}}(u^2+p) - z_\eta\sqrt{g_{\xi\xi}}vu + \sqrt{g^*}wu\Big|_{\sigma=\sigma_{k+1/2}} \equiv -z_\xi\sqrt{g_{\eta\eta}}p_a\big|_{\sigma=\sigma_{k+1/2}}, \qquad k = k_{\max}^{i+1/2,j;n} \tag{3.44}$$

Since the volume dimensions change in time, the volume size cannot be put outside the time derivative. An appropriate discretization is given by

$$\frac{\partial Ju}{\partial\tau} = \sqrt{g^*}\frac{\partial z_\sigma u}{\partial\tau} \approx \sqrt{g_{\xi\xi}}\frac{q^{n+1}-q^n}{\Delta t} \tag{3.45}$$

introducing the mass-flux variable $q = \sqrt{g_{\eta\eta}}z_\sigma u$ (more details are provided in appendix A).

Summarizing, two time integration methods for the momentum equations have been presented

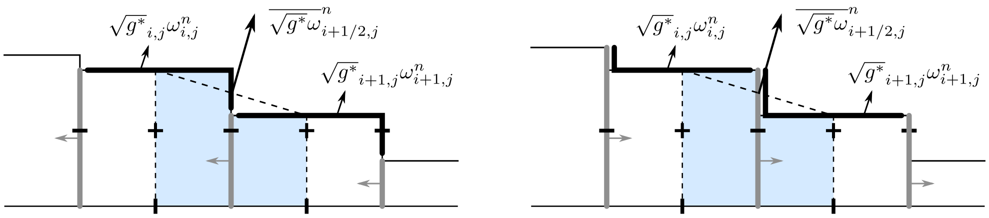
- *Frozen-grid geometry.* The time integration method can be seen as a predictor-corrector method in the folowing sense. First the grid-geometry is frozen during the time step. After having determined the new water levels and velocities by solving the pressure-correction equations and momentum-correction equations, the z-layer administration is updated to fit the new water levels and the new solution is obtained by constant extrapolation of the velocity to the new volume dimensions. In this way the momentum equations can easily be expressed in velocity and water level variables and the change of the water level is accounted for by discretization of the grid-surface flux (3.42).
- *Locally boundary-fitted geometry.* Using this approach it is possible to substitute the kinematic and atmospheric pressure condition at the free-surface. It is not necessary anymore to discretize a flux through the upper faces of the surface cells. The time derivative preferably needs to be in conservative formulation and the most natural choice is a discretization in terms of mass-flux variables incorporating the change in volume size (see appendix A).

In the present work the finite volume method is based on a frozen-grid geometry. This method best fits the prediction-correction solution procedure as used in the existing non-hydrostatic version of Delft3D-FLOW. Implementing a locally boundary-fitted grid asks for some thorough adaptations in the existing discretization of the pressure correction equations as well as the momentum prediction equations. Nonetheless a boundary fitted approach has some clear advantages over the frozen-grid approach, in particular for momentum conservation properties, and stability and accuracy of the time-integration. A discussion is given in chapter A.

3.4.2 Discretization of the grid-surface flux

For the discretization of the convection flux (3.42) in σ-direction through the upper faces of surface cells we need an approximation of the horizontal velocity u at the free surface, denoted by u_ζ and an approximation of the velocity normal to the grid surface, which for brevity will be denoted by $\sqrt{g^*}\omega$,

$$\sqrt{g^*}\omega := -z_\xi\sqrt{g_{\eta\eta}}u - z_\eta\sqrt{g_{\xi\xi}}v + \sqrt{g^*}w\Big|_{\sigma=\sigma_{k+1/2}}, \qquad k = k_{\max}^{i+1/2,j;n}$$


$\sqrt{g^*}_{i,j}\omega^n_{i,j}$
$\overline{\sqrt{g^*}\omega}^n_{i+1/2,j}$
$\sqrt{g^*}_{i+1,j}\omega^n_{i+1,j}$
$\sqrt{g^*}_{i,j}\omega^n_{i,j}$
$\overline{\sqrt{g^*}\omega}^n_{i+1/2,j}$
$\sqrt{g^*}_{i+1,j}\omega^n_{i+1,j}$

(a) Flow from the right

(b) Flow from the left

Figure 3.5: Illustration of the approximation of the grid-surface flux for horizontal momentum volumes. The interface of the mass fluxes through the free-surface are indicated with thick black lines, the mass flux interfaces with gray lines and the interface of the grid surface flux for the horizontal momentum volume with a dashed line.

An approximation can be obtained by means of an explicit discretization of the continuity equation at the current time level (compare (3.35)).

$$\sqrt{g^*}\omega^n_{i,j} = -D_\xi\left[\sqrt{g_{\eta\eta}}\Delta z^n u^n\right] - D_\eta\left[\sqrt{g_{\xi\xi}}\Delta z^n v^n\right] + \sqrt{g^*}_{i,j}w^n_{i,j,k-1/2} \tag{3.46}$$

So far only the surface velocities at the integer grid indices have been obtained. The free-surface velocities at the non-integer grid indices follow from arithmetic averaging (illustrated in figure 3.5(a)):

$$\begin{aligned}\overline{\sqrt{g^*}\omega}^n_{i+1/2,j,k+1/2} &= \frac{1}{2}\left(\sqrt{g^*}_{i,j}\omega^n_{i,j,k+1/2} + \sqrt{g^*}_{i+1,j}\omega^n_{i+1,j,k+1/2}\right) \\ &= \frac{\overline{\sqrt{g^*}\zeta}^n_{i+1/2,j} - \overline{\sqrt{g^*}\zeta}^{n-1}_{i+1/2,j}}{\Delta t} = \frac{V^n_{i+1/2,j} - V^{n-1}_{i+1/2,j}}{\Delta t}\end{aligned} \tag{3.47}$$

where $\overline{\sqrt{g^*}\zeta}_{i+1/2,j} = \left(\sqrt{g^*}_{i+1,j}\zeta_{i+1,j} + \sqrt{g^*}_{i,j}\zeta_{i,j}\right)/2$.

To complete the discretization of the grid surface flux an approximation of u_ζ is needed. Because no information is available about the horizontal velocity outside the free-surface it is not possible to obtain an upwind discretization of the convection. The horizontal velocity is approximated by means of constant extrapolation, leading to the discretized *grid-surface*-flux:

$$\overline{\sqrt{g^*}\omega}^n_{i+1/2,j}u^n_{i+1/2,j,k} \qquad\qquad k = k^{i+1/2,j;n}_{\max} \tag{3.48}$$

Discussion In the finite-difference approach used in the existing version of Delft3D-FLOW the vertical convection is only discretized in case of outflow. If the vertical velocity in the surface cells points inwards the convection is set to zero, which is not a realistic assumption. The finite volume formulation of the vertical convection on the other hand also includes a discretization in case of inwards pointing flow. Furthermore the finite volume flux is discretized at the free-surface and is based on the velocity normal to the free-surface whereas this geometric aspect is not included in the finite-difference discretization. For these reasons the finite-volume discretization is expected to give a more accurate approximation of convection at the free-surface. The boundary conditions are already incorporated at in the analytical derivation of the finite volume method so at the discrete level the boundary condition does not need to be discretized explicitly anymore.

The accuracy of the discretization of the grid-surface fluxes depends on the accuracy of the discretization of the continuity equation. Since the size of the mass-flux interfaces is approximated with $O(\Delta x)$ accuracy the discretization of the free-surface fluxes $\omega_{i,j}$ is also only $O(\Delta x^2)$ accurate. From this it follows that the discretization of the grid-surface fluxes is also $O(\Delta x)$ accurate. This can be improved by obtaining a better approximation of the size of the mass-flux interfaces by means of a higher order reconstruction of the water levels at the non-integer indices.

For the approximation of the horizontal velocity u_ζ at the free surface constant extrapolation is used. For outwards flow this comes down to an upwind discretization, but for inwards flow this comes down to a downwind discretization and this has a negative effect on the stability of the numerical scheme. It is however more accurate to use constant extrapolation than to set the grid-surface flux to zero which comes down to assuming that the horizontal velocity at the free-surface equals zero.

Momentum conservation in time By adapting the discretization of ω it is possible to obtain a grid-surface flux which results in a momentum-conservative discretization in time. By setting

$$\sqrt{g^*}\omega^n_{i,j} = -D_\xi\left[\sqrt{g_{\eta\eta}}\Delta z^n u^{n-1+\theta_\zeta}\right] - D_\eta\left[\sqrt{g_{\xi\xi}}\Delta z^n v^{n-1+\theta_\zeta}\right] + \sqrt{g^*}_{i,j} w^{n-1+\theta_\zeta}_{i,j,k-1/2} \tag{3.49}$$

the discretization of the surface velocity is identical to the discretization used for time integration of the continuity equation at the previous time level, hence from equation (3.37b) (or (3.37d) similarly) it follows that

$$\sqrt{g^*}_{i,j}\omega^n_{i,j} = -D_\xi\left[u^n\right] - D_\eta\left[v^n\right] + \sqrt{g^*}_{i,j} w^n_{i,j,k-1/2} = \sqrt{g^*}_{i,j}\frac{\zeta^n_{i,j} - \zeta^{n-1}_{i,j}}{\Delta t} \tag{3.50}$$

The discretization of the convective part of the grid surface flux (3.42) effectively becomes

$$\begin{aligned}\overline{\sqrt{g^*}\omega}_{i+1/2,j,k+1/2} u_{i+1/2,j,k} = u_{i+1/2,j,k}&\frac{\overline{\sqrt{g^*}\zeta}^n_{i+1/2,j} - \overline{\sqrt{g^*}\zeta}^{n-1}_{i+1/2,j}}{\Delta t}\\ &= \frac{V^n_{i+1/2,j,k} - V^{n-1}_{i+1/2,j,k}}{\Delta t} \approx u\frac{\partial J}{\partial t}\end{aligned}$$

hence altogether a discretization is obtained of the conservative time-derivative in which the grid geometry is lagging behind one time step

$$V^n_{i+1/2,j,k}\frac{u^{n+1} - u^n}{\Delta t} + u^n\frac{V^n_{i+1/2,j,k} - V^{n-1}_{i+1/2,j,k}}{\Delta t}\frac{V^n_{i+1/2,j,k}u^{n+1} - V^{n-1}_{i+1/2,j,k}u^n}{\Delta t} \approx \frac{\partial Ju}{\partial t}$$

A numerical scheme is obtained which is now also momentum-conservative in time, with the volume integrated momentum defined as

$$V^{n-1}_{i+1/2,j}u^n_{i+1/2,j}$$

It is important to observe that the discretization is only truly conservative if at the discrete level the volume size $V_{i+1/2,j,k}$ equals the arithmetic average of the neighbouring mass volume sizes $V_{i,j,k}$ and $V_{i+1,j,k}$, which is cf. (3.30). This discretization method is comparable to the one described in [12],[13]. However in the currently presented finite-volume method the spatially conservative discretization is made fully conservative, i.e. also in time, by choosing an appropriate discretization of the grid-surface flux. (In chapter 2 a similar method could be used in order to obtain a fully momentum-conservative adaptation of the SD-scheme.)

Because the adapted discretization of the grid-surface flux is not only explicit but also based on the geometry of the previous time level it can negatively influence the stability of the time-integration. It has been tried to implement the adapted discretization (3.49) in Delft3D-FLOW but this led to an unstable scheme.

