5 | Numerical framework

5.1 Introduction

An important characteristic of many coastal structures is that they are constructed of porous media, i.e. coarse granular material, rock or concrete units in variable sizes. An effective treatment of the flow in porous media is one of the main requirements in the numerical simulation of wave interaction with a RMB.

A second specific feature which characterizes the problem is the presence of a free surface. Different techniques for tracking and locating the free surface in NS solvers have been developed, as discussed in Chapter 2.

In the following chapter, the numerical framework is presented in which the present research has been carried out. The model equations and their numerical implementation are discussed, providing a background for the numerical study. In this discussion, the main features of the considered wave-structure interaction are emphasized: porous media flow and free-surface modeling.

5.2 Motivation

From a scientific point of view, it is obviously preferred to have full access to the source code of the CFD model. There are a number of open source CFD codes available, e.g. OpenFOAM®[1] or TRUCHAS[2]. In general however, these models are not specifically developed for solving the fluid flow in wave-structure interaction, but encompass a wide variety of physical features such as electromagnetism, phase change, solid mechanics or heat transfer. It was not until very recently that an open source model specifically designed for coastal engineering applications has been released, called IHFOAM (Higuera et al., 2013). The model is based on OpenFOAM® and is still under

[1] www.openfoam.org
[2] telluride.lanl.gov

development. At present stage, it does not include porous media flow.

Even when disposing of a generic open source model, developing and validating a code requires a considerable amount of time and financial resources. Considering the aforementioned, it was decided to employ a readily available, state-of-the-art commercial CFD model. There are a number of widely used commercial codes available, so the question rises which one is most suitable to address the research goals formulated in Chapter 1. The answer to that question is not straightforward, and probably a number of arguments can be used in favor of one or another specific model. After screening a number of available codes, the model FLOW-3D® was selected, developed by Flow Science Inc.[3]. This choice is mainly based on the following arguments:

- FLOW-3D makes use of the VOF technique, developed by dr. C.W. Hirt and founder of Flow Science. The rigorous and efficient implementation of the VOF method is considered to be one of the main strengths of this particular model, providing an accurate and robust method for free-surface tracking. In this respect, the numerical algorithm to evolve the shape and location of the free surface whilst maintaining its character as a discontinuity and the application of proper free-surface boundary conditions are mentioned (Flow Science, Inc., 2012);
- the availability of customizable subroutines, providing some flexibility to the user to implement or adjust features of the model;
- the capability to model moving objects, which will be of importance to represent a piston wavemaker (cfr. infra);
- an efficient approach for grid generation and obstacle representation based on a cut-cell method, greatly reducing the amount of work on the users's part.

FLOW-3D is a multi-physics solver with options for a broad wide range of flow problems. The physical background of the model is presented hereafter, together with a description of the numerical implementation. The discussion is concentrated on the particular problem of wave interaction with a permeable structure.

5.3 Model equations

The following section presents the physical background of the numerical model. The basic equations are the fundamental equations for fluid

[3] www.flow3d.com

dynamics, which reflect the conservation of mass and momentum. They are presented here first in general form in a clear fluid region, i.e. in absence of obstacles of any kind. Next, the extension of the conservation equations with models for porous media flow and turbulence is presented. In addition to the conservation equations, the treatment of a free surface or fluid interface, in respectively single-fluid or two-fluid problems, forms one of the cornerstones of the model and deserves particular attention. Finally, a brief discussion is given on the initial and boundary conditions completing the specific problem setup.

5.3.1 Conservation equations

Mass cannot be created nor destroyed. The rate of accumulation of mass inside an infinitely small control volume has to be balanced by the net outflow of mass through its boundaries:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \tag{5.1}$$

where ρ is the fluid density and $u_i (i = x, y, z)$ the Cartesian components of the velocity. Note the use of the Einstein summation convention. For an incompressible fluid, eq. (5.1) reduces to

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{5.2}$$

The conservation of momentum is based on Newton's second law, stating that the change in momentum in a control volume is due to the forces that act on that volume. The gravity force (acceleration g) is assumed to be the only acting body force. In differential form, the equation for an incompressible fluid reads:

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = g\delta_{ij} - \frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial x_j} \tag{5.3}$$

where the terms on the left-hand-side of eq. (5.3) express the local and convective acceleration, equalized by the gravity force $g\delta_{ij}$, pressure forces (p) and viscous accelerations (τ). The momentum equations (5.3), often referred to as the Navier-Stokes equations, describe the fluid motion with full consideration of nonlinear effects.

Eq. (5.3) is generally employed assuming that the shear stress τ_{ij} is proportional to the velocity gradient. A fluid with such properties is called Newtonian. When the fluid is incompressible, the shear stresses read:

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{5.4}$$

with μ the dynamic viscosity.

5.3.2 Porous media flow model

In FLOW-3D, the porous medium flow resistance is modeled by inclusion of a drag term in the momentum equations, as discussed in section 2.3.2. Several drag models are available in the code to represent saturated or unsaturated porous media flow, including effects of capillary pressure. Coarse granular material is used in most coastal engineering applications, in which case the Forchheimer model is suitable (referred to as 'Reynolds number dependent drag' in FLOW-3D). Using this model, a drag term $F_d u_i$ is added to the right-hand-side of eq. (5.3), obtained by combination of eq. (2.3) and eq. (2.30):

$$F_d u_i = -g(anu_i + bn^2|u_i|u_i) \tag{5.5}$$

where $u_i = u_i^s$ are the velocity components solved in the momentum equation (5.3), representing the seepage velocity in the porous medium. $|u_i|$ is the norm of the (seepage) velocity vector and n the porosity. It is noticed that the inertial drag term in unsteady flow is neglected in eq. (5.5). The dimensional coefficients a and b were discussed in section 2.1.5. In FLOW-3D, the following formulation is used:

$$a = \alpha_F \frac{(1-n)^2}{n^3} \frac{\nu}{gD^2} \ , \ b = \beta_F \frac{(1-n)}{n^3} \frac{1}{gD} \tag{5.6}$$

where D is a characteristic grain size diameter (e.g. D_{50} or $D_{n,50}$) and α_F and β_F dimensionless shape factors.

It is noticed that the effect of macroscopic turbulence (see section 2.3.2) inside porous media is not considered in FLOW-3D. To the author's knowledge, no validation studies for coastal engineering applications have been reported that clearly identify the contribution of this term in the total flow loss inside porous media. It may be expected that macroscopic gradients of seepage velocity are intrinsically small, due to the averaging approach, and so will be the related turbulent flow losses. An exception may be found near the interfaces between clear fluid and porous medium or interfaces between porous media with different characteristics, where larger velocity gradients arise.

It is noticed that the presence of a porous medium does not only affect the equation for momentum conservation, but also the mass conservation equation and kinematic free-surface boundary condition. The specific numerical approach in FLOW-3D to satisfy these conditions will be treated in section 5.4.

5.3.3 Turbulence modeling

The most accurate approach for turbulence modeling, referred to as Direct Numerical Simulation (DNS), is to solve the Navier-Stokes equations with

proper initial and boundary conditions, resolving the whole range of spatial and temporal scales of the turbulence, from the smallest dissipative scales up to the integral scale associated with the motions containing most of the kinetic energy. The computational cost of DNS is extremely high, even at low Reynolds numbers. Besides these constraints involving computational time and cost, it is even not desirable to apply DNS in cases where porous flow modeling is applied because of the volume-averaging approach.

When DNS is out of scope, turbulent effects are to be approximated in the model. Mainly two options are available: Large-Eddy Simulation (LES) and Reynolds-averaging of the Navier Stokes equations, which are both available in FLOW-3D. With LES, the basic idea is to directly compute all turbulent flow structures that can be resolved by the computational grid and approximate only those features that are too small to be resolved, using a subgrid-scale model. Since a considerable amount of kinetic energy in the flow must be resolved, a high grid resolution is still needed. To allow the large flow structures to break up into smaller ones, the flow has to be simulated in three dimensions and time-accuracy has to be retained. Moreover, an energy-conserving discretization of the momentum convection is mostly needed in LES, in order not to dampen out resolved, turbulent fluctuations (Rauwoens, 2008). Because of these imperatives, and since the numerical simulations will be restricted to 2D in the following, LES is not explored further in this work.