3.5 Discretization of the vertical momentum equation

For the vertical momentum the conservation law (3.15c) is integrated across the finite-volumes in computational space. The derivative with respect to τ can be put outside the integral since the region of integration is independent of τ. Application of the divergence theorem in $(\tau, \xi, \eta, \sigma)$-space then results in

$$\begin{aligned}\frac{\partial}{\partial\tau}\iiint\limits_{\mathcal{W}_{i,j,k+1/2}} Jw\,d\xi d\eta d\sigma + F^{\mathcal{W}}(\xi_{i+1/2}, \eta_j, \sigma_{k+1/2}) - F^{\mathcal{W}}(\xi_{i-1/2}, \eta_j, \sigma_{k+1/2})\\ + G^{\mathcal{W}}(\xi_i, \eta_{j+1/2}, \sigma_{k+1/2}) - G^{\mathcal{W}}(\xi_i, \eta_{j-1/2}, \sigma_{k+1/2})\\ + H^{\mathcal{W}}(\xi_i, \eta_j, \sigma_{k+1}) - H^{\mathcal{W}}(\xi_i, \eta_j, \sigma_k) = RHS \quad (3.51)\end{aligned}$$

where

$$\begin{aligned}F^{\mathcal{W}}(\xi,\eta,\sigma) &= \int_{\eta_{j-1/2}}^{\eta_{j+1/2}}\int_{\sigma_k}^{\sigma_{k+1}} z_\sigma\sqrt{g_{\eta\eta}}uw\,d\eta d\sigma\\ G^{\mathcal{W}}(\xi,\eta,\sigma) &= \int_{\xi_{i-1/2}}^{\xi_{i+1/2}}\int_{\sigma_k}^{\sigma_{k+1}} z_\sigma\sqrt{g_{\xi\xi}}vw\,d\xi d\sigma\\ H^{\mathcal{W}}(\xi,\eta,\sigma) &= \int_{\xi_{i-1/2}}^{\xi_{i+1/2}}\int_{\eta_{j-1/2}}^{\eta_{j+1/2}} \left[-z_\tau\sqrt{g^*}w - z_\xi\sqrt{g_{\eta\eta}}uw - z_\eta\sqrt{g_{\xi\xi}}vw + \sqrt{g^*}(w^2+p)\right]d\xi d\eta\end{aligned}$$

The integrals are approximated at the midpoints of the interfaces and a discretization is made of the following fluxes. The volume integral in the time derivative is approximated by taking the midpoint of the volume. A finite-volume discretization is obtained by discretizing the transformed equation (3.15c) on the computational grid in $(\tau, \xi, \eta, \sigma)$-space (recall that $\Delta\xi = \Delta\eta = \Delta\sigma = 1$). Several simplifcations are made for the fluxes in σ-direction (compare section 3.4). For internal interfaces the fluxes reduce to

$$\begin{aligned}- z_\tau\sqrt{g^*}w - z_\xi\sqrt{g_{\eta\eta}}uw - z_\eta\sqrt{g_{\xi\xi}}vw + \sqrt{g^*}(w^2+p) \equiv \sqrt{g^*}(w^2+p),\\ \sigma = \sigma_k,\ k_{\min}^{i,j} + 1 \leqslant k < k_{\max}^{i,j;n} \quad (3.52)\end{aligned}$$

As already mentioned the upper and lower interfaces of the vertical momentum volumes are located at the centers of the mass volumes (both in computational and physical grid space). In this way the vertical momentum volumes do not reach up to the bottom and free-surface resulting in the presence of vertical fluxes which do not point into another volume. Therefore full conservation of momentum is difficult to achieve. The discretization of the convection and pressure fluxes is described in sections 3.9 and 3.8 respectively.

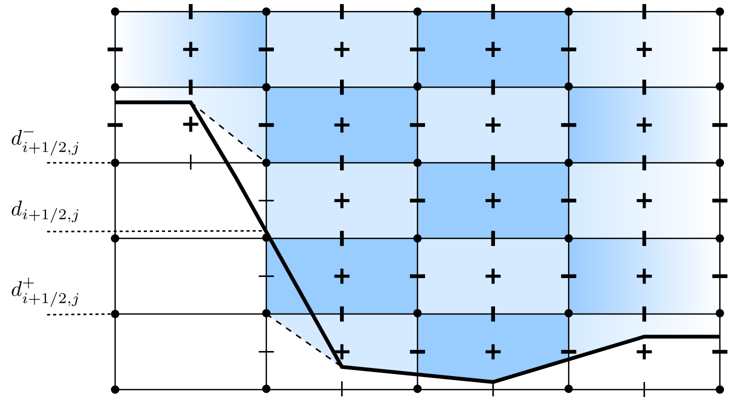


Figure 3.6: Reconstruction of the water depth in the velocity points. The thick line gives a piecewise linear interpolation of the bed levels $\{\dots, -d_i, -d_{i+1/2}, -d_{i+1}, \dots\}$. The dashed lines indicate the size of the mass volumes.

3.6 Reconstruction of bathymetry and free-surface

The bottom and free-surface levels are only provided at the centers of the mass volumes, at the non-integer indices they need to be reconstructed by some interpolation method. In this section we describe the existing procedure for the reconstruction of the free-surface level and have a look at a new reconstruction of the bottom level.

First let us have a look at the reconstruction of the bed level. The height of the bottom layer is only available in the water level points. At non-integer indices a reconstruction is needed, which should give an approximation of

$$\Delta z = \min(\zeta(x_{i+1/2,j}, y_{i+1/2,j}, t^n), z_{k+1/2}) + d(x_{i+1/2,j}, y_{i+1/2,j}), \qquad k = k_{\text{min}}^{i+1/2,j} \tag{3.53}$$

Only the second term at the right hand side needs our consideration. The bottom level is currently discretized as piecewise constant using a minimum approach for the depth at the non-integer indices (see [16], p. 18).

$$d_{i+1/2,j} = \min(d_{i+1,j}, d_{i,j})$$

The discretization of the bottom level was originally designed for hydrostatic shallow-water equations. For the hydrostatic case the bottom gradient is only of small importance and for this reason only a piecewise constant approximation was used. A short analysis shows that this results in first-order error terms in the discretization of the mass fluxes in the continuity equation. This truncation error can be interpreted as the addition of thin vertical walls.

In the non-hydrostatic case it can be worthwile to use an improved reconstruction of the depth values. On uniform Cartesian grids it is easily shown that the accuracy can be improved by using linear interpolation for the water depth. In this way second-order accuracy is obtained as long as the bottom remains within a single layer.

$$d_{i+1/2,j} = \frac{1}{2}\left(d_{i,j} + d_{i+1,j}\right) \tag{3.54}$$

In this approach the slope of the bottom level needs to be bounded so that sloped boundaries only occur in the lowest grid layer. This condition cannot always be met. Too steep slopes

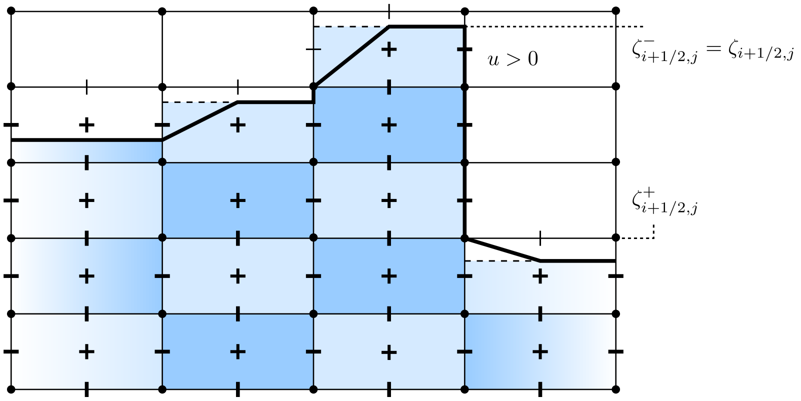


Figure 3.7: Reconstruction of the free-surface levels in the velocity points. The thick line gives a piecewise linear interpolation of the water levels $\{\ldots, \zeta_i, \zeta_{i+1/2}, \zeta_{i+1}, \ldots\}$. The dashed lines indicate the size of the mass volumes.

are therefore replaced by vertical sections (see figures 3.6 and 3.7). At vertical boundaries the bottom (and free-surface) are therefore not uniquely defined. The left and right bottom levels are obtained using

$$d^+_{i+1/2,j} = \begin{cases} \min(-z_{k-1/2}, d_{i+1/2,j}), & k = k^{i+1,j}_{\min} > k^{i,j}_{\min} \\ \max(-z_{k+1/2}, d_{i+1/2,j}), & k = k^{i+1,j}_{\min} < k^{i,j}_{\min} \\ d_{i+1/2,j}, & k^{i+1,j}_{\min} = k^{i,j}_{\min} \end{cases} \tag{3.55a}$$

$$d^-_{i+1/2,j} = \begin{cases} \min(-z_{k-1/2}, d_{i+1/2,j}), & k = k^{i,j}_{\min} > k^{i+1,j}_{\min} \\ \max(-z_{k+1/2}, d_{i+1/2,j}), & k = k^{i,j}_{\min} < k^{i+1,j}_{\min} \\ d_{i+1/2,j}, & k^{i+1,j}_{\min} = k^{i,j}_{\min} \end{cases} \tag{3.55b}$$

This is done similarly in the η-direction. Reconstructing too steep slopes in this manner results in a loss of accuracy. More advanced cut-cell techniques would be needed to resolve this problem. For applications in which the bottom needs to be modeled accurately, small grid spacing is needed.

Also at the free-surface an approximation of the layer thickness is needed at the non-integer indices. The height of the top layer should give an accurate approximation of

$$\Delta z = \zeta(x_{i+1/2,j}, y_{i+1/2,j}, t^n) - \max(z_{k-1/2}, -d(x_{i+1/2,j}, y_{i+1/2,j}) \qquad k = k^{i+1/2,j;n}_{\max} \tag{3.56}$$

For stability reasons a piecewise constant reconstruction of the free-surface is used, as was already done in the existing model. Depending on the flow direction in the surface layer a first-order upwind reconstruction of the water level is used.

$$\zeta_{i+1/2,j} = \begin{cases} \zeta_{i+1,j}, & u_{i+1/2,j,k^{i+1/2,j;n}_{\max}} < 0 \\ \zeta_{i,j}, & u_{i+1/2,j,k^{i+1/2,j;n}_{\max}} > 0 \\ \frac{1}{2}\left(\zeta_{i,j} + \zeta_{i+1,j}\right), & u_{i+1/2,j,k^{i+1/2,j;n}_{\max}} = 0 \end{cases} \tag{3.57a}$$

and in the η-direction similarly. As we will see later, also values directly at the left and right side of the discontinuities are needed. Because the reconstruction is piecewise constant this amounts to

$$\zeta^+_{i+1/2,j} = \zeta_{i+1,j}, \qquad \zeta^-_{i+1/2,j} = \zeta_{i,j} \tag{3.57b}$$

Finally it is remarked that a velocity variable can also become active if it is located at vertical sections of the free-surface. Whether the velocity variable is active follows from (3.24) which depends on the local water level $\zeta_{i+1/2}$. In figure 3.7 active and inactive variables are indicated with thick and thin dashes respectively.

3.7 Vertical spacing of the control volumes

After having discretized the bottom and free-surface level we now have a look at the vertical size of the control volumes and the size of the open flux interfaces which will be used for the definition of the numerical fluxes.

The geometry of the horizontal momentum volumes follows from the filled height of the kth z-layers. For partially filled volumes the local upper and lower boundaries of the kth layer are given by

$$z^\pm_{i+1/2,j,k+1/2} = \min(z_{k+1/2}, \zeta^\pm_{i+1/2,j}) \tag{3.58a}$$

$$z^\pm_{i+1/2,j,k-1/2} = \max(z_{k+1/2}, -d^\pm_{i+1/2,j}) \tag{3.58b}$$

$$z^\pm_{i+1/2,j,k} = \frac{1}{2}\left(z^\pm_{i+1/2,j,k-1/2} + z^\pm_{i+1/2,j,k+1/2}\right) \tag{3.58c}$$

inside active horizontal momentum volumes the local wetted layer height is then defined as

$$\Delta z^\pm_{i+1/2,j,k} = z^\pm_{i,j,k+1/2} - z^\pm_{i,j,k-1/2}, \tag{3.59a}$$

$$\tag{3.59b}$$

The filled layer height at the interface between two active volumes is obtained by

$$\Delta z_{i+1/2,j,k} = \max\left(0, \min(z^+_{i+1/2,j,k+1/2}, z^-_{i+1/2,j,k+1/2}) - \max(z^+_{i+1/2,j,k-1/2}, z^-_{i+1/2,j,k-1/2})\right) \tag{3.60a}$$

The open flux interfaces of the volumes are discretized as rectangles. Variations of the layer-thickness in ξ-direction (for fluxes parallel to the η-lines) and in η-direction (for fluxes parallel to the ξ-axis) are ignored and this introduces higher order truncation errors which can be neglected.