The last approach for turbulence modeling treats the flow from a statistical point of view, restricting the description of turbulence to a subset of statistical properties, e.g. the mean value and (co-)variance of velocity components. The Reynolds-decomposition technique leads to the so-called RANS equations, where an instantaneous flow variable q is separated into a mean (ensemble-averaged) component $\overline{q}$ and a fluctuating component q'. In RANS, the breaking of large turbulent structures into smaller ones is not simulated, hence a 3D grid is not strictly necessary. In general, the requirements to the discretization are also less stringent, because of the relatively large importance of the turbulent model terms. The RANS equations for an incompressible fluid read:

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0 \tag{5.7a}$$

$$\frac{\partial \overline{u}_i}{\partial t} + \overline{u}_j \frac{\partial \overline{u}_i}{\partial x_j} = g\delta_{iz} - \frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_i} + \frac{1}{\rho}\frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right) - \rho\overline{u_i' u_j'}\right] \tag{5.7b}$$

The shear stresses in eq. (5.7b) arise from momentum transfer at molecular level (viscous contribution) and from the fluctuating velocity field (turbulent contribution). The turbulent shear stresses $-\rho\overline{u_i' u_j'}$, often referred to as the Reynolds shear stresses, require additional modeling to close the RANS

equation for solving. This has led to the creation of a number of turbulence models. In FLOW-3D, two types of two-equation turbulence transport models are incorporated: the standard k-ϵ and the ReNormalization Group (RNG) model. Both are highlighted briefly further on. It is of importance to mention that the application of the turbulence models is limited to the clear-fluid region, thus not within porous media.

The k-ϵ and RNG turbulence models, like many others, are based on the turbulent-viscosity hypothesis of Boussinesq (1877). This hypothesis, analogous to the stress-rate-of-strain relation of a Newtonian fluid, states that the Reynolds shear stresses are directly proportional to the mean rate of fluid deformation:

$$-\rho\overline{u_i'u_j'} = \mu_T\left(\frac{\partial\overline{u}_i}{\partial x_j}+\frac{\partial\overline{u}_j}{\partial x_i}\right)-\frac{2}{3}\rho k\delta_{ij} \tag{5.8}$$

where μ_T=$\mu_T(x_i,t)$ is the dynamic eddy or turbulent viscosity which varies in space but is assumed to be isotropic however. k is the turbulent kinetic energy, defined as:

$$k=\frac{1}{2}\overline{u_i'u_i'} \tag{5.9}$$

Substitution of eq. (5.8) in eq. (5.7b) yields:

$$\frac{\partial\overline{u}_i}{\partial t}+\overline{u}_j\frac{\partial\overline{u}_i}{\partial x_j}=g\delta_{iz}-\frac{1}{\rho}\frac{\partial}{\partial x_i}\left(\overline{p}+\frac{2}{3}\rho k\right)+\frac{\partial}{\partial x_j}\left[\nu_{eff}\left(\frac{\partial\overline{u}_i}{\partial x_j}+\frac{\partial\overline{u}_j}{\partial x_i}\right)\right] \tag{5.10}$$

where

$$\nu_{eff}(x_i,t)=\nu+\nu_T(x_i,t) \tag{5.11}$$

is the effective kinematic viscosity, i.e. the sum of molecular (ν) and turbulent kinematic viscosity (ν_T). Equation (5.10) has the same appearance as the incompressible Navier-Stokes equation (5.3), with $\overline{u}_i$ and ν_{eff} in place of u_i and ν and with $\overline{p}+\frac{2}{3}\rho k$ as the modified pressure term.

Standard $k-\epsilon$ model

The standard k-ϵ model (Harlow and Nakayama, 1967; Launder and Spalding, 1974) consists of two transport equations, for the turbulent kinetic energy k and dissipation rate ϵ, respectively:

$$\frac{\partial k}{\partial t}+\overline{u_j}\frac{\partial k}{\partial x_j}=\nu_T\left(\frac{\partial\overline{u}_i}{\partial x_j}+\frac{\partial\overline{u}_j}{\partial x_i}\right)\frac{\partial\overline{u}_i}{\partial x_j}+\frac{\partial}{\partial x_j}\left[\left(\nu+\frac{\nu_T}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right]-\epsilon \tag{5.12a}$$

$$\frac{\partial\epsilon}{\partial t}+\overline{u}_j\frac{\partial\epsilon}{\partial x_j}=C_{\epsilon1}\frac{\epsilon}{k}\nu_T\left(\frac{\partial\overline{u}_i}{\partial x_j}+\frac{\partial\overline{u}_j}{\partial x_i}\right)\frac{\partial\overline{u}_i}{\partial x_j}+\frac{\partial}{\partial x_j}\left[\left(\nu+\frac{\nu_T}{\sigma_\epsilon}\right)\frac{\partial\epsilon}{\partial x_j}\right]-C_{\epsilon2}\frac{\epsilon^2}{k} \tag{5.12b}$$

where the terms are from left to right: local derivative, convective derivative, production, diffusion and dissipation. A dimensional analysis yields a definition of the turbulent viscosity, necessary to link the momentum equation (5.10) with the transport equations (5.12) :

$$\nu_T = C_\mu k^2/\epsilon \tag{5.13}$$

Standard values of the model constants in the $k-\epsilon$ turbulence model equations are:

$$C_\mu = 0.09, C_{\epsilon 1} = 1.44, C_{\epsilon 2} = 1.92, \sigma_k = 1.0, \sigma_\epsilon = 1.3 \tag{5.14}$$

RNG $k-\epsilon$ model

The RNG method, similar to the k-ϵ model, accounts for the effects of smaller scales of motion by applying a renormalization technique to the Navier-Stokes equations, see e.g. Yakhot et al. (1992). This method results in a modified form of the ϵ equation, attempting to account for different scales of motion through changes in the production term. This modification makes the RNG model more sensitive to flows having strong shear regions, due to the presence of the source term R:

$$\frac{\partial \epsilon}{\partial t} + \overline{u}_j \frac{\partial \epsilon}{\partial x_j} = C_{\epsilon 1} \frac{\epsilon}{k} \nu_T \left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i} \right) \frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_T}{\sigma_\epsilon} \right) \frac{\partial \epsilon}{\partial x_j} \right] - C_{\epsilon 2} \frac{\epsilon^2}{k} - R \tag{5.15a}$$

$$R = \frac{C_\mu \eta^3 (1 - \eta/\eta_0)}{1 + \beta \eta^3} \frac{\epsilon^2}{k} \tag{5.15b}$$

$$\eta = \frac{k}{\epsilon} \sqrt{2 \overline{S}_{ij} \overline{S}_{ij}} \tag{5.15c}$$

where $\overline{S}_{ij}$ is the mean-rate-of-strain tensor, defined for incompressible flow as:

$$\overline{S}_{ij} = \frac{1}{2} \left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i} \right) \tag{5.16}$$

All constants (except β) appearing in eqs. (5.15) are derived explicitly in the RNG procedure. It is noticed that the source term according to eq. (5.15b) is an *ad hoc* model, not derived explicitly from RNG theory (Pope, 2000). Standard values of the model constants in the RNG model equations are:

$$\begin{aligned} C_\mu = 0.0845, C_{\epsilon 1} = 1.42, C_{\epsilon 2} = 1.68, \\ \sigma_k = 0.7194, \sigma_\epsilon = 0.7194, \eta_0 = 4.38, \beta = 0.012 \end{aligned} \tag{5.17}$$

Minimum dissipation rate and limits for turbulent scales

A particular numerical challenge of both the standard and RNG k-ϵ model is to limit the value of ϵ from below. In case eq. (5.12b) or (5.15a) yields values of ϵ close to zero, the turbulent kinetic energy should approach zero as well. If for numerical reasons this is not the case, eq. (5.13) yields large, unphysical values of ν_T. Therefore, a minimal value for ϵ is defined as:

$$\epsilon_{min} = C_\mu \sqrt{\frac{3}{2}} \frac{k^{3/2}}{TLEN} \tag{5.18}$$

where TLEN is a maximum turbulent length scale. In FLOW-3D, this parameter can be defined by the user. Alternatively, a value of TLEN (varying in space and time) can be computed by the program. In the latter procedure, the lower bounds of the turbulent length (L_T) and time scales (T_T) are based on the Kolmogorov scales, whereas the upper bounds are based on the rapid distortion theory (Isfahani and Brethour, 2009):

$$L_{T,min} = 70\nu^{3/4}\epsilon^{-1/4} \tag{5.19a}$$

$$L_{T,max} = \frac{0.86}{C_\mu} \frac{\sqrt{k}}{S} \tag{5.19b}$$

$$T_{T,min} = 6\sqrt{\frac{\nu}{\epsilon}} \tag{5.19c}$$

$$T_{T,max} = \frac{0.35}{C_\mu} \frac{1}{S} \tag{5.19d}$$

where ν is the molecular kinematic viscosity and S the mean strain rate magnitude computed from the second invariant of the strain tensor S_{ij}.

The length scale L_T, subject to the limits given by eqs. (5.19a, 5.19b) is then replacing TLEN in eq. (5.18). The inverse of the time scale T_T, subject to the limits given by eqs. (5.19c, 5.19d), is used in the right-hand-side of eq. (5.12b) or (5.15a), where ϵ/k appears.

5.3.4 Free-surface modeling

Different techniques for free-surface modeling applied in NS models were discussed in section 2.3.1. FLOW-3D employs the VOF method (Hirt and Nichols, 1981), in which fluid configurations are defined in terms of a VOF function $F(x, y, z, t)$. The interpretation of the F-function depends on how the fluid problem is being solved. Two options exist for the specific case of water in contact with air. In the first approach, referred to as *single-fluid modeling*, the air is not treated as a fluid but rather as a *void*, a region

without fluid mass with a uniform reference pressure assigned to it. In the void, no fluid properties are transported, which means a considerable reduction of computational effort. In this case, F represents the volume fraction occupied by the fluid. Thus, fluid exists where $F = 1$, and void regions correspond to locations where $F = 0$. Averaged over a control volume, the value of F will be within the segment [0,1].

In *two-phase modeling*, when water and air are explicitly treated as two different phases, the F-function represents the volume fraction of the incompressible phase (water), whereas the complementary region with volume fraction $1-F$ represents the compressible phase (air) that may have a constant density or a density computed from the fluid equation-of-state.

It is important to recognize that the F-function is defined a discontinuous function in order to accurately track a sharp free surface or two-fluid interface. A prerequisite to the numerical implementation of the VOF method is to advance the fluid interface in time without destroying its character as a discontinuity. Moreover, it is necessary to impose proper boundary conditions at the free surface in single-fluid modeling.

5.3.5 Initial and boundary conditions

The model equations for momentum, turbulence and the free surface are partial differential equations, approximating the physics of the processes they model for every possible manifestation of the process of interest. In order to solve a specific problem, initial and boundary conditions are required.

Initial conditions specify the flow at the initial time step at every location in the computational domain. In case of waves propagating in a wave flume, the fluid at rest in the whole computational domain is a common and trivial initial condition, imposing a hydrostatic pressure distribution and a zero velocity field. An initial reference pressure in the void or air region can be specified as well.

Boundary conditions represent the external factors acting in the specific flow problem through the boundaries of the domain. Hereafter, the boundary conditions for the momentum, turbulence and free-surface model equations are discussed.

Velocity

The velocity at the boundary will be affected by the type of boundary, which can be either solid or open. A solid boundary represents solid objects in the computational domain, such as the bottom or side wall of the flume, or any other object placed inside the domain. For a solid boundary, mainly two types of conditions apply:

1. no-slip condition : this condition states that on the solid boundary, there is no motion of the fluid relative to the solid

$$u_i = v_{s,i} \tag{5.20}$$

with v_s the velocity of the solid. This type of condition is valid for a viscous fluid ($\mu \neq 0$) and is used to model the flow in the boundary layer next to the rigid wall.

2. free-slip condition : this condition states that the tangential shear stress exerted by the solid is zero. In this case, the fluid velocity next to the solid boundary is only forced to match the normal velocity of the surface:

$$u_i n_{s,i} = v_{s,i} n_{s,i} \tag{5.21}$$

where $n_{s,i}$ are the components of the unit vector $\boldsymbol{n_s}$ normal to the solid surface, positive pointing outward.

An open boundary is a boundary through which fluid can enter or leave the domain. It is used to model the inflow of waves and/or currents or represent an outflow boundary through which waves or currents leave the domain without reflection. Multiple velocity boundaries are implemented in the code, going from a constant velocity (representing a current) to different types of linear and nonlinear wave boundary conditions. In case of a wave boundary, both the position of the free surface and velocity components at the boundary are specified according to the governing wave theory. More details on the wave boundary condition are given in section 6.2.3.

Pressure

Different pressure conditions exist, depending on the flow modeling type. In case of one-fluid flow, the pressure of the void region can be initialized to a value p_0 and stays constant during the computation.

In case of two-phase flow, where the water and air are treated as two separate fluids, an initial pressure condition can be applied to the air region. This is generally the atmospheric pressure, acting as a reference to any other pressure in the flow. This condition is usually applied to the upper closure of the computational domain.

A pressure boundary condition can also be applied at the mesh boundary. In the specific case of a numerical wave flume however, this kind of boundary is not employed.

Free surfaces and fluid interfaces

At the free surface or fluid interface, the F-function satisfies a kinematic boundary condition. Assuming the continuity of velocity, the free surface is

ensured to be a material surface that always consists of the same particles[4]. In the absence of mass sources, the kinematic condition reads:

$$\frac{\partial F}{\partial t} + \frac{\partial}{\partial x_i}\left(F u_i\right) = 0 \tag{5.22}$$

In two-phase flow, a diffusion term is added to the right-hand side of eq. (5.22) to account for the turbulent mixing of both phases:

$$\frac{\partial F}{\partial t} + \frac{\partial}{\partial x_i}\left(F u_i\right) = \frac{\partial}{\partial x_i}\left(\frac{\nu_T}{Sc}\frac{\partial F}{\partial x_i}\right) \tag{5.23}$$

where Sc is the turbulent Schmidt number, a dimensionless number used to characterize fluid flows in which there are simultaneous momentum and mass diffusion-convection processes. It physically represents the relative thickness of the hydrodynamic layer and the mass-transfer boundary layer. The diffusion term with a simple gradient transport according to eq. (5.23) is only appropriate for homogeneous flows where the size of the energy-containing eddies is smaller than the distance over which the gradient varies appreciably (Shirani et al., 2006). For flows near the interface with inhomogeneous turbulence, a more appropriate model would include both gradient and convective transport terms, see e.g. Lumley (1975).

In case of single-fluid flow, the normal and tangential stress need to be specified as dynamic boundary conditions at the free surface, guaranteeing the continuity of stress components. The normal stress arises from the prescribed void pressure p_0 and the equivalent surface tension pressure. Denoting $\boldsymbol{n}$ as the unit normal on the free surface and n_i as the projection of $\boldsymbol{n}$ on the coordinate directions x_i, the continuity of the normal stress at the free surface boundary is written as:

$$p - \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) n_i n_j = p_0 + \sigma\kappa_s \tag{5.24}$$

where σ is the fluid surface tension coefficient (in units of force per unit length) and κ_s the local free-surface curvature.

For 3D problems, two unit tangential vectors t^k (k=1,2) are needed to define the local tangent plane on the free surface. t_i^k is defined as the projection of t^k on the coordinate directions x_i. The continuity of the tangential stress across the free surface is expressed as:

$$\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) n_i t_j^k = t_j^k \frac{\partial \sigma}{\partial x_j} \tag{5.25}$$

[4]This only holds for a free surface that does not break up for instance due to wave breaking.