The geometry of the vertical momentum volumes follows from the filled height of the staggered $(k \pm 1/2)$th layers.

$$\Delta z^\pm_{i+1/2,j,k+1/2} = z^\pm_{i+1/2,j,k+1/2} - z^\pm_{i+1/2,j,k-1/2} \tag{3.61}$$

The filled height inside the vertical momentum volumes becomes

$$\Delta z^\pm_{i+1/2,j,k+1/2} = z^\pm_{i+1/2,j,k+1} - z^\pm_{i+1/2,j,k}, \tag{3.62a}$$

$$\Delta z_{i,j,k+1/2} = z_{i,j,k+1} - z_{i,j,k} \tag{3.62b}$$

The filled heights at the interface between two neighbouring volumes is then obtained using

$$\Delta z_{i+1/2,j,k+1/2} = \max\left(0, \min(z^+_{i+1/2,j,k+1}, z^-_{i+1/2,j,k+1}) - \max(z^+_{i+1/2,j,k}, z^-_{i+1/2,j,k})\right) \tag{3.63a}$$

3.8 Discretization of the pressure gradient

Pressure gradient in the horizontal momentum equations For the presentation of some details we consider only the momentum equation in ξ-direction. Similar principles are applied for the pressure derivatives in the η-direction. The pressure gradient in ξ-direction is given by (see (3.15a))

$$-\frac{\partial \sqrt{g_{\eta\eta}}z_{\sigma}p}{\partial \xi} + z_{\sigma}p\frac{\partial \sqrt{g_{\eta\eta}}}{\partial \xi} + \frac{\partial \sqrt{g_{\eta\eta}}z_{\xi}p}{\partial \sigma} \tag{3.64}$$

where we have omitted the density. For internal grid cells the horizontal interfaces are flat ($z_{\xi} = z_{\eta} = 0$) hence the last term in (3.64) vanishes. The conservative first term is discretized as

$$-\Delta z_{i+1,j,k}\sqrt{g_{\eta\eta}}_{i+1,j,k}p_{i+1,j,k} + \Delta z_{i,j,k}\sqrt{g_{\eta\eta}}_{i,j,k}p_{i,j,k} \tag{3.65}$$

the remaining non-conservative second term is discretized as

$$\overline{\Delta z p}_{i+1/2,j,k}\left[\sqrt{g_{\eta\eta}}_{i+1,j,k} - \sqrt{g_{\eta\eta}}_{i,j,k}\right] \tag{3.66}$$

where the bar is used to denote central averaging

$$\overline{\Delta z p}_{i+1/2,j,k} = \frac{1}{2}\left(\Delta z_{i,j,k}p_{i,j,k} + \Delta z_{i+1,j,k}p_{i+1,j,k}\right) \tag{3.67}$$

Adding up (3.65) and (3.66) the discretization becomes

$$-\overline{\sqrt{g_{\eta\eta}}}\left[\Delta z_{i+1,j,k}p_{i+1,j,k} - \Delta z_{i,j,k}p_{i,j,k}\right] \tag{3.68}$$

where

$$\overline{\sqrt{g_{\eta\eta}}}_{i+1/2,j,k} = \frac{1}{2}\left(\sqrt{g_{\eta\eta}}_{i,j,k} + \sqrt{g_{\eta\eta}}_{i+1,j,k}\right) \tag{3.69}$$

A similar procedure can be used for discretization of the pressure gradient in bottom grid cells, however now also variations of the bottom level need to be taken into account. As an example let us consider grid cells that only contain the bottom and not the free-surface. The third term of (3.64) now describes the pressure flux through the bottom boundary. This term is discretized as

$$\overline{\sqrt{g_{\eta\eta}}}_{i+1/2,j,k}\overline{p}_{i+1/2,j,k}\left[\Delta z_{i+1} - \Delta z_{i}\right] \tag{3.70}$$

where again the missing pressure point is obtained using central averaging. Together with (3.65) and (3.66) the pressure discretization becomes

$$-\overline{\sqrt{g_{\eta\eta}}}_{i+1/2,j,k}\overline{\Delta z}_{i+1/2,j,k}\left[p_{i+1} - p_{i}\right] \tag{3.71}$$

It follows that both in the case of internal and bottom grid cells the discretization almost comes down to using a finite-difference approach with the only difference being a multiplication factor. For simulations on non-curvilinear grids this discretization is fully consistent with physical momentum conservation laws at the boundary.

In free-surface grid cells it is not possible to use the atmospheric pressure condition since the grid transformation is not boundary fitted in time. The same discretization approach described here for bottom grid cells can also be applied in the free-surface layer.

Pressure gradient in the vertical momentum equation The pressure term in (3.15c) is already conservative and is simply discretized as

$$\frac{\partial \sqrt{g^*} p}{\partial \sigma} \approx \sqrt{g^*}_{i,j,k+1} p_{i,j,k+1} - \sqrt{g^*}_{i,j,k} p_{i,j,k} \tag{3.72}$$

Furthermore none of the vertical momentum volumes has interfaces in σ-direction coinciding with horizontal boundaries of the domain and therefore no pressure fluxes across boundaries need to be discretized.

Pressure correction equations If the coefficients in the discretization of the pressure gradient are changed in the momentum prediction equations this also needs to be done in the momentum correction equations. Changing the discretization of the pressure gradient therefore also leads to different coefficients in the pressure-correction equations. However, the changes are small and the structure of the pressure-correction matrix remains the same. Therefore the solution procedures as described in [16] can be used without significant performance loss.

3.9 First order upwind convection

The convection fluxes are discretized using a first-order explicit upwind scheme. Special attention needs to be paid to different boundary situations of which a full discussion is given in section 3.10

3.9.1 Horizontal momentum equations

Only the flux discretization for $\mathcal{U}$ volumes is discussed. Due to symmetric arrangement of the variables the fluxes for $\mathcal{V}$ momentum volumes can be discretized similarly. The velocities at the interfaces are obtained by linear interpolation. The transported quantity is determined in an upwind manner. In ξ-direction the discretization of the convection term is given by

$$\frac{\partial \sqrt{g_{\eta\eta}} z_\sigma u^2}{\partial \xi} \approx \Delta z \sqrt{g_{\eta\eta}} \overline{u} \overset{*}{u} \Big|_{i+1,j,k} - \Delta z \sqrt{g_{\eta\eta}} \overline{u} \overset{*}{u} \Big|_{i,j,k} \tag{3.73}$$

where

$$\overset{*}{u}_{i,j,k} = \begin{cases} u_{i-1/2,j,k}, & \overline{u} > 0, \\ u_{i+1/2,j,k}, & \overline{u} \leqslant 0 \end{cases}, \qquad \overline{u}_{i,j,k} = \frac{1}{2} \left(u_{i-1/2,j,k} + u_{i+1/2,j,k} \right)$$

in η-direction the discretization is given by

$$\frac{\partial \sqrt{g_{\xi\xi}} z_\sigma v u}{\partial \eta} \approx {}^{(\eta)}\overline{\sqrt{g_{\eta\eta}} \Delta z} \, \overline{v} \overset{*}{u} \Big|_{i+1/2,j+1/2,k} - {}^{(\eta)}\overline{\sqrt{g_{\eta\eta}} \Delta z} \, \overline{v} \overset{*}{u} \Big|_{i+1/2,j-1/2,k} \tag{3.74}$$

where

$$\overset{*}{u}_{i+1/2,j+1/2,k} = \begin{cases} u_{i+1/2,j,k}, & \overline{v} > 0, \\ u_{i+1/2,j+1,k}, & \overline{v} \leqslant 0 \end{cases}, \qquad \overline{v}_{i+1/2,j+1/2,k} = \frac{1}{2} \left(v_{i,j+1/2,k} + v_{i+1,j+1/2,k} \right)$$

finally the discretization of the vertical convection fluxes is given by

$$\frac{\partial\sqrt{g^*}wu}{\partial\sigma} \approx \overline{w}\sqrt{g^*}\overset{*}{u}\Big|_{i+1/2,j,k+1/2} - \overline{w}\sqrt{g^*}\overset{*}{u}\Big|_{i+1/2,j,k-1/2} \tag{3.75}$$

where

$$\overset{*}{u}_{i+1/2,j,k+1/2} = \begin{cases} u_{i+1/2,j,k}, & \overline{w} > 0, \\ u_{i+1/2,j,k+1}, & \overline{w} \leqslant 0 \end{cases}, \quad \overline{w}_{i+1/2,j,k+1/2} = \frac{1}{2}\left(w_{i,j,k+1/2} + w_{i+1,j,k+1/2}\right)$$

3.9.2 Vertical momentum equation

The flux interfaces of the vertical momentum volumes are all exactly centered between the mass flux interfaces of the continuity grid cells. The velocities at the interfaces are obtained by linear interpolation and the transported quantity is determined in an upwind manner.

The discretization of the convection term in ξ-direction is given by

$$\frac{\partial\sqrt{g_{\eta\eta}}uw}{\partial\xi} \approx \sqrt{g_{\eta\eta}}\Delta z\overline{u}\overset{*}{w}\Big|_{i+1/2,j,k+1/2} - \sqrt{g_{\eta\eta}}\Delta z\overline{u}\overset{*}{w}\Big|_{i-1/2,j,k+1/2} \tag{3.76}$$

where

$$\overset{*}{w}_{i+1/2,j,k+1/2} = \begin{cases} w_{i,j,k+1/2}, & \overline{u} > 0, \\ w_{i+1,j,k+1/2}, & \overline{u} \leqslant 0 \end{cases}, \quad \overline{u}_{i+1/2,j,k+1/2} = \frac{1}{2}\left(u_{i+1/2,j,k} + u_{i+1/2,j,k+1}\right)$$

This is done similarly in η-direction

$$\frac{\partial\sqrt{g_{\xi\xi}}vw}{\partial\eta} \approx \sqrt{g_{\xi\xi}}\Delta z\overline{v}\overset{*}{w}\Big|_{i,j+1/2,k+1/2} - \sqrt{g_{\xi\xi}}\Delta z\overline{v}\overset{*}{w}\Big|_{i,j-1/2,k+1/2} \tag{3.77}$$

where

$$\overset{*}{w}_{i,j+1/2,k+1/2} = \begin{cases} w_{i,j,k+1/2}, & \overline{v} > 0, \\ w_{i,j+1,k+1/2}, & \overline{v} \leqslant 0 \end{cases}, \quad \overline{v}_{i,j+1/2,k+1/2} = \frac{1}{2}\left(v_{i,j+1/2,k} + v_{i,j+1/2,k+1}\right)$$

The vertical convection term is discretized using the following scheme

$$\frac{\partial\sqrt{g^*}w^2}{\partial\sigma} \approx \sqrt{g^*}\overline{w}\overset{*}{w}\Big|_{i,j,k+1} - \sqrt{g^*}\overline{w}\overset{*}{w}\Big|_{i,j,k} \tag{3.78}$$

where

$$\overset{*}{w}_{i,j,k} = \begin{cases} w_{i,j,k-1/2}, & \overline{w} > 0, \\ w_{i,j,k+1/2}, & \overline{w} \leqslant 0 \end{cases}, \qquad \overline{w}_{i,j,k} = \frac{1}{2}\left(w_{i,j,k+1/2} + w_{i,j,k-1/2}\right)$$

In the top z-layer $k = k_{\max}^{i,j;n}$ the surface velocity $w_{i,j,k+1/2}^n = \omega_{i,j}^n$ is substituted (where $\omega_{i,j}^n$ is defined by (3.46) and similarly in the bottom layer $k = k_{\max}^{i,j;n}$ the substitution $w_{i,j,k-1/2}^n = 0$ is made.

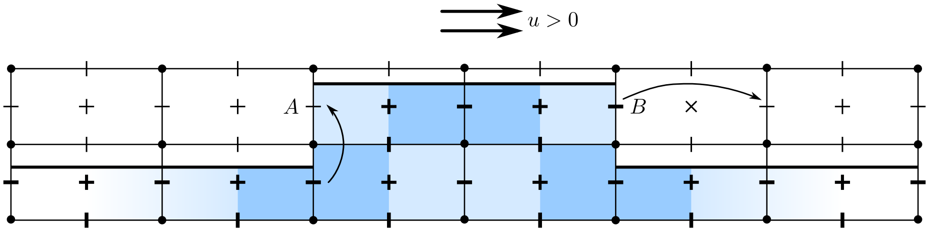


Figure 3.8: A small block wave propagating with a uniformly positive velocity illustrating the treatment at vertical free-surface sections. The momentum equations are solved for all velocity variables indicated with thick dashes.

3.10 Boundary conditions and z-layer artifacts

In this section a brief description is given of the discretization of the boundaries and peculiarities arising due to the discretization on a staggered z-layer grid.

3.10.1 Open boundaries

All open and closed boundaries are strictly vertical and in the horizontal plane they are aligned with the grid lines in ξ and η-direction. At open boundaries the horizontal velocity is assumed to be perpendicular to the boundary interface, hence $uv = 0$. Regarding the convection fluxes uv and vu open boundaries can therefore be treated as closed walls. On the other hand, in the vertical plane the in- and outflow is not assumed to be perpendicular to the interface and for the vertical velocity a Neumann condition is applied.