In case of two-phase flow modeling, the contact region between water and air appears as an interface in the domain. The interface does not require the specification of a dynamic boundary condition since the solution at the interface comes out naturally of the equations of motion.

Turbulent model boundary conditions

Boundary conditions to the turbulence transport equations (5.12) and (5.15a) are to be specified either at a free surface, an inflow/outflow or a solid boundary.

Advective fluxes of turbulent quantities into empty cells are set to zero since there is no fluid in those cells to either supply or receive the fluxed quantities. The same condition applies to an outflow condition. Mathematically, this takes the form of a null-flux condition:

$$\frac{\partial k}{\partial x_i} n_i = 0 \ , \ \frac{\partial \epsilon}{\partial x_i} n_i = 0 \tag{5.26}$$

At inflow boundaries, the turbulent kinetic energy k and dissipation rate ϵ must be specified:

$$k = k_0 \ , \ \epsilon = \epsilon_0 \tag{5.27}$$

At solid domain boundaries or internal obstacles where a no-slip condition is applied, contributions to the transport equation for k and ϵ need to be included, which arise from tangential wall shear stresses τ_w. Because the number of mesh points required to resolve all the details in the turbulent boundary layer would become prohibitively large in practical calculations, the flow is forced to match the *law of the wall* or *log-law*, which is an approximation for fully-developed, steady flow along a flat boundary:

$$u_{\parallel} = u^{\star} \left[\frac{1}{\kappa} ln \left(\frac{u^{\star} d}{\nu} \right) + 5 \right] \tag{5.28}$$

where $u_{\parallel}$ is the velocity component parallel to wall, κ is the von Karman constant and d the normal distance from the wall to the location where the velocity $u_{\parallel}$ is computed. The local *shear* or *friction* velocity $u^{\star}$ is defined from the wall shear stress τ_w as:

$$u^{\star} = \sqrt{\frac{\tau_w}{\rho}} \tag{5.29}$$

For eq. (5.28) to be valid, the point where $u_{\parallel}$ is calculated should fall within the turbulent log-law region. This is equivalent with the following restraint to the *viscous length scale* y^+ (Versteeg and Malalasekera, 1995):

$$30 < y^+ = \frac{u^{\star} d}{\nu} < 500 \tag{5.30}$$

For a high-Reynolds number, fully-developed flow, the boundary values of ϵ and k can be derived under the assumption of the logarithmic velocity profile eq. (5.28) and the turbulent viscosity hypothesis eq. (5.8). In the log-law region, it can be assumed that turbulent production and dissipation are in balance, leading to the boundary conditions for k and ϵ (Pope, 2000):

$$k = \frac{(u^\star)^2}{\sqrt{C_\mu}} \ , \ \epsilon = \frac{(u^\star)^3}{\kappa d} \tag{5.31}$$

5.4 Implementation

In the previous section, the equations governing the physics of the wave interaction with a permeable coastal structure have been described. Due to the nonlinear nature of the partial differential equations, analytical solutions are not readily available. For almost every particular problem of interest, flow predictions are to be obtained by solving the equations with numerical methods.

In this numerical solution, the governing equations will be approximated by systems of algebraic equations. This requires a *discretization* of the equations, approximating continuous functions of time and space with a finite amount of information. The partial differential operators appearing in the governing equations express a variation in time and space. Due to their different character, they will be discretized in a different manner.

In the following, the numerical solution methods implemented in FLOW-3D are discussed. First, practical details are given on the mesh construction, the representation of obstacles and the allocation of flow variables. Next, the spatial discretization of the governing equations is treated, followed by a discussion on the time advancement of the discretized equations and the application of stability conditions to the time advancement.

Again, this discussion is organized with a focus on wave-structure interaction, paying special attention to the representation of porous media and free-surface tracking. Great part of the following is a synopsis based on the available information in the FLOW-3D manual (Flow Science, Inc., 2011). For details on the code not included here, reference is made to this manual.

5.4.1 Mesh generation and obstacle representation

Mesh generation involves the definition of a set of non-overlapping polygons (in 2D) or polyhedra (in 3D) which completely fill a well-defined domain in space. In FLOW-3D, the mesh is *structured*, meaning that the volume elements or *cells* are well ordered. The cells can be mapped on simple data

structures in a structured mesh, using a simple scheme to label elements and identify neigboring cells. Both *Cartesian* and *cylindrical* meshes can be defined in FLOW-3D, and cell dimensions can be uniform or non-uniform.

One of the main benefits of a structured mesh is the ease of grid generation, with a minimal amount of information to be stored. One of the main drawbacks involves the representation of complex geometries. Unlike unstructured meshes, where general hexahedral cells can be used which conform with specified geometric shapes (a *body-fitted* mesh), the rectangular elements employed in a Cartesian mesh cannot accurately define complex geometric surfaces. The latter would have to be approximated by blocking out entire cells, leading to boundaries having discrete steps. These steps introduce flow losses and produce other undesirable effects.

In order to overcome this problem, obstacles are allowed to cut through the cells (*cut-cell method*). The latter method is referred to in FLOW-3D as the Fractional Area/Volume Obstacle Representation (FAVOR) method (Hirt and Sicilian, 1985). Curved obstacles, wall boundaries or other geometric features are embedded in the mesh by defining the fractional areas (A_i) and volumes ($V_{F_{i,j,k}}$) of the cells that are open to flow. The $V_{F_{i,j,k}}$ function is defined as the ratio of the open volume to the total volume in a mesh cell, and three A_i functions are defined as the ratio of the open area to the total area, at the three cell faces in the increasing cell-index direction.

The philosophy behind the cut-cell method is that the numerical algorithms are based a limited amount of information for each cell (each flow variable), so it would be inconsistent to use an excessive amount of information to define geometry (Flow Science, Inc., 2012). The cut-cell technique retains the simplicity of rectangular elements while representing complex geometric shapes at a level consistent with the use of averaged flow quantities within each volume element. Three area fractions and one volume fraction for each cell are stored, which is relatively little information compared with body-fitted grids.

Grids and geometry are free to be defined independently of one another, with very little time or effort on the part of a user. The work of computing the intersections between a grid and an obstacle description, is fully automized. This is a main advantage over unstructured grids, where the grid generation process is not completely automatic and may require considerable user interaction to produce grids with acceptable degrees of local resolution while at the same time having a minimum of element distortion.

An important point to recognize is that approximations of fluid-dynamic quantities are restricted to the open regions of cells in this cut-cell method. This restriction introduces the factors A_i and $V_{F_{i,j,k}}$ directly into the discrete approximations. For example, the flux of a quantity from one cell to another

has the fractional area of the fluxing boundary that is open to flow as a multiplier. In general, the area and volume fractions are time independent, except when the moving obstacle model (GMO) is employed.

In spite of the aforementioned advantages of structured meshes in combination with a cut-cell method, some inconveniences exist as well. First of all, applying a local mesh refinement in a particular zone of interest is not possible in a structured mesh. This can be overcome by employing multiple meshes (*mesh blocks*) with different cell dimensions that are coupled at their boundaries (either adjacent to each other or nested in each other). However, in order to limit numerical approximation errors, restrictions exist to the maximum ratio of adjacent cell dimensions of different mesh blocks, and hence to the increase in resolution achieved by adding one extra mesh block.

As a second drawback of the cut-cell method, it is important to recognize that it is limited by the resolution of the computational grid (Flow Science, Inc., 2011). This limitation is associated with the way area fractions are defined. For each cell face in a mesh, first it is determined which corners of the face are inside and which are outside of a defined geometry component:

- If all four corners of a cell face are inside the component, then the entire face is defined to be within the component.

- Similarly, if all corners lie outside, then the entire face is assumed to be outside the component.

- When some face corners are inside a component and some are outside, the area fraction generator computes the intersection of the component with the face edges. Area fractions are then computed from these intersection points assuming straight-line connections between intersection points within the face. The straight-line assumption introduces a small error in the fractional area when the component boundary is curved inside the cell. The approximation improves as the grid resolution is refined.

The implication of this face construction method is that any piece of a component extending across a cell face, but not including a corner of that face, is not recognized by the area fraction generator. For instance, a spherical subcomponent smaller than a mesh cell will not be recorded unless it covers at least one cell vertex. If the component surface has sharp edges then a multiple intersection is likely to occur, with the cell face intersecting more than one neighboring edge. In this case the corresponding cell edge is assumed to be either fully inside the object or fully outside it, leading to a representation error. The representation is improved as the mesh resolution is increased.

5.4.2 Arrangement of flow variables

With each cell there are associated local average values of all dependent variables. A *staggered* scheme is used to represent fluid velocities and pressures, illustrated in Figure 5.1: u-velocities and fractional areas A_x at the centers of cell faces normal to the x-direction, v-velocities and fractional areas A_y at the centers of cell faces normal to the y-direction, and w-velocities and fractional areas A_z at the centers of cell faces normal to the z-direction. Pressures p, fluid fractions F, volume fractions $V_{F_{i,j,k}}$, area fractions A_i, densities ρ, turbulence quantities (k and ϵ), and viscosity μ are at cell centers.

The staggered arrangement is attractive because it maintains strong coupling between pressure and velocity and does not lead to spurious pressure oscillations, often referred to as the *checkerboard* problem. Furthermore, staggered arrangements do not require ad-hoc boundary conditions for the pressure, and can simultaneously conserve mass, momentum and kinetic energy for an inviscid flow (Harlow and Welch, 1965).

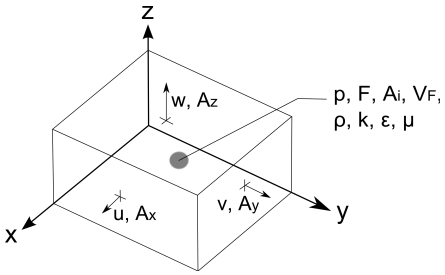


Figure 5.1: Location of variables in a mesh cell.

Before discussing the numerical solution method, some notes on the notation are given, adopted from the FLOW-3D user manual (Flow Science, Inc., 2011). The finite-difference, Cartesian mesh used for numerically solving the governing equations consists of rectangular cells of width δx_i, depth δy_j and height δz_k. The active mesh region has $IBAR$ cells in the x-direction labeled with the index i, $JBAR$ cells in the y-direction labeled with the index j, and $KBAR$ cells in the z-direction labeled with the index k. This region is surrounded by layers of fictitious or boundary cells used to set mesh boundary conditions. In total, there are $(IBAR+2)*(JBAR+2)*(KBAR+2)$ cells in a complete mesh block.

A variable Q at the center of a cell (i,j,k) is denoted as $Q^n_{i,j,k}$, where the superscript n refers to the n-th time step value.

Velocities are located at the cell-faces, denoted as e.g. $u^n_{i,j,k}$ for the x-component located at the middle of the cell face between cells (i,j,k)

and $(i+1, j, k)$ at time level $n\delta t$. Fractional areas are denoted as $AFR_{i,j,k}$ (between cells (i, j, k) and $(i+1, j, k)$), $AFB_{i,j,k}$ (between cells (i, j, k) and $(i, j+1, k)$) and $AFT_{i,j,k}$ (between cells (i, j, k) and $(i, j, k+1)$).

When free surfaces or fluid interfaces are present, it is necessary to distinguish those cells that are empty, contain a surface, or are full of one fluid. By definition, a surface cell is a cell containing fluid #1 and having at least one adjacent cell (at $i \pm 1$, $j \pm 1$, $k \pm 1$), that is empty or full of fluid #2. A cell with an F value less than unity, but with no empty neighbor, is considered a full cell in single-fluid problems. A flag $NF_{i,j,k}$ is used to label the cells and also, in the case of surface cells, to indicate which neighboring cell lies in the direction of the inward normal to the surface. The flag values are indicated in Table 5.1. $NF_{i,j,k}$ is used to indicate the orientation of the interface between two fluids in an analogous manner.

Table 5.1: Definition of flag values $NF_{i,j,k}$

$\boldsymbol{NF_{i,j,k}}$	**interpretation**
0	full or obstacle cell
1	surface ($i-1$ inward neighbor)
2	surface ($i+1$ inward neighbor)
3	surface ($j-1$ inward neighbor)
4	surface ($j+1$ inward neighbor)
5	surface ($k-1$ inward neighbor)
6	surface ($k+1$ inward neighbor)
7	cell undergoing cavitation
8	empty cell

5.4.3 Spatial discretization of momentum terms

The discretization of different terms in the momentum equation containing a spatial derivative is discussed in the following. Finite-difference approximations are formulated, based on Taylor series expansions of flow variables. The terms to be discretized in eq. (5.3) concern, from left to right : momentum advection, a pressure gradient and viscous shear.

In FLOW-3D, the term $\partial \tau_{i,j}/\partial x_j$ refers to internal viscous shear, i.e. away from solid boundaries. Wall shear stresses originating from solid boundaries with a no-slip condition are added separately in the momentum equation as force contributions. With the inclusion of fractional area functions A_i, vanishing at solid walls, it is then straightforward to apply the specific wall-boundary condition. The discretiation of the wall shear stress is also included in this discussion.

The discretization of the transport of turbulent quantities (advection-diffusion) is not treated here. More details on this topic can be found in the user manual (Flow Science, Inc., 2011).

Momentum advection : basic approach

In order to obtain finite-difference approximations to the advective fluxes, momentum control volumes are centered in each cell about the right face for x-momentum, the top face for y-momentum and the back face for z-momentum transport. For brevity's sake, the discussion is limited here to the transport of $u_{i,j,k}$-momentum in the x-direction. Figure 5.2 shows the control volume.

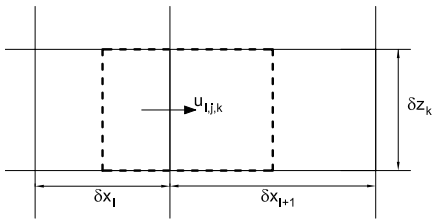


Figure 5.2: Control volume (dashed line) in (x,z)-plane used in finite-difference approximation for $u_{i,j,k}$-momentum (after Flow Science, Inc., 2011).

The advective fluxes can be written either conservatively ($\nabla \boldsymbol{uu}$) or nonconservatively ($\boldsymbol{u}\nabla\boldsymbol{u}$). The conservative approach is preferred, since it automatically ensures the conservation of momentum in a finite-difference approximation. In a non-uniform mesh however, the order of accuracy of the convective flux discretized in the conservative form is reduced by one, because the control volume is not centered about the position of the velocity component u_i at the cell face (Hirt and Nichols, 1981). In order to maintain at least first-order accuracy in a non-uniform mesh, the fluxes are approximated in a non-conservative form in FLOW-3D.

In the basic approach, a first-order upwind and centered-difference approximation are combined into a single expression with a parameter (ALPHA, α) that controls the relative amount of each. The approximation to the advection of u-momentum in x-direction, denoted as FUX, reads:

$$\begin{aligned}(FUX)_{i,j,k} \;=\; & \tfrac{1}{2VFC}\Big[(UAR-\alpha|UAR|)\left(\tfrac{\partial u}{\partial x}\right)_{i+1,j,k} \\ & +(UAL+\alpha|UAL|)\left(\tfrac{\partial u}{\partial x}\right)_{i,j,k}\Big]\end{aligned} \tag{5.32a}$$

$$UAR = \frac{u_{i+1,j,k}AFR_{i+1,j,k} + u_{i,j,k}AFR_{i,j,k}}{2} \tag{5.32b}$$

$$UAL = \frac{u_{i,j,k} AFR_{i,j,k} + u_{i-1,j,k} AFR_{i-1,j,k}}{2} \tag{5.32c}$$

$$VFC = \frac{\delta x_i V_{F_{i,j,k}} + \delta x_{i+1} V_{F_{i+1,j,k}}}{\delta x_i + \delta x_{i+1}} \tag{5.32d}$$

where the velocity gradients in eq. (5.32a) are defined as:

$$\left(\frac{\partial u}{\partial x}\right)_{i,j,k} = \frac{u_{i,j,k} - u_{i-1,j,k}}{\delta x_i} \tag{5.33a}$$

$$\left(\frac{\partial u}{\partial x}\right)_{i+1,j,k} = \frac{u_{i+1,j,k} - u_{i,j,k}}{\delta x_{i+1}} \tag{5.33b}$$

UAR and UAL are the advecting velocities, averaged between the cell face velocity $u_{i,j,k}$ and the velocities at cells to the right and left of cell (i,j,k), respectively. Similar terms FUY and FUZ are developed for the advection of u in y- and z-direction respectively, as well as terms *FVX(Y,Z)* and *FWX(Y,Z)* for the advection of v- and w-momentum.