In the existing Delft3D-FLOW package a hydrostatic pressure assumption is made at open boundaries. Altogether the following assumptions are made for open boundaries along the ξ-axis

$$\begin{array}{llllll} \textit{velocity boundary}: & u = 0 & v = v(t) & w_\xi = 0 & \zeta_\xi = 0 & \delta p = 0 \\ \textit{water level boundary}: & u_\eta = 0 & v_\eta = 0 & w_\eta = 0 & \zeta = \zeta(t) & \delta p = 0 \end{array}$$

and for open boundaries along the η-axis

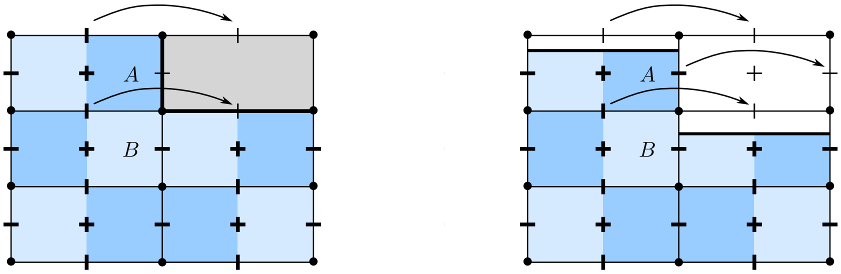
$$\begin{array}{llllll} \textit{velocity boundary}: & u = u(t) & v = 0 & w_\xi = 0 & \zeta_\xi = 0 & \delta p = 0 \\ \textit{water level boundary}: & u_\xi = 0 & v_\xi = 0 & w_\xi = 0 & \zeta = \zeta(t) & \delta p = 0 \end{array}$$

with δp denoting the non-hydrostatic pressure component. For strongly non-hydrostatic applications it might be better to apply a Neumann condition for the non-hydrostatic component of the pressure (see (1.8)).

In order to obtain a stable convection scheme the following approach is used at boundaries with missing velocity variables:

- in case of inflow the velocity gradient is unknown and the convection is assumed to be equal to zero

- in case of outflow the velocity outside the domain is obtained by means of constant extrapolation and the flux is discretized using the same scheme as in the internal of the domain



(a) Top view of velocity boundaries in the horizontal plane.

(b) Side view of a free-surface boundary.

Figure 3.9: Examples of boundary situations for $\mathcal{U}$-volumes.

3.10.2 The z-layer at the free-surface

In case of large steps in the free-surface level vertical sections can occur. In these situations it is difficult to decide for the best discretization method. Several simplifications need to be made that are inherent to the z-layer geometry.

Depending on the upwind reconstruction of the water level the velocity variables in vertical sections of the free-surface can be active or inactive. Since a first-order *upwind* reconstruction of the water level is used, velocity variables can only be active in case of outflow (e.g. location B in figure 3.8). If a velocity variable at a vertical section is active an approximation is needed for the missing velocity and pressure variable outside the water. The missing velocity is obtained by means of constant extrapolation in the horizontal direction and an approximation of the pressure gradient is obtained by means of constant extrapolation in the vertical direction by taking the highest located available pressure gradient in the same column (see figure 3.8).

The outwards pointing fluxes point into inactive cells. Therefore at vertical sections in the free-surface the discretization of the convection is not momentum-conservative.

For the discretization of the horizontal convection of u, v and w, at vertical free-surface sections the same approach is used as described above for open boundaries.

3.10.3 Corner flow

Due to grid staggering both in the horizontal and the vertical plane, corner-shaped boundaries can result in half-closed flux interfaces. This can happen around corners in the horizontal plane, e.g. figure 3.9(a) and around corners at the free-surface, e.g. figure 3.9(b). In these cases only one velocity variable is avaiable for the approximation of the transport velocity at the flux interface. In order to use the same scheme as in the internal of the domain the missing variable is obtained by means of constant extrapolation, illustrated with arrows in figures 3.9(a) and 3.9(b).

For vertical flux interfaces at the free-surface constant extrapolation is used for obtaining the missing velocity variable (volume A in figure 3.9(b)). However for horizontal flux interfaces at the free-surface the continuity equation is used to obtain the velocity at the free-surface using the procedure as described in section 3.4.2.

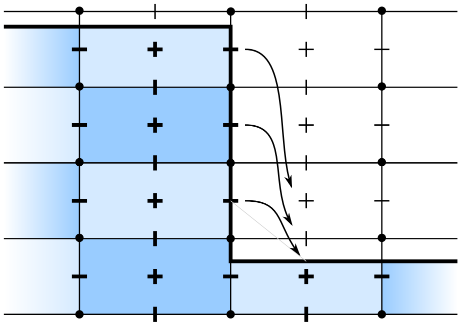


Figure 3.10: Mass flowing out of the vertical free-surface section is added to the neighbouring mass volume at the free-surface using an explicit bookkeeping procedure.

3.10.4 Conservation of mass

Difficulties arising at vertical sections in the free-surface form an important drawback of using a z-layer grid for simulations with strong surface dynamics. Near vertical tiles of water bookkeeping needs to be used in order to obtain conservation of mass. Mass fluxes at vertical sections are modeled as falling water as shown in figure 3.10. At the end of the time step the mass fluxes along the vertical tile are added up to the grid cell in the top layer. Although being mass conservative at the discrete level, this redistribution obviously does not give an accurate approximation of the physical conservation equations and unlike the discretization of the rest of the mass fluxes the bookkeeping procedure is not implicit and therefore negatively influences stability of the model. Using an implicit discretization of these mass fluxes is also not an option, because this would destroy the banded structure of the pressure correction matrix.

3.10.5 Stability in the vertical direction

In the horizontal layers at the bottom and free-surface the height of the volumes can become very small. Since the discretization of the momentum prediction equations contains explicit terms this can lead to violation of the CFL condition. For obtaining a stable time integration scheme a lower bound is set on the height of the mass- and momentum volumes. For this purpose the numerical parameter `dzmin` is introduced, denoting the lower bound of the volume

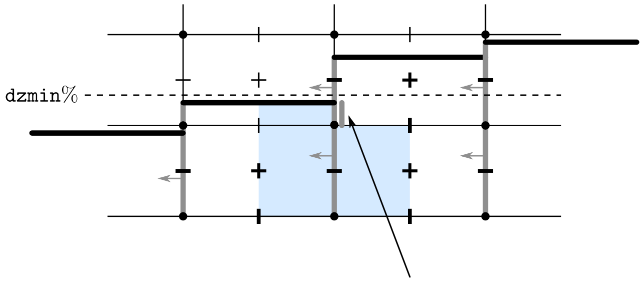


Figure 3.11: A situation in which the height of the mass volume is larger than the layer thickness but stays below the `dzmin` threshold

height expressed as percentage of the fixed layer thickness Δz_k. It is difficult to determine an appropriate percentage. The value should not be taken too small in order to avoid too severe time step restrictions. On the other hand it should not be taken too large because this would introduce inconsistencies in the discretization of the mass- and momentum fluxes at the bottom and free-surface, as illustrated in figure 3.11.

During the numerical experiments it was found that a lower bound of 5% leads to a stable method without significant loss of accuracy. In comparison with the existing finite-difference scheme, which includes the vertical convection and viscosity implicitly, a stronger stability restriction is posed on the time step. This forms one of the major drawbacks of the currently presented finite-volume method. For accurate modeling of surface dynamics it is desirable to use a smaller value for `dzmin`. This could be accomplished by implementing an unconditionally stable scheme for the vertical convection and viscosity terms, for example a θ-scheme.

Chapter 4

Simulations

In this chapter are presented several 2DV wave-simulations to validate the proposed finite-volume model. Also a comparison is made with the existing finite-difference implementation of the momentum equations.

> *Note: Since this chapter only concerns 2DV simulations on a Cartesian grid for convenience the coordinate transformation is omitted.*

In the comparison it is important to keep in mind all differences between the finite-volume method introduced in chapter 3 and the existing finite-difference method as implemented in Delft3D-FLOW.

(i) momentum conservation properties: the introduced finite-volume implementation is momentum conservative whereas the original finite-difference implementation is not

(ii) inherent to difference (i) is the different treatment of the boundaries at the free-surface and bottom

(iii) vertical convection: the original model adopts a central implicit discretization scheme for vertical convection which at the boundaries reduces to first-order upwind

In view of the current work our primary interest is in the differences (i) and (ii). However, research after the effect of these differences is complicated by the treatment of vertical convection in the original Delft3D-FLOW model. As mentioned before, the central scheme with implicit Euler discretization in time was originally implemented within the context of a hydrostatic shallow-water z-layer model. For hydrostatic flow situation the central scheme remains stable because vertical velocities are generally small and the oscillatory behaviour of the spatial discretization is compensated for by the implicit Euler discretization in time. A short analysis shows that the truncation error of the implicit central scheme is of the form

$$\Delta t w \frac{\partial^2 w}{\partial z \partial t} + O(\Delta z^2) \tag{4.1}$$

Since hydrostatic flow is dominated by horizontal flow the Courant number with respect to the vertical velocity is generally large (that is $\gg 1$) and the amount of damping due to the implicit treatment accordingly.

Since the introduction of the non-hydrostatic version of Delft3D-FLOW by Bijvelds in 2001 this part of the scheme has remained unchanged. In typical non-hydrostatic applications vertical flow has the same order of magnitude as horizontal flow. For these situations the existing scheme becomes less stable. On one hand because the vertical convection terms become more important, hence the oscillatory behaviour of the central scheme more prominent, on the other hand because the numerical damping due to implicit treatment becomes relatively small since the Courant number with respect to vertical flow is generally much smaller. This is confirmed in the numerical results obtained for simulations of short large-amplitude waves.

Difference (iii) is eliminated by replacing the central scheme for vertical convection with a first-order upwind scheme making the convection discretization first-order upwind in all three directions. Regarding the discretization in time the implicit Euler scheme is replaced by an explicit Euler scheme, because the first-order upwind discretization in space already provides sufficient numerical dissipation. A further analysis of the existing finite difference model as well as a full description of the adaptations made, can be found in appendix B.

4.1 Standing wave in a basin

A typical test case for code validation is the simulation of a standing wave. Already in [16] simulations were performed for a linear wave and a short wave with intermediate amplitude. With respect to the newly developed finite-volume method we are primarily interested in the simulation of waves with strongly nonlinear behaviour. Therefore the simulations in [16] are only used here for code validation and a brief comparison is made in case of linear and weakly nonlinear behaviour.

A wave of length λ and amplitude A is introduced in a basin with average depth H and length L equal to the wave length, i.e. $L = \lambda$ (compare e.g. [5],[16]).

An approximate wave solution is imposed by setting

$$\zeta(x, t=0) = A\cos(k(x - x_0)) \qquad\qquad u(x, t=0) = 0 \tag{4.2}$$

Because the initial velocity is set to zero non-linear effects are not taken into account and the solution is only by approximation a standing wave. After startup higher harmonics are introduced. To get an idea of the relative importance of the different terms in the Navier-Stokes equations a dimensionless form can be derived based on a Boussinesq approach for the dimensionless quantities. In chapter 2 it was already seen for the shallow-water equations that the importance of the convection terms is proportional to the relative wave amplitude A/H (cf. equation 2.6). Now we also have a look at the importance of the terms in the vertical momentum equation (here we only present a dimensionless form including the convection terms and the pressure gradient[1]):

$$\frac{\partial u}{\partial t} + \frac{A}{H}\left(\frac{\partial u^2}{\partial x} + \frac{\partial vu}{\partial y} + \frac{\partial wu}{\partial z}\right) + \frac{\partial \zeta}{\partial x} + (kH)^2\frac{\partial \delta p}{\partial x} = RHS \tag{4.3a}$$

$$\frac{\partial v}{\partial t} + \frac{A}{H}\left(\frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} + \frac{\partial wv}{\partial z}\right) + \frac{\partial \zeta}{\partial y} + (kH)^2\frac{\partial \delta p}{\partial y} = RHS \tag{4.3b}$$

$$\frac{\partial w}{\partial t} + \frac{A}{H}\left(\frac{\partial uw}{\partial x} + \frac{\partial vw}{\partial y} + \frac{\partial w^2}{\partial z}\right) + \frac{\partial \delta p}{\partial z} = RHS \tag{4.3c}$$

[1] I thank Mart Borsboom for providing me with useful information on the subject of non-dimensionalization in the context of non-hydrostatic flow.

The dimensionless quantities vary from experiment to experiment and are specified later on. From the dimensionless form (4.3) it is observed that the size of the convection of vertical momentum is proportional to the product of the relative amplitude A/H and the squared relative wave length kH. Considering the quantity kH leads to the notion of relative wave length. Waves can be classified as long waves ($kH \ll 1$), waves of intermediate length ($kH \approx 1$) and short waves ($kH \gg 1$). A rough division between different flow regimes can be made by considering the quantity A/H as a measure of nonlinearity and considering the quantity kH as a measure of the importance of the non-hydrostatic pressure component.