The basic idea underlying eqs. (5.32) is to weight the upstream quantity being fluxed more than the downstream value. The weighting factors are $(1+\alpha)$ and $(1-\alpha)$ for the upstream and downstream direction, respectively. The stream direction is determined by the sign of velocities UAR and UAL. When $\alpha = 0$, the approximation reduces to a centered-difference approximation that is spatially second-order accurate when the mesh is uniform. When $\alpha = 1$, the first-order, upwind discretization is retained.

The presence of an obstacle is directly taken into account by incorporation of the area functions in the advecting velocities and by the factor VFC, defined as the averaged volume fraction of both cells (i,j,k) and $(i+1,j,k)$ surrounding $u_{i,j,k}$. It is noticed that the formulation of the advecting velocity in eq. (5.32a) is different from other formulations from predecessor codes (e.g. NASAVOF-2D, RIPPLE), where the advecting velocity is taken as the face velocity $u_{i,j,k}$. The formulation of eq. (5.32a) has the advantage however to reduce to a conservative approximation when the mesh is uniform.

Second-order approximation to momentum advection

The first-order upwind approximation yields stable results and has the property of *monotonicity*, which means that no new extrema are created and the value of a local minimum/maximum is non-decreasing/non-increasing in time. However, the upwind scheme also leads to high levels of numerical dissipation, causing sharp gradients of the convected quantity to be smoothed out in subsequent time levels. This may require a relatively

high mesh resolution to obtain an accurate first-order solution. In such cases, it can be worthwhile to use higher-order schemes that are less prone to numerical dissipation. In FLOW-3D, two different second-order approximations methods are incorporated.

The essence of the first method is a double pass through the first-order advection subroutine in the code. In the first pass, the first-order downwind method is used with $\alpha = -1$. The resulting velocities are then stored in the arrays for the previous time velocities. The first-order calculations are then repeated, but this time with the upwind-differencing approximation ($\alpha = 1$). Finally, the results of the two calculations are averaged to give the desired second-order approximation to the new time-level velocities. The resulting approximation is second-order in time in general, and second-order in space in a uniform mesh. This algorithm is the least numerically diffusive of the three advection methods available in FLOW-3D. However, it does not possess the transportive property (monotonicity), which may lead to instabilities (*wiggles*) in the solution.

The other method is based on the second-order monotonicity-preserving upwind-difference method by Van Leer (1977). The higher-order discretization scheme consists of second-order polynomial approximations to the advected quantity, using a splitting method in which each coordinate direction is treated separately. The method is applicable to momentum advection as well as scalar quantities such as density, (turbulent) energy and fluid fraction. The method is briefly described here, limiting the discussion to the approximation of a variable Q advected in the x-direction. More details on the method are given by Bronisz and Hirt (1991).

To achieve a second-order accurate approximation in a non-uniform mesh, the approximation to the value fluxed through a cell face, denoted as Q^*, needs to be third-order. The approach adopted by Van Leer (1977) is then to make polynomial approximations to the variable Q in function of h, the distance measured from the center of cell (i, j, k):

$$Q(h) = Q_i + Ah + \frac{1}{2}B\left(h^2 - 2hh_0 - \frac{1}{12}\delta x_i^2\right) + O(h^3) \tag{5.34}$$

where Q_i is the cell-centered value of Q, and A and B are defined in combination with h_0 such that:

$$\frac{\partial Q}{\partial h} = A + O(h^2) \; at \; h = h_0 \tag{5.35a}$$

$$\frac{\partial^2 Q}{\partial h^2} = B + O(h) \tag{5.35b}$$

h_0 is thus the location where A is a second-order approximation to the first derivative of Q.

The third-order accurate approximation to Q^* is obtained by integrating eq. (5.34) over the volume of the cell that is fluxed across the boundary in one time step δt, i.e. from $h = \delta x_i/2 - u_{i,j,k}\delta t$ to $h = \delta x_i/2$. If h_0 is carefully chosen, the approximation simplifies to:

$$Q^* = Q_i + \frac{A(1-C)\delta x_i}{2} \tag{5.36}$$

where Q_i is the cell-centered value and C is the CFL number ($u_{i,j,k}\delta t/\delta x_i$). Eq. (5.36) holds exclusively for the location:

$$h_0 = \frac{(1-2C)\delta x_i}{6} \tag{5.37}$$

The coefficient A can easily be computed from two neighboring first derivatives by linear interpolation, provided these derivatives are second-order accurate. The latter can be achieved by computing the derivatives at the midpoints between Q_i locations; for example,

$$\left(\frac{\partial Q}{\partial x}\right)_{i+1/2} = 2\frac{Q_{i+1} - Q_i}{\delta x_{i+1} + \delta x_i} \tag{5.38}$$

is a second-order accurate first derivative of Q at the point between Q_i and Q_{i+1}. With this approach, the extension of the second-order, monotonicity-preserving method to non-uniform grids is straightforward.

To ensure monotonicity, it is necessary to restrict the value of the derivative A to twice the minimum magnitude of the centered Q-derivatives used in its computation (Van Leer, 1977):

$$A \leq 2min\left(\frac{dQ}{\delta x_i}, \frac{dQ}{\delta x_{i+1}}\right) \tag{5.39}$$

Furthermore, if Q_i is a local minimum or maximum value - that is, if the two centered derivatives appearing in the equation (5.39) are of opposite sign - then A is set to zero and the method reduces to a first-order upwind approximation.

Pressure gradient

Pressure is evaluated in the center of cell (i, j, k). The discretized form of the pressure gradient $-\frac{1}{\rho}\frac{\partial p}{\partial x_i}$ reads, e.g. in x-direction:

$$-\frac{1}{\rho}\frac{p_{i+1,j,k} - p_{i,j,k}}{\delta x_{i+1/2}} \tag{5.40}$$

In the notation of cell dimensions, the fractional subindex implies an averaging of neighboring cell dimensions, e.g. for the x-direction:

$$\delta x_{i+1/2} = \frac{\delta x_i + \delta x_{i+1}}{2} \ , \ \delta x_{i-1/2,j,k} = \frac{\delta x_{i-1} + \delta x_i}{2} \tag{5.41}$$

Viscous shear stress

The term $\partial\tau_{ij}/\partial x_i$ in eq. (5.3), representing internal viscous shear (i.e. away from solid boundaries), is discretized using a standard second-order central-differencing scheme. For an incompressible fluid, the discretization in the x-direction reads:

$$\begin{aligned}(VISX)_{i,j,k} &= \frac{1}{\rho}\left(\frac{\partial\tau_{xx}}{\partial x}+\frac{\partial\tau_{yx}}{\partial y}+\frac{\partial\tau_{zx}}{\partial z}\right)_{i,j,k} \\ &= \frac{1}{\rho}\left(\frac{(\tau_{xx})_{i+1,j,k}-(\tau_{xx})_{i,j,k}}{\delta x_{i+1/2}}+\frac{(\tau_{yx})_{i,j,k}-(\tau_{yx})_{i,j-1,k}}{\delta y_j}\right. \\ &\quad \left.+\frac{(\tau_{zx})_{i,j,k}-(\tau_{zx})_{i,j,k-1}}{\delta z_k}\right)\end{aligned} \tag{5.42}$$

where the normal stress component is approximated in the cell center:

$$(\tau_{xx})_{i,j,k} = 2\mu_{i,j,k}\left(\frac{u_{i,j,k}-u_{i-1,j,k}}{\delta x_i}\right) \tag{5.43}$$

and tangential stress components at the vertices, e.g.:

$$(\tau_{yx})_{i,j,k} = \mu_{i,j,k}\left(\frac{u_{i,j+1,k}-u_{i,j,k}}{\delta y_{j+1/2}}+\frac{v_{i+1,j,k}-v_{i,j,k}}{\delta x_{i+1/2}}\right) \tag{5.44}$$

Similar terms $VISY$ and $VISZ$ are developed in y- and z-direction, respectively. In case a RANS turbulence model is used, the viscosity is evaluated as the effective viscosity according to eq. (5.11).

Due to the application of the porous media flow model, which includes the microscopic viscous shear, terms *VISX(Y,Z)* inside porous media are to be considered as macroscopic, related to gradients in seepage velocity.

Wall shear stress

The wall shear stresses are modeled assuming a zero relative tangential velocity on solid boundaries with the no slip-condition prescribed. It is important to remark that the approach in FLOW-3D, modeling the wall shear stresses as a separate term in the momentum equation, only holds for solid boundaries, since in that case the interface between the solid boundary and the fluid is resolved. Under the averaging approach (seepage velocity concept, see section 2.3.2), microscopic velocity gradients and resulting shear stresses at the interface between a porous medium and the clear fluid cannot be computed. Instead, all viscous and pressure forces are lumped into a flow loss term, described by the porous media flow model (Barkhudarov, 2012).

The wall shear acceleration for the w-velocity equation is derived here, denoted as WSZ. Terms $WSX(Y)$ are likewise obtained in the remaining directions. Wall shear influencing w can arise from any of the four wall

areas located on x or y cell-faces surrounding $w_{i,j,k}$. For any one of these faces, if the fractional flow area A_i is less than unity, the remaining area fraction $(1 - A_i)$ is considered to be a wall on which a stress is generated.

If the flow is treated laminar, the acceleration due to tangential wall shear is proportional to the molecular viscosity μ and local velocity gradients, and can be approximated as:

$$\frac{1}{\rho}\frac{\partial}{\partial x}\left(\mu\frac{\partial w}{\partial x}\right) \tag{5.45}$$

For instance on an x-face to the right of $w_{i,j,k}$, the discretized approximation to eq. (5.45) reads:

$$(WSZ)_{i,j,k} = -\frac{2\mu_{i,j,k}}{\rho V_{F_{i,j,k}} AFT_{ijk}\delta x_i^2}\left[1 - \frac{(AFR_{i,j,k} + AFR_{i,j,k+1})}{2}\right](w_{i,j,k} - w_0) \tag{5.46}$$

where the velocity w_o is either zero or equal to the z-direction tangential velocity at a moving solid boundary. Because $w_{i,j,k}$ is located on the boundary between cells (i, j, k) and $(i, j, k + 1)$, an averaged value for the fraction area AFR is used. Similar stress components as in eq. (5.46) are evaluated at each of the four surrounding cell walls, and their sum is taken as the total stress WSZ.

For turbulent flows, a logarithmic velocity profile according to eq. (5.28) is assumed near the wall, which modifies the wall shear stress magnitude. Using the definition of the shear velocity eq. (5.29), the approximation to the wall shear acceleration reads:

$$(WSZ)_{i,j,k} = \frac{1}{V_{F_{i,j,k}}\delta x_i}\left[1 - \frac{(AFR_{i,j,k} + AFR_{i,j,k+1})}{2}\right](u_z^\star)^2 \tag{5.47}$$

Since the FAVOR method does not precisely locate wall locations within a cell, approximations must be introduced to find $u_\parallel$, $u^\star$ and d. For this purpose, the direction of the wall normal in the cell is first determined, and $u_\parallel$ is computed as the component of the cell-centered velocity parallel to the wall (relative to the wall velocity in the considered direction). The average distance to the wall (d_0) is estimated to be half of the cell width in the wall normal direction. That is, the triplet (δx_i,δy_j,δz_k) is treated as a vector whose inner product with the wall normal is defined as the cell width in the normal direction. Finally, $u^\star$ is iteratively computed from eq. (5.28) in terms of $u_\parallel$ and d_0.

Both laminar and turbulent wall shear stresses can be modified by defining a wall roughness length, specified through the parameter ROUGH. The wall roughness length is incorporated into the usual shear stress calculations by adding to the molecular viscosity the product of ρ, ROUGH and u_i, the latter being the difference between the local fluid velocity and the

wall velocity in the considered direction. For turbulent flow, the law-of-the-wall relation retains the same form as for a smooth wall, except the change in viscosity (i.e., from ν_{eff} to $\nu_{eff} + \text{ROUGH} * u_i$) automatically converts the logarithm dependence from a characteristic length scale defined by ν_{eff}/u_i to the roughness length, when ROUGH is the larger of the two characteristic lengths.

5.4.4 Time advancement of conservation equations

The spatial discretization of the conservation equations results in a system of coupled ordinary differential equations with respect to time, in which pressure and velocity are the unknowns. In the following, the method to compute the equations one increment in time (δt) is discussed.

First, the discretized form of the conservation equations is given. In case of an incompressible fluid, the mass conservation eq. (5.2) reads:

$$\frac{AFR_{i,j,k}u^{n+1}_{i,j,k} - AFR_{i-1,j,k}u^{n+1}_{i-1,j,k}}{\delta x_i} + \frac{AFB_{i,j,k}v^{n+1}_{i,j,k} - AFB_{i,j-1,k}v^{n+1}_{i,j-1,k}}{\delta y_j} + \frac{AFT_{i,j,k}w^{n+1}_{i,j,k} - AFT_{i,j,k-1}w^{n+1}_{i,j,k-1}}{\delta z_k} = 0 \tag{5.48}$$

Note that the presence of an obstacle, either solid or porous, is accounted for in eq. (5.48) by inclusion of the area fractions ($AFR_{i,j,k}$,$AFB_{i,j,k}$,$AFT_{i,j,k}$).

The momentum eqs. (5.3) for each direction read, with gravity (g) as the only acting body force:

$$\frac{u^{n+1}_{i,j,k} - u^{n}_{i,j,k}}{\delta t} = -(FUX + FUY + FUZ)^n_{i,j,k} - \frac{1}{\rho}\frac{p^{n+1}_{i+1,j,k} - p^{n+1}_{i,j,k}}{\delta x_{i+1/2}} + g_x + (VISX)^{n+1|n}_{i,j,k} - F_d u^{n+1}_{i,j,k} - (WSX)^{n+1}_{i,j,k} \tag{5.49a}$$

$$\frac{v^{n+1}_{i,j,k} - v^{n}_{i,j,k}}{\delta t} = -(FVX + FVY + FVZ)^n_{i,j,k} - \frac{1}{\rho}\frac{p^{n+1}_{i,j+1,k} - p^{n+1}_{i,j,k}}{\delta y_{j+1/2}} + g_y + (VISY)^{n+1|n}_{i,j,k} - F_d v^{n+1}_{i,j,k} - (WSY)^{n+1}_{i,j,k} \tag{5.49b}$$

$$\frac{w^{n+1}_{i,j,k} - w^{n}_{i,j,k}}{\delta t} = -(FWX + FWY + FWZ)^n_{i,j,k} - \frac{1}{\rho}\frac{p^{n+1}_{i,j,k+1} - p^{n+1}_{i,j,k}}{\delta z_{k+1/2}} + g_z + (VISZ)^{n+1|n}_{i,j,k} - F_d w^{n+1}_{i,j,k} - (WSZ)^{n+1}_{i,j,k} \tag{5.49c}$$

where the terms *FU(V,W)X(Y,Z)* and *VISX(Y,Z)* discussed in the previous section denote the convective fluxes and acceleration due to internal viscous shear, respectively. F_d is the porous drag coefficient according to eq. (5.5) and *WSX(Y,Z)* the acceleration due to the wall shear stress.