Short waves ($kH \gg 1$) are dominated by vertical flow whereas long waves ($kH \ll 1$) are dominated by horizontal flow, and in general the wave starts to show more nonlinear behaviour as the relative amplitude is increased.

The wave celerity follows from the linear dispersion relation (see e.g. [8]).

$$c = \pm\sqrt{\frac{g}{k}\tanh kH} \tag{4.4}$$

For long waves it follows that $c \approx \pm\sqrt{gH}$, for waves of intermediate length the full dispersion relation needs to be used and for short waves the phase speed is approximated by $c \approx \pm\sqrt{g/k}$.

It must be mentioned that Delft3D-FLOW is not primarily designed for wave simulations and the distinction just made does not capture all aspects of flow simulation.

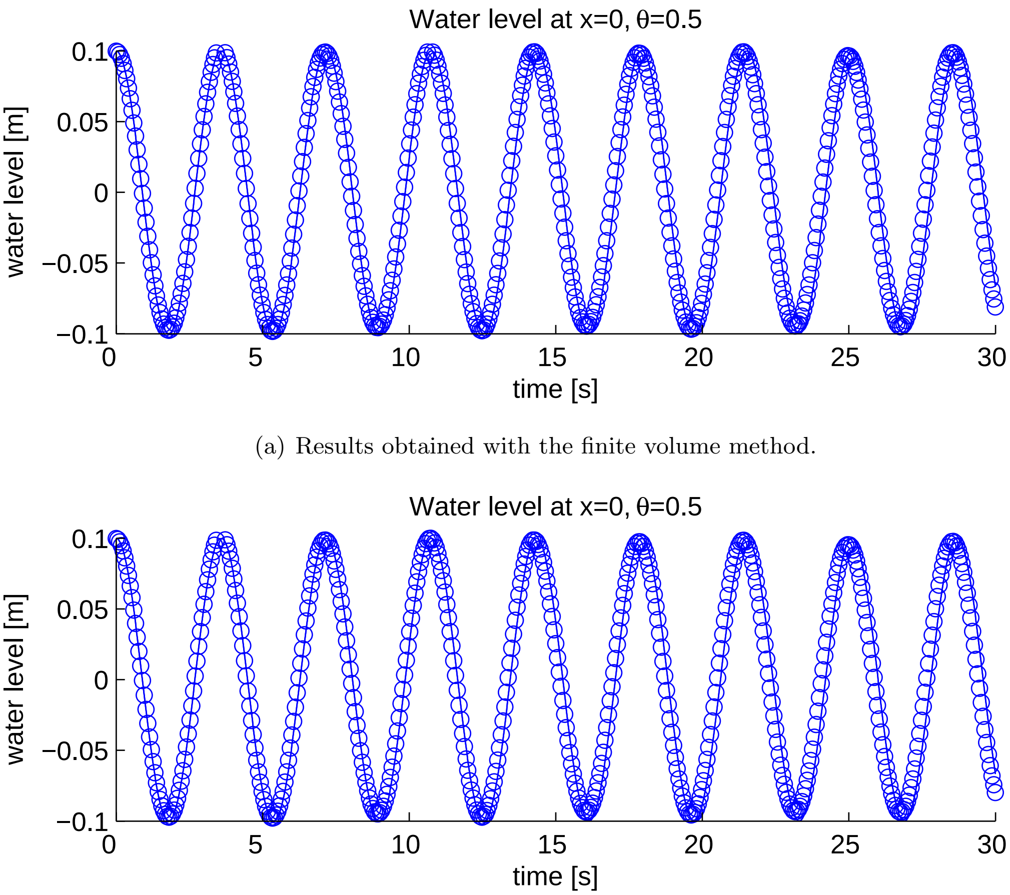
The present work focuses on the improvement of the convection discretization for flow situations of strong or intermediate nonlinearity, e.g. a wave solution characterized by large relative amplitudes $A/H \gg 0$. The discretization of the convection terms should give an improvement in all flow situations, however the simulations in this chapter primarily center around flow situations characterized by $kH \approx 1$ and $kH \gg 1$.

4.1.1 A small-amplitude wave solution

First a simulation has been done of a wave with length $\lambda = 20.0\,\mathrm{m}$ and amplitude $A = 0.1\,\mathrm{m}$ in a basin of intermediate depth $H = 10.0\,\mathrm{m}$. Since $k = 2\pi/(20\,\mathrm{m}) \approx 0.314\,\mathrm{m}^{-1}$ the characteristic dimensionless quantities are given by $kH \approx 0.31$ and $A/H \approx 0.01$, showing that the wave solution behaves almost linearly. The linear dispersion relation (4.4) shows that $\omega \approx 1.75\,\mathrm{s}^{-1}$. Following [16] pp. 44-50, the experiment is set up in Delft3D-FLOW on a grid of 11 equally spaced z-layers. The vertical grid spacing is chosen so that $\zeta = 0$ is exactly located at the center of the top z-layer. In this way the wave solution remains within a single horizontal layer and inaccuracies due to crossing of horizontal layers is avoided. (The effect of layer-crossings is investigated for large-amplitude waves later on in this chapter.) In the horizontal direction a grid spacing of $\Delta x = 0.5\,\mathrm{m}$ is used. Since an explicit scheme is used for the convection the Courant number must be kept below unity. For this reason the time step is set to $\Delta t = 0.012\,\mathrm{s}$, resulting in a wave Courant number of $(\omega/k)\cdot \Delta t/\Delta x \approx 0.134$.

To avoid damping of the θ-scheme, the time-integration parameters are set to $\theta_p = \theta_\zeta = 0.5$ (see [16], p. 46 for an illustration of the damping properties of the θ-scheme).

Since the wave has a small amplitude and is of intermediate length the solution is expected to behave linearly and not demonstrate any numerical damping. In view of the observations made in chapter 2, for linear waves the differences between the finite-volume discretization and the existing finite-difference implementation are expected to be minimal. By setting the viscosity to zero and removing the bottom friction the only source of numerical damping consists of the convection schemes. Since the convection terms are negligibly small the results



(a) Results obtained with the finite volume method.

(b) Results obtained with the finite-difference method.

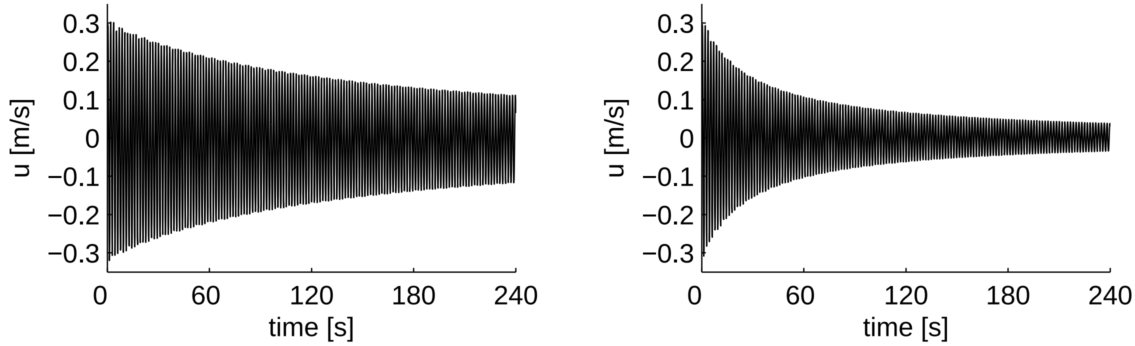
Figure 4.1: Checkpoint analysis of the water level at $x = 0\,\mathrm{m}$ for a linear wave solution with parameters $A/H = 0.01$, $kH \approx 0.31$. In both cases the method is seen to approximate the predicted wave solution very well.

are expected to be almost absent of numerical damping. This is confirmed by the numerical results presented in figures 4.1(a) and 4.1(b).

4.1.2 A short wave with intermediate amplitude

The wave length is changed to $\lambda = 5\,\mathrm{m}$ and for comparison with the results presented in [16], pp. 51-54, the wave is discretized on a grid of 20 grid cells in the horizontal direction leading to a grid spacing of $\Delta x = 0.25\,\mathrm{m}$. The reference depth is kept at $H = 10\,\mathrm{m}$ and in the vertical direction the domain is divided in 11 equally spaced horizontal layers as done in the previous experiment. The dimensionless quantities are now given by the relative amplitude $A/H \approx 0.01$ and relative wave length $kH \approx 12.6$. Looking at the dimensionless form of the Navier-Stokes equations (4.3) it is seen that the convection terms in the vertical momentum equation become of larger importance. However, the overall importance of the convection terms is still fairly small.

The purpose of the current experiment is a comparison of the spatial discretization of



(a) Results obtained with the finite-volume method.

(b) Results obtained with the existing finite-difference method.

Figure 4.2: The horizontal velocity in the top layer at $x = 1.25\,\mathrm{m}$ for a wave with parameters $A/H = 0.01$, $kH \approx 12.6$.

the momentum equations. Numerical damping due to the θ-scheme is avoided by setting $\theta_p = \theta_\zeta = 0.5$. Furthermore, by taking a small time step the effect of time discretization errors is reduced making it possible to compare the numerical schemes for spatial accuracy. By taking a small time step in particular the numerical damping due to the implicit vertical convection scheme in the existing finite-difference model is reduced. By setting the time step to $\Delta t = 0.012\,\mathrm{s}$ the effect of numerical dissipation due to the implicit Euler discretization of the vertical convection is much smaller than the numerical damping due to the upwind spatial discretization. For the current test case the results obtained with both versions of the existing finite-difference method show no significant differences since the spatial discretization of the vertical convection in the surface layer in both cases comes down to a first-order upwind scheme and due to the small time step the difference between explicit and implicit treatment is negligible.

The results obtained with the finite-volume scheme are less influenced by numerical damping than the results of the existing finite-difference implementation as can be seen from figures 4.2(a) and 4.2(b).

Although the existing finite-difference discretization is based on a central scheme for the vertical convection, near boundaries it reduces to a first-order upwind scheme (see equation (B.1) in appendix B). Furthermore, in case of flow pointing from outside the domain the convection is assumed to be equal to zero. At open boundaries where no information is available this seems to be an appropriate assumption. However, at the free-surface the continuity equation provides additional information about the velocity normal to the free-surface that can be used to improve the discretization of the convection in case of inwards pointing flow. When using a finite-difference approach it is more complicated to use this information since the normal velocity is generally not coordinate aligned, in particular for flow with strong surface dynamics. In a finite-volume setting, on the other hand, the normal velocity can be used directly to complete the discretization of the convection flux at the free-surface.

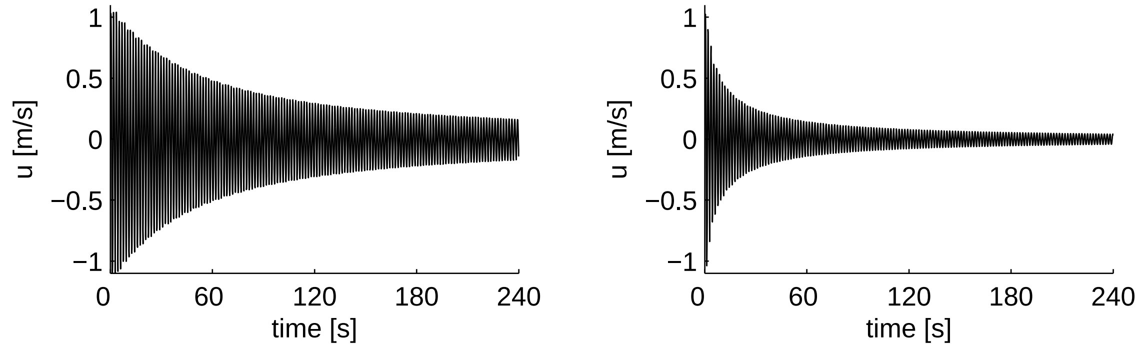


Figure 4.3: Horizontal velocity in the top layer at $x = 1.25\,\mathrm{m}$ for the fully first order upwind adaptation of the existing finite difference method (left) and the finite volume method (right). The amplitude of the initial cosine profile is changed to $A = 0.4\,\mathrm{m}$.

4.1.3 Large amplitude waves

In the test case just presented the numerical damping is still relatively small. After thirty wave periods the amplitude is reduced with approximately one third with the finite volume approach and with two thirds with the finite-difference approach. If the wave amplitude is increased, the amount of numerical damping grows rapidly as is illustrated in figure 4.3 where the damping behaviour for a wave with amplitude $A = 0.4$ is shown.

For short waves with large amplitudes the differences between the numerical schemes should become even more prominent and therefore provide a useful instrument for comparison. However a useful comparison can only be made at the beginning of the simulation. because the numerical schemes used for simulation are all first-order accurate and introduce artificial wave damping. Therefore solutions which are initially characterized by large values of A/H are quickly characterized by much smaller values of A/H. Higher harmonics which are introduced due to nonlinear effects are even more sensitive to numerical damping and hardly become visible in the simulations. It is clear that for accurate modeling of nonlinear effects the numerical damping needs to be reduced and that a higher-order convection discretization is needed.

Three different large-amplitude waves have been considered with relative amplitudes of $A/H = 0.15$ and larger. The water depth is kept constant at $H = 10\,\mathrm{m}$ and waves are simulated with a length of $\lambda = 5, 50, 500\,\mathrm{m}$ leading to the dimensionless quantities $kH \approx 12.6, 1.26, 0.126$ respectively. In order to sufficiently capture details of the wave, the standing wave solutions are discretized on a grid of 40 grid cells in the horizontal direction, leading to $\Delta x = 0.125, 1.25, 12.5\,\mathrm{m}$ respectively. In vertical direction the domain reaches from $z = -10.5\,\mathrm{m}$ to $z = 5.5\,\mathrm{m}$ and the solution is discretized on a grid of 32 horizontal layers with uniform vertical spacing $\Delta z = 0.5\,\mathrm{m}$. For the short wave the number of horizontal layers is increased to 64 in order to restrict the grid-cell ratio $\Delta z/\Delta x$. At the bottom a coarser grid could be used, but in order to exclude effects due to non-uniform grid spacing this is not done in the current simulations.

The central scheme for vertical convection as used in the existing Delft3D-FLOW model introduces spurious oscillations which posed a problem for the simulation of short large-amplitude waves. This becomes more and more visible when the quantities A/H and kH are increased. Eventually the finite-difference model becomes unstable whereas the finite-volume method, which adopts a first-order upwind discretization for all directions, remains stable. For further details we refer to the last part of appendix B.

> *Note: In view of the remarks above, all comparison was continued with the fully three-dimensional upwind adaptation of the finite difference model.*

During the numerical experiments it was found that for short waves with large amplitudes the θ-scheme with paramater $\theta = 0.5$ makes both models unstable. The instability may be due to the fact that the momentum prediction equations are still based on an explicit discretization of convection and horizontal viscosity. Furthermore it must be noted that the mass-fluxes in the continuity equation are based on a frozen grid-geometry and at the free surface the θ-scheme only applies to the velocity variables: with respect to ζ the time-integration is only partly implicit. This problem is inherent to the discretization in terms of water level and velocity variables. Using a discretization in terms of mass-flux variables would solve this problem (see appendix A).

In some cases the time-integration scheme can be made stable again by choosing θ slightly above 0.5. In this way the introduced instabilities are suppressed by introducing a small amount of numerical damping. For some cases, in particular for simulations of short waves with large amplitudes, this was not sufficient. For waves of intermediate length this did not pose a problem, even when introducing waves with very large amplitudes. For steady state calculations using a larger value for θ does not pose a problem, in fact it can even speed up convergence towards the steady state. However, in order to further explore the applicability of the model to dynamic flow situations, which can be characterized by large kH and large A/H, the time integration needs further consideration.

Chapter 5

Conclusions and recommendations

This thesis describes a three-dimensional model to simulate non-hydrostatic free-surface flows. The model was originally introduced by Bijvelds [4] and several improvements have been made since then, of which the latest in 2008 by Ullmann [16]. In the current work a new discretization has been presented for the convection and pressure terms in the momentum prediction equations.

The existing model is based on a non-conservative finite difference discretization of the convection terms. In the horizontal direction a first order upwind scheme is used which was thought to be the main source of numerical damping its origin in the horizontal convection term. Following the observations made by Ullmann [16], the primary objective of the current research was to introduce an improved discretization for the horizontal convection terms. Additional research brought forward other sources of numerical damping and other properties of the original scheme that pose a problem for accurate modeling of convection-dominated non-hydrostatic flow problems:

- The vertical convection is based on a implicit Euler scheme with central discretization in space. This discretization forms heritage the original hydrostatic version of Delft3D-FLOW and is less convenient for non-hydrostatic flow. For non-hydrostatic flow simulations in which the vertical velocity becomes more important the central scheme tends to introduce numerical wiggles.

- At the bottom and free-surface the original vertical convection scheme reduces to first-order upwind or vanishes, depending on the flow direction. This leads to an inaccurate treatment of convection at the free surface.

- A short study pointed out other sources of numerical damping, consisting of the first-order upwind reconstruction of the water levels and the upwind discretization of the vertical convection near boundaries.

For strongly non-hydrostatic applications the velocity components are often of similar size and the grid spacing accordingly. Therefore, as also suggested in the conclusions of [16], there is no reason to treat horizontal and vertical convection differently. Because numerical experiments have shown that the existing discretization of the vertical convection terms is not suitable for strongly non-hydrostatic applications it was chosen to replace it with an explicit first order upwind scheme as already used for horizontal convection.

A model study showed the importance of adopting a momentum-conservative discretization for the momentum equations. It also demonstrated that for strongly time-varying flow it is important to use a momentum-conservative time-integration. Another source of inaccuracy of the existing model was formed by the inaccurate discretization around the bottom and free-surface. Using a finite-volume method leads to a natural discretization of the kinematic boundary conditions at the bottom and free surface which in the existing non-hydrostatic version of Delft3D-FLOW were poorly implemented. With these objectives in mind a finite-volume discretization for the momentum prediction equations has been presented having the following properties:

- An explicit first-order upwind scheme is used in all three directions, avoiding numerical damping due to implicit treatment of vertical convection and unstable behaviour due to the central discretization of these same terms. Of course the numerical damping due to the spatial discretization of the convection is still present.

- The momentum equations are discretized in terms of water level and velocity variables using a frozen grid-geometry and this fits in easily into the existing Delft3D-FLOW framework.

- A convection discretization is obtained which is momentum-conservative on rectangular grids. On curvilinear grids, the equations in terms of contravariant velocities are not conservative.

- Within the possibilities given by the z-layer architecture, the reconstruction of the bottom levels is made second-order accurate.

- The grid-surface fluxes lead to a more accurate discretization of convection at the free-surface. On the other hand it is based on an explicit discretization of the continuity equation and for the horizontal momentum equations contains constant extrapolation for the velocity at the free surface.

Simulations have been carried out for validation of the new model and for comparison of its performance with the existing as well as adapted finite-difference model. Simulations of standing waves showed that the finite-volume method gives results that are similar to those obtained with the finite-difference method. Both methods still suffer from numerical damping and the differences are mainly observed directly after startup of the simulations. It was observed that the finite-volume method shows visibly less numerical damping than the existing finite difference method.

At the free-surface the implementation of a finite-volume method is more complicated than that of a finite-difference method, due to its stronger dependency on the description of the geometry as well as the z-layer administration at the programming level. For the finite-volume method the variable thickness of the free-surface grid layer leads to a stronger stability restriction on the time step. With standard settings the height of surface cells can become as small as five percent of the fixed layer thickness, making an explicit discretization of convection and viscosity less attractive, in particular in the vertical direction. This suggests the need of an implicit time discretization. Preferably the discretization of the convection is made unconditionally stable (e.g. by using a θ-scheme) to avoid time step restrictions due to the variable sized free-surface grid layer. At the same time an unconditionally stable time-integration scheme allows for smaller values of the lower bound on the layer thickness `dzmin`, reducing inconsistencies in the flux-discretization.

During the simulations of short large-amplitude waves the time-integration scheme of flow becomes unstable unless $\theta_p \gg 0.5$ and $\theta_\zeta \gg 0.5$ are used. Although a θ-scheme is used for the continuity equation the model still contains several parts that are discretized explicit in time. Apart from vertical viscosity the discretization of the momentum prediction equations is fully explicit. Also the discretization of the continuity equation itself is partly explicit, because the mass-fluxes are based on the geometry of the solution at the old time level. The explicit parts of the discretization lead to a reduction of stability. Generally it suffices to use a value for θ slightly above 0.5. In the case of short large-amplitude waves the value had to be taken much larger, however taking $\theta = 1$ in all cases sufficed. For dynamic applications this solution is however not acceptable, because in this way too much artificial damping is introduced.

It was observed that instabilities frequently find their origin in the free-surface layer, most frequently around vertical sections. In those areas several approximations are needed to complete the discretization scheme and these are all obtained with some type of extrapolation procedure. This affects the stability of solutions with strong surface dynamics.

The main differences with the previous finite-difference model discretization and the finite-volume discretization presented in this thesis are related to momentum-conservation properties and treatment of the free-surface boundary. Further research is necessary to fully assess the impact of conservation properties on non-hydrostatic flow simulations. The new three-dimensional first-order upwind scheme still suffers from numerical damping. This give rise to the following possible improvements:

- In order to make the model suitable for simulations with large vertical length scales the convection discretization should be made higher-order accurate for example by adopting a higher-order flux-limiter scheme. Preferably the scheme is based on a compact stencil since flow simulations are frequently discretized on small grids in which case a in a large part of the domain a limited number of neighbouring velocity variables is active.

- For accurate simulation of flow with strong surface dynamics (large vertical length scales) the first-order upwind reconstruction of the free-surface should be replaced by a higher-order reconstruction where some form of limiting is needed to ensure monotonicity. The damping effect of the first order water level reconstruction is small compared with the damping due to the convection discretization and therefore, removing it has a lower priority.

Based on our experiences during the numerical simulations it can be concluded that the existing Delft3D-FLOW framework is, without additional changes, not suitable for strongly convection-dominated flow. The source of this problem lies in the explicit discretizations and extrapolation procedures occurring in special situations at the free-surface. This comprises all major aspects of the flow discretization: convection, mass-fluxes as well as the pressure gradient. A part of these drawbacks can be eliminated by using a locally boundary-fitted grid in terms of mass-flux variables. This leads to the following primary recommendation for further research.

Recommendation for further research

Implementation of a locally boundary-fitted finite-volume discretization for the momentum equations in Delft3D-FLOW has several clear advantages over using a finite-difference method or a finite-volume method based on a frozen grid-geometry (for more details we refer to appendix A):

- A model study in chapter 2 showed that the momentum-conservative convection scheme in h, q-formulation is more accurate than other basic schemes in h, u-formulation. It is expected that this observation generalizes to the context of the Navier-Stokes equations.
- The pressure terms are implemented using a finite-volume method, making the discretization of the pressure at the free-surface fully conservative.
- At the free-surface the kinematic condition and atmospheric pressure can be substituted avoiding the need of discretizing grid-surface fluxes. In this way extrapolation of the velocity towards the free-surface is avoided and the method becomes more stable.
- By using a discretization in terms of mass-flux variables, a genuinely implicit time-integration scheme is obtained for the continuity equation.
- The time-integration scheme of the momentum equation is automatically conservative since it is expressed in momentum variables.
- Obtaining a fully $O(\Delta t^2)$ time integration with respect to the pressure and water levels is fairly straightforward and is not inconsistent with the upwind reconstruction of the water levels as would be the case when linearizing the existing finite-difference or finite-volume method.

Changing to a locally boundary-fitted grid asks for some thorough adaptations of the existing model, but is not expected to be more complicated than using a discretization in terms of water level and velocity variables.

Bibliography

[1] *Delft3D-FLOW, User Manual v. 3.14.,rev. 6375.* Delft, 2009.

[2] *WAQUA/TRIWAQ, two- and -three-dimensional shallow-water flow model. Technical Documentation*, 2009.

[3] A. Arakawa and V.R. Lamb. Computational design of the basic dynamical processes of the ucla general circulation model. *Methods in Computational Physics*, 17:173–265, 1977.

[4] M. Bijvelds. *Numerical Modelling of Estuarine Flow over Steep Topography, PhD thesis.* Delft Univ. of Techn., 2001.

[5] M. Bijvelds. A non-hydrostatic module for delft3d-flow. Technical report, Deltares, Delft, 2003.

[6] M.J.A. Borsboom. Discretization of physical domain and equations. Technical report, Deltares, Delft, 2007.

[7] J. Gaarthuis. A non-hydrostatic pressure model for shallow water flow. Technical report, Technische Universiteit Eindhoven, 1995.

[8] L.H. Holthuijsen. *Waves in Oceanic and Coastal Waters.* Cambr. Univ. Press, 2007.

[9] C. B. Laney. *Computational Gasdynamics.* Cambr. Univ. Press, 1998.

[10] R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambr. Univ. Press, 2002.

[11] G.S. Stelling and M.M. Busnelli. Numerical simulation of the vertical structure of discontinuous flow. *J. Numer. Meth. Fluids*, 37:23–43, 2001.

[12] G.S. Stelling and S.P.A. Duinmeijer. A staggered conservative scheme for every froude number in rapidly varied shallow water flows. *Int. J. Numer. Meth. Fluids*, 43:1329–1354, 2003.

[13] G.S. Stelling and Laguzzi M.M. Kernkamp, H.W.J. Delft flooding system: a powerful tool for inundation assessment based upon a positive flow simulation. *Hydroinformatics '98: proceeedings of the third International Conference on Hydroinformatics*, pages 449–456, 1998.

[14] G.S. Stelling and J.A.Th.M. Van Kester. A non-hydrostatic flow model in cartesian coordinates. Technical report, Delft Hydraulics, 1995.

[15] G.S. Stelling and M. Zijlema. An accurate and efficient finite-difference algorithm for non-hydrostatic free-surface flow with application to wave propagation. *J. Numer. Meth. Fluids*, 43:1–23, 2003.

[16] S. Ullmann. *Three-dimensional computation of non-hydrostatic free-surface flows, MSc thesis.* Delft Univ. of Techn., 2008.

[17] R.W.C.P. Verstappen and A.E.P. Veldman. Symmetry-preserving discretization of turbulent flow. *J. Comput. Phys.*, 187:343–368, 2003.

[18] C.B. Vreugdenhil. *Numerical methods for shallow-water flow.* Kluwer, 1994.

[19] P. Wesseling. *Principles of Computational Fluid Dynamics.* Springer, 2001.

[20] P. Wesseling, A. Segal, and C.G.M. Kassels. Computing flows on general three-dimensional nonsmooth staggered grids. *J. Comput. Phys.*, 149:333–362, 1999.

[21] Stelling G.S. Willemse, J.B.T.M. and G.K. Verboom. Solving the shallow water equations with an orthogonal coordinate transformation. *J. Numer. Meth. Fluids*, 37:23–43, 2001.

Appendix A

Discussion: a locally boundary-fitted momentum volumes

Based on out experiences it seems to be better to use a locally boundary-fitted grid. In this way the discretization around the free-surface could be made more stable. The kinematic condition and atmospheric pressure can be substituted avoiding the need of discretizing a flux at the free surface.

In this appendix we have a short look at the discretization of the momentum equations on a boundary fitted grid. For discretization it is more convenient to describe the equations in terms of mass-flux variables describing the total mass fluxes integrated over the interfaces of the mass volumes:

$$q^n = \sqrt{g_{\eta\eta}} z_\sigma u \tag{A.1a}$$

$$r^n = \sqrt{g_{\xi\xi}} z_\sigma v \tag{A.1b}$$

$$s^n = \sqrt{g^*} w \tag{A.1c}$$

For brevity only the equations in the surface layer are considered, the discretization for internal and bottom grid cells is similar. First the discretized *momentum prediction equations*, based on (3.15), are solved

$$\sqrt{g_{\xi\xi}} \frac{q^{n+1*} - q^n}{\Delta t} = RHS \tag{A.2}$$

$$\sqrt{g_{\eta\eta}} \frac{r^{n+1*} - r^n}{\Delta t} = RHS \tag{A.3}$$

$$z_\sigma^n \frac{s^{n+1*} - s^n}{\Delta t} = RHS \tag{A.4}$$

In this form the discretized momentum equations are integrated over the entire moving water layer and take into account the change in time of the water level. The next step consists of solving the *momentum correction equations*

$$\sqrt{g_{\xi\xi}}\frac{q^{n+1}-q^{n+1*}}{\Delta t} = -\theta_p\left[\sqrt{g_{\eta\eta}}z^n_\sigma\Delta p\right]^{i+1,j,k}_{i,j,k} + \theta_p\left[\sqrt{g_{\eta\eta}}z^n_\xi\Delta p\right]^{i+1/2,j,k+1/2}_{i+1/2,j,k-1/2} \tag{A.5a}$$

$$\sqrt{g_{\eta\eta}}\frac{r^{n+1}-r^{n+1*}}{\Delta t} = -\theta_p\left[\sqrt{g_{\xi\xi}}z^n_\sigma\Delta p\right]^{i,j+1,k}_{i,j,k} + \theta_p\left[\sqrt{g_{\xi\xi}}z^n_\eta\Delta p\right]^{i,j+1/2,k+1/2}_{i,j+1/2,k-1/2} \tag{A.5b}$$

$$z^n_\sigma\frac{s^{n+1}-s^{n+1*}}{\Delta t} = -\theta_p\left[\sqrt{g^*}\Delta p\right]^{i,j,k+1}_{i,j,k} \tag{A.5c}$$

The pressure terms are implemented using a θ-scheme. Note that in order to avoid implicit coupling the vertical grid spacing in the discretization of the pressure gradient is kept at the old time level. This is done similarly in the discretization of the time derivative of the vertical momentum correction equation.

The pressure-correction variables Δp are calculated in such a way that the velocities at the new time level satisfy the continuity equation. This is done by substituting the momentum correction equations in the continuity equation. For the discretization in time of the continuity equation at the free-surface a θ-scheme is used (denoted with θ_ζ). As is done similarly in section 3.3 a discretization of the continuity equation in terms of mass-flux variables can be derived:

$$D_\xi\left[q^{n+1}\right] + D_\eta\left[r^{n+1}\right] + D_\sigma\left[s^{n+1}\right] = 0, \qquad k^{i,j}_{\text{min}} < k < k^{i,j;n}_{\text{max}} \tag{A.6a}$$

$$D_\xi\left[q^{n+1}\right] + D_\eta\left[r^{n+1}\right] - s^{n+1}_{i,j,k+1/2} = 0, \qquad k^{i,j}_{\text{min}} = k < k^{i,j;n}_{\text{max}} \tag{A.6b}$$

$$-\sqrt{g^*}_{i,j}\frac{\zeta^{n+1}_{i,j}-\zeta^n_{i,j}}{\Delta t} = D_\xi\left[q^{n+\theta_\zeta}\right] + D_\eta\left[r^{n+\theta_\zeta}\right], \qquad k^{i,j}_{\text{min}} = k = k^{i,j;n}_{\text{max}} \tag{A.6c}$$

$$-\sqrt{g^*}_{i,j}\frac{\zeta^{n+1}_{i,j}-\zeta^n_{i,j}}{\Delta t} = D_\xi\left[q^{n+\theta_\zeta}\right] + D_\eta\left[r^{n+\theta_\zeta}\right] + s^{n+\theta_\zeta}_{i,j,k-1/2}, \qquad k^{i,j}_{\text{min}} < k = k^{i,j;n}_{\text{max}} \tag{A.6d}$$

where we used the mass-flux difference operators (3.36), furthermore we define

$$q^{n+\theta_\zeta} = \theta_\zeta q^{n+1} + (1-\theta_\zeta)q^n, \tag{A.7a}$$

$$r^{n+\theta_\zeta} = \theta_\zeta r^{n+1} + (1-\theta_\zeta)r^n, \tag{A.7b}$$

$$s^{n+\theta_\zeta} = \theta_\zeta s^{n+1} + (1-\theta_\zeta)s^n \tag{A.7c}$$

An important difference is that the discretization of the mass fluxes is no longer based on the geometry of the solution at the old time level.

Substitution of the momentum correction equations (A.5) in the continuity equation (A.6) we obtain a preliminary form of the pressure-correction equations for surface cells

$$\begin{aligned}-\sqrt{g^*}\frac{\zeta^{n+1}-\zeta^n}{\Delta t} &+ \Delta t\theta_\zeta\theta_p\left(D_{\xi\xi}\left[\Delta p\right] + D_{\eta\eta}\left[\Delta p\right] + D_\sigma\left[\Delta p\right]\right)\\ &= \theta_\zeta\left(D_\xi\left[q^{n+1*}\right] + D_\eta\left[r^{n+1*}\right] + s^{n+1*}_{i,j,k-1/2}\right)\\ &+ (1-\theta_\zeta)\left(D_\xi\left[q^n\right] + D_\eta\left[r^n\right] + s^n_{i,j,k-1/2}\right),\\ &\qquad k^{i,j}_{\text{min}} < k = k^{i,j;n}_{\text{max}}\end{aligned} \tag{A.8}$$

introducing the second-difference operators

$$D_{\xi\xi}\left[\Delta p\right] = \frac{\left[\sqrt{g_{\eta\eta}}\Delta z^n \Delta p\right]_{i,j}^{i+1,j}}{\sqrt{g_{\xi\xi}}_{i+1/2,j}} - \frac{\left[\sqrt{g_{\eta\eta}}\Delta z^n \Delta p\right]_{i-1,j}^{i,j}}{\sqrt{g_{\xi\xi}}_{i-1/2,j}} \tag{A.9a}$$

$$D_{\eta\eta}\left[\Delta p\right] = \frac{\left[\sqrt{g_{\xi\xi}}\Delta z^n \Delta p\right]_{i,j}^{i,j+1}}{\sqrt{g_{\eta\eta}}_{i,j+1/2}} - \frac{\left[\sqrt{g_{\xi\xi}}\Delta z^n \Delta p\right]_{i,j-1}^{i,j}}{\sqrt{g_{\eta\eta}}_{i,j-1/2}} \tag{A.9b}$$

$$D_{\sigma}\left[\Delta p\right] = -\frac{\left[\sqrt{g^*}\Delta p\right]_{i,j,k-1}^{i,j,k}}{\Delta z^n_{i,j,k-1/2}} \tag{A.9c}$$

Here we have simplified the second-difference operators in horizontal direction by using the fact that at the free-surface the atmospheric pressure can be substituted and that internal interfaces are strictly horizontal, i.e. $z_\xi = z_\eta = 0$.

The differences with the existing implementation are twofold. The pressure-correction terms are now discretized using a finite-volume method. Secondly, instead of the predicted velocities $u^{n+1*}, v^{n+1*}, w^{n+1*}$ now the predicted mass fluxes q^{n+1*},r^{n+1*},s^{n+1*} are substituted. This is more in line with the conservative formulation used here. In the free-surface layer the pressure-correction equation still contains the unknown water levels at the new time level. These can be eliminated by using a linearization of the dynamic pressure condition at the free-surface of the new time level, $p^{n+1}\big|_{\zeta^{n+1}} = p_0$, which yields an expression for $\zeta^{n+1} - \zeta^n$ in terms of pressure-correction variables (for details about the derivation consult [16][1])

$$\zeta^{n+1} - \zeta^n = \frac{\alpha^n_{i,j}}{g}\left(\Delta p_{i,j,k-1} + \delta p^n_{i,j,k-1}\right) + \frac{1-\alpha^n_{i,j}}{g}\left(\Delta p_{i,j,k} + \delta p^n_{i,j,k}\right) \tag{A.10}$$

where δp denotes the non-hydrostatic pressure component. After substitution of (A.10), we obtain the *pressure-correction equations for surface cells*:

$$\begin{aligned} -\sqrt{g^*}\frac{\alpha^n_{i,j}\Delta p_{i,j,k-1} - (1-\alpha^n_{i,j})\Delta p_{i,j,k}}{g\Delta t} &+ \Delta t \theta_\zeta \theta_p \left(D_{\xi\xi}\left[\Delta p\right] + D_{\eta\eta}\left[\Delta p\right] + D_\sigma\left[\Delta p\right]\right) \\ &= \theta_\zeta\left(D_\xi\left[q^{n+1*}\right] + D_\eta\left[r^{n+1*}\right] + s^{n+1*}_{i,j,k-1/2}\right) \\ + (1-\theta_\zeta)\left(D_\xi\left[q^n\right] + D_\eta\left[r^n\right] + s^n_{i,j,k-1/2}\right) &+ \sqrt{g^*}\frac{\alpha^n_{i,j}\delta p^n_{i,j,k-1} + (1-\alpha^n_{i,j})\delta p^n_{i,j,k}}{g\Delta t}, \\ & k^{i,j}_{\min} < k = k^{i,j;n}_{\max} \end{aligned} \tag{A.11}$$

After solving the continuity equation the pressure-correction values Δp can be substituted in the momentum correction equations to calculate the mass fluxes $q^{n+1}, r^{n+1}, s^{n+1}$.

[1] Within the current finite-volume context the same derivation can be used, however taking into account that the location of the pressure variables in the surface layer is no longer fixed in time. This results in a slightly modified parameter (compare [16], p. 20)

$$\alpha^n_{i,j} = \frac{z^n_{i,j,k} - \zeta^n_{i,j}}{\Delta z^n_{i,j,k-1/2}}$$

Remark The time integration is not $O(\Delta t^2)$ yet because the discretization of the pressure gradients in the momentum correction equations is based on a frozen grid geometry. This could be improved by incorporating the grid geometry in the pressure-correction term

$$\Delta[\Delta z p] = (\zeta^{n+1} - z_{i,j,k-1/2})p^{n+1} - (\zeta^n - z_{i,j,k-1/2})p^n$$

Implicit coupling can be avoided by linearizing the pressure-correction term, leading to

$$\sqrt{g_{\xi\xi}}\frac{q^{n+1} - q^{n+1*}}{\Delta t} = -\theta_p \left[\sqrt{g_{\eta\eta}}\Delta z^n \Delta p + p^n \sqrt{g_{\eta\eta}}(\zeta^{n+1} - \zeta^n)\right]_{i,j,k}^{i+1,j,k} \tag{A.12a}$$

$$\sqrt{g_{\eta\eta}}\frac{r^{n+1} - r^{n+1*}}{\Delta t} = -\theta_p \left[\sqrt{g_{\xi\xi}}\Delta z^n \Delta p + p^n \sqrt{g_{\eta\eta}}(\zeta^{n+1} - \zeta^n)\right]_{i,j,k}^{i,j+1,k} \tag{A.12b}$$

Linearizing the pressure fluxes in time is similar to linearizing the height of the mass-flux interfaces in the continuity equation as given by (3.37), followed by substitution of the momentum prediction equations in their original finite-difference form (as proposed in [6]). However in the latter case one would need to linearize the water levels at the non-integer indices, which introduces inconsistencies with the upwind reconstruction of the water levels at the new time level.

Appendix B

Further study on the existing finite-difference scheme

B.1 Sources of numerical damping

The discussion in this section centers around the test case concerning a short wave of intermediate amplitude, presented in chapter 4, section 4.1.2.

A short analysis performed in [16] suggested that the numerical damping in figure 4.2(b) is mainly caused by the first-order upwind discretization of the horizontal convection terms. However it might be that there are other significant sources of damping related to the discretization in the grid layer at the free-surface.

Apart from the convection scheme for horizontal convection, the accuracy of the approximation in the free-surface layer can also be influenced by

- the discretization of vertical convection for which a backwards Euler scheme is used
- the first-order upwind reconstruction of the free-surface in the velocity points (equation (3.57))

Furthermore it must be noted that the second-order central scheme for the vertical convection at the boundaries reduces to a first-order upwind scheme. In the test case under consideration the wave propagates entirely in the top layer of the grid and the discretization of the convection terms outside this layer is of secondary importance, hence for this specific wave solution the discretization of vertical convection is effectively first-order upwind and this is expected to be a significant source of damping.

In order to further investigate which effects are causing the damping observed in figure 4.2(b) we have carried out several simulations with adapted versions of the original finite-difference scheme.

Vertical convection It has been found that also the vertical convection term plays a role. In the existing finite-difference implementation of the momentum equations an implicit Euler scheme is used for the vertical convection terms. The discretization in space consists of a second-order central scheme which near the boundaries reduces to a first-order upwind

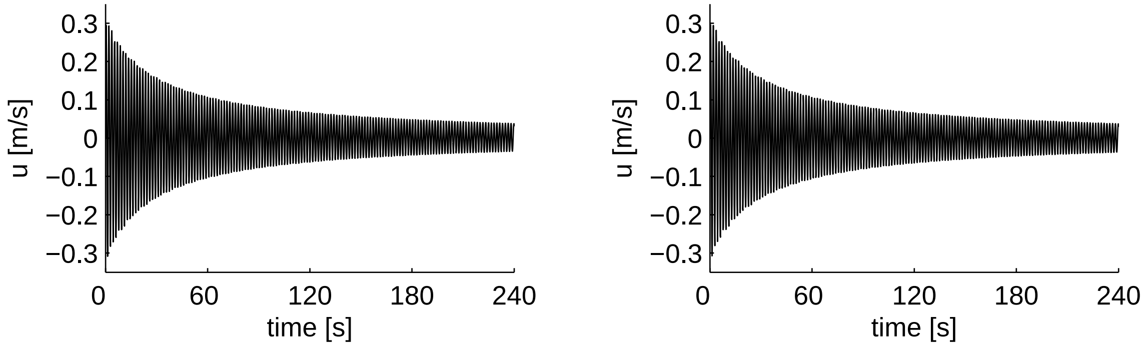


Figure B.1: Illustration of the damping behaviour of the original finite difference model (left) and the adapted, fully first-order upwind finite-difference method. The figures describe the horizontal velocity in the top layer at $x = 1.25\,\mathrm{m}$. As expected the damping behaviour of both schemes is similar because a small time step is used and the spatial discretization in the surface layer is identical.

scheme. For the convection of vertical momentum the scheme is given by

$$\max(w^n_{i,j,k+1/2}, 0)\cdot \frac{w^{n+1*}_{i,j,k+1/2} - w^{n+1*}_{i,j,k-1/2}}{\Delta z_k} \qquad k = k^{i,j;n}_{\max} - 1 \tag{B.1a}$$

$$w^n_{i,j,k+1/2}\cdot \frac{w^{n+1*}_{i,j,k+3/2} - w^{n+1*}_{i,j,k-1/2}}{\Delta z_k + \Delta z_{k+1}}, \qquad k^{i,j}_{\min} < k < k^{i,j;n}_{\max} - 1 \tag{B.1b}$$

$$\min(w^n_{i,j,k+1/2}, 0)\cdot \frac{w^{n+1*}_{i,j,k+3/2} - w^{n+1*}_{i,j,k+1/2}}{\Delta z_{k+1}} \qquad k = k^{i,j}_{\min} \tag{B.1c}$$

and for the convection of horizontal momentum it is given by (vertical convection of v is done similarly)

$$\max(\overline{w}^n_{i+1/2,j,k}, 0)\cdot \frac{u^{n+1*}_{i+1/2,j,k} - u^{n+1*}_{i+1/2,j,k-1}}{\Delta z_{k-1/2}} \qquad k = k^{i,j;n}_{\max} \tag{B.1d}$$

$$\overline{w}^n_{i+1/2,j,k}\cdot \frac{u^{n+1*}_{i+1/2,j,k+1} - u^{n+1*}_{i+1/2,j,k-1}}{\Delta z_{k-1/2} + \Delta z_{k+1/2}}, \qquad k^{i,j}_{\min} < k < k^{i,j;n}_{\max} \tag{B.1e}$$

$$\min(\overline{w}^n_{i+1/2,j,k}, 0)\cdot \frac{u^{n+1*}_{i+1/2,j,k+1} - u^{n+1*}_{i+1/2,j,k}}{\Delta z_{k+1/2}} \qquad k = k^{i,j}_{\min} \tag{B.1f}$$

where $\overline{w}^n$ is obtained by arithmetic averaging of the neighbouring vertical velocity variables. The implicit Euler discretization is thought to be an important source of numerical damping. To confirm this idea the existing scheme for vertical convection in the finite-difference method is replaced by an explicit scheme. Switching to an explicit treatment of the vertical convection an alternative needs to be found for the central scheme which in explicit form is unstable. For this reason it is replaced by a first-order upwind scheme as already used for horizontal convection. So for internal grid layers the spatial discretization of scheme (B.1) is changed to

$$\max(w^n_{i,j,k+1/2},0)\cdot\frac{w^n_{i,j,k+1/2}-w^n_{i,j,k-1/2}}{\Delta z_k}+\min(w^n_{i,j,k+1/2},0)\cdot\frac{w^n_{i,j,k+3/2}-w^n_{i,j,k+1/2}}{\Delta z_{k+1}} \tag{B.2a}$$

and

$$\max(\overline{w}^n_{i+1/2,j,k},0)\cdot\frac{u^n_{i+1/2,j,k}-u^n_{i+1/2,j,k-1}}{\Delta z_{k-1/2}}+\min(\overline{w}^n_{i+1/2,j,k},0)\cdot\frac{u^n_{i+1/2,j,k+1}-u^n_{i+1/2,j,k}}{\Delta z_{k+1/2}} \tag{B.2b}$$

After this adaptation the discretization of the convection is first-order upwind in all directions and apart from conservation properties becomes comparable to the scheme presented in chapter 3. The expectations are twofold:

- For large time steps the adapted model should perform better since it does not suffer from numerical damping in time as the original model does.
- For small time steps the differences should become small since the spatial discretization of the convection terms in both models is identical.

Figure B.1 shows a comparison of the results obtained with the original model and the fully first-order upwind adaptation of the existing finite-difference method. As expected the results are very similar because a small time step is used and the spatial discretization used in the free-surface layer is identical.

Free-surface reconstruction Now let us continue with the fully first-order upwind adaptation of the original finite-difference scheme. Another source of numerical damping is expected to be the reconstruction of the water levels at the velocity points. The first-order approximation of the water levels leads to significant numerical damping in the mass fluxes of the continuity equation in the grid layer at the free-surface. The effect of this on the solution is shortly investigated by replacing the first-order upwind reconstruction (3.57) with the following second-order central approximation

$$\zeta_{i+1/2,j}=\frac{1}{2}\left(\zeta_{i,j}+\zeta_{i+1,j}\right) \tag{B.3}$$

For the fully first-order upwind adaptation of the existing finite-difference approach, using a central reconstruction of the free-surface level results in further reduction of the numerical damping, as shown in figure B.2. The model is still stable and this could be explained by the smooth solution under consideration, but also by the larger numerical damping of the finite-difference convection scheme which compensates for the unstable property of the central scheme. For the finite-volume method, which already shows somewhat less numerical damping than the finite-difference implementation, the use of a central free-surface reconstruction results in an unstable discretization scheme. In the latter case a limiter approach could be used to obtain higher order accuracy while ensuring monotonicity. The results of this simulation indicate the benefits that can be obtained from using a higher order approximation for the water levels at the velocity points, but on the other hand clearly demonstrate the importance of using a monotone (limiter) approach.

In this section two sources of numerical damping have been identified and after (preliminary) adaptations the only remaining source of first-order numerical damping is the upwind discretization of the convection terms.

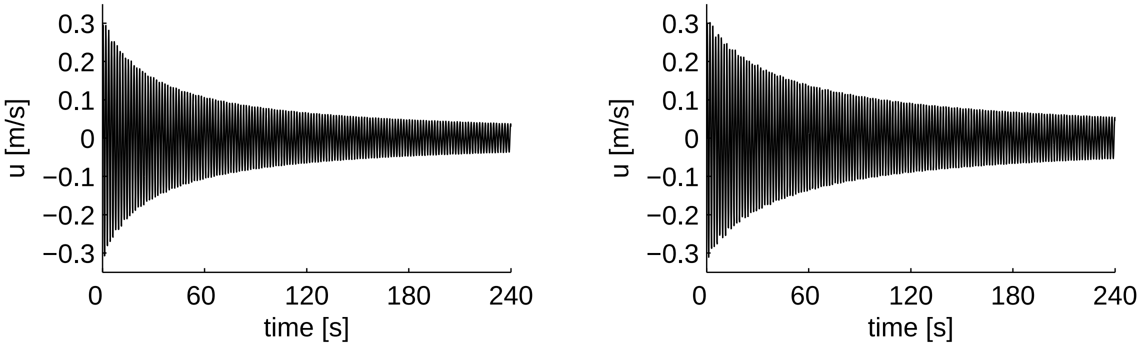


Figure B.2: Horizontal velocity in the top layer at $x = 1.25\,\mathrm{m}$ for the fully first-order upwind adaptation of the existing finite-difference method with a first-order upwind reconstruction of the free-surface (left) and a second-order central reconstruction of the free-surface (right). The second-order discretization of the free-surface results in an additional reduction of the numerical damping.

B.2 Instability of the implicit central scheme

During the simulation of large amplitude waves it was found that the finite-volume method is more stable than the finite-difference method. Increasing the amplitude even further, the finite-difference method is not able to produce stable results anymore, whereas the finite-volume method remains stable. The source of the unstable behaviour of the finite-difference scheme is thought to be the central discretization of the vertical convection. Since the finite-volume method also uses an upwind scheme also in the vertical direction it introduces stabilizing dissipation. This idea is confirmed by taking a higher coefficient for the vertical viscosity. After increasing the amount of vertical viscosity the finite-difference scheme start to produce similar results as the finite-volume scheme. For the smooth wave solutions under consideration the differences between the two methods are minimal.