The superscript $n+1|n$ refers to the time discretization scheme, which can be either implicit (evaluation at time level t^{n+1}) or explicit (at time level t^n). Convective fluxes are treated explicitly by default. Viscous shear stresses can be treated either explicitly or implicitly, upon judgment of the user which method is most suitable for solving the specific flow problem. Wall shear stresses are treated in an implicit way to avoid possible numerical instabilities arising in cells with large wall areas and small flow volumes. An implicit treatment is then no longer susceptible to the time step size. Since the wall shear terms are linear in flow velocities, their solution is straightforward.

Flow in porous media are modeled with a drag force proportional to the first power of the velocity, $F_d u_i$. For the applications with coarse granular media, the formulation according to eq. (5.5) is suitable. To compute a limit to the drag term for incompressible flow, it is necessary to treat the drag terms implicitly, not only in the momentum equations but also in the continuity equation. This is accomplished by using the velocity in the drag term at time level $n+1$ and algebraically solving the difference equation for the new velocity. The result is a division of all contributions to the new velocity by the term $(1 + F_d \delta t)$. Keeping the effect of this extra term throughout all pressure/velocity adjustments then ensures that a balance between pressure gradient and drag forces can be achieved that also satisfies the continuity equation.

The major difficulty related to the time advancement of the discrete NS equations is that the mass-conservation equation does not contain an explicit derivative in time if the flow is incompressible. The incompressibility constraint rather acts as a kinematic constraint to the velocity field and couples pressure and velocity implicitly. The pressure can be considered as an auxiliary variable needed to maintain the incompressibility constraint. Fractional-step methods (Chorin, 1968) are without any doubt the most widespread technique to decouple the computation of the pressure from the advancement of the momentum equation. The advantage of such an approach is that the decoupled systems for p and u_i can be solved at a lower expense.

The basic idea of the fractional-step method is to isolate the pressure gradient from the other terms in the momentum equation and use it for the projection of the velocity field onto a solenoidal field. The several steps of this method include:

1. The *intermediate* velocities u_i^* are computed from the current-time (t^n) advective, pressure, and other accelerations:

$$\frac{u^*_{i,j,k} - u^n_{i,j,k}}{\delta t} = -(FUX + FUY + FUZ)^n_{i,j,k} - \frac{1}{\rho}\frac{p^n_{i+1,j,k} - p^n_{i,j,k}}{\delta x_{i+1/2}} + g_x + (VISX)^{*|n}_{i,j,k} - F_d u^*_{i,j,k} - (WSX)^*_{i,j,k} \tag{5.50a}$$

$$\frac{v^*_{i,j,k} - v^n_{i,j,k}}{\delta t} = -(FVX + FVY + FVZ)^n_{i,j,k} - \frac{1}{\rho}\frac{p^n_{i,j+1,k} - p^n_{i,j,k}}{\delta y_{j+1/2}} + g_y + (VISY)^{*|n}_{i,j,k} - F_d v^*_{i,j,k} - (WSY)^*_{i,j,k} \tag{5.50b}$$

$$\frac{w^*_{i,j,k} - w^n_{i,j,k}}{\delta t} = -(FWX + FWY + FWZ)^n_{i,j,k} - \frac{1}{\rho}\frac{p^n_{i,j,k+1} - p^n_{i,j,k}}{\delta z_{k+1/2}} + g_z + (VISZ)^{*|n}_{i,j,k} - F_d w^*_{i,j,k} - (WSZ)^*_{i,j,k} \tag{5.50c}$$

where terms with a superscript * are evaluated with intermediate velocities, i.e. implicitly. In case the viscous shear stresses are treated explicitly, solving for intermediate velocities is straightforward. In case they are treated implicitly, different solver algorithms (Jacobi iteration or an Alternating Direction Implicit (ADI) method) are used to solve for intermediate velocities u^*_i, depending on the pressure-velocity method in the next step (Yao, 2004). In case the GMRES solver is used (see next step), a Generalized Conjugate Gradient (GCG) algorithm is applied.

2. The 'new' velocity at time level $n+1$ is related to the intermediate one u^*_i through the following relationship:

$$\frac{u^{n+1}_{i,j,k} - u^*_{i,j,k}}{\delta t} = -\frac{1}{\rho}\frac{p'_{i+1,j,k} - p'_{i,j,k}}{\delta x_{i+1/2}} \tag{5.51a}$$

$$\frac{v^{n+1}_{i,j,k} - v^*_{i,j,k}}{\delta t} = -\frac{1}{\rho}\frac{p'_{i,j+1,k} - p'_{i,j,k}}{\delta y_{j+1/2}} \tag{5.51b}$$

$$\frac{w^{n+1}_{i,j,k} - w^*_{i,j,k}}{\delta t} = -\frac{1}{\rho}\frac{p'_{i,j,k+1} - p'_{i,j,k}}{\delta z_{k+1/2}} \tag{5.51c}$$

where $p'_{i,j,k} = p^{n+1}_{i,j,k} - p^n_{i,j,k}$ represents the pressure change in each cell (i,j,k). Substitution of eqs. (5.51) into the mass continuity eq. (5.48) yields the *pressure-Poisson* equation (written here in differential form for the sake of brevity):

$$\frac{\partial(A_i u^*_i)}{\partial x_i} = \frac{\delta t}{\rho}\frac{\partial^2(A_i p')}{\partial x_i^2} \tag{5.52}$$

Basically two methods are incorporated in FLOW-3D in order to solve eq. (5.52). The first one, referred to as Successive-Over-Relaxation (SOR) uses a Newton type of relaxation process, adjusting the pressures on a cell-by-cell basis to enforce the mass conservation. A second method, using the Generalized Minimal Residual (GMRES) algorithm, solves the linear system of eq. (5.52) simultaneously throughout the domain by an

iterative technique (Brethour, 2009). In both algorithms, a convergence criterion is applied to the velocity divergence in each cell:

$$\left|\frac{\partial u_i}{\partial x_i}\right| \leq \text{EPSADJ} * 9.10^{-5}\delta t^{-1} \tag{5.53}$$

Convergence can be tightened optionally by setting EPSADJ to a value lesser than 1. It is also possible to supply a fixed value to the convergence parameter EPSI, which then replaces the right-hand-side in eq. (5.53) and makes the convergence criterion independent from δt.

3. After convergence is obtained, the velocities a time level $n + 1$ are corrected with the gradient of the new pressure p' using eqs. (5.51).

4. In a last step in fluid problems with a free surface or fluid interface, eq. (5.22) must be updated to give the new fluid configuration. The advection of other scalars (e.g. turbulence quantities) is also performed in this step. The next subsection discusses the time-advancement of fluid configuration into more detail.

Repetition of these steps will advance a solution through any desired time interval. At each step, suitable boundary conditions must be imposed at all mesh, obstacle, and free-surface boundaries. More details on the numerical implementation of these boundary conditions can be found in the user manual (Flow Science, Inc., 2011).

5.4.5 Time advancement of fluid configuration

For the general case of fluid flow in the presence of an obstacle, either porous or solid, the kinematic free-surface boundary condition eq. (5.22) needs to be discretized (written here in the absence of mass sources and, in case of two-phase flow, neglecting turbulent mixing):

$$V_F \frac{\partial F}{\partial t} + \frac{\partial}{\partial x_i}\left(A_i F u_i\right) = 0 \tag{5.54}$$

The numerical solution of eq. (5.54) must prevent unphysical distortion of the free surface and preserve its sharpness (Barkhudarov, 2004). The original VOF advection method developed for both single and two-fluid problems (referred to as the 'standard method' in FLOW-3D) is based on the donor-acceptor approach first introduced by Hirt and Nichols (1981). Numerous enhancements have been made to the original algorithm to improve its accuracy and stability in complex one- and two- fluid flows with sharp interfaces. The standard method uses operator splitting and old time-level values of the F-function to compute fluxes in three coordinate

directions. The approach creates a possibility of overfilling or over-emptying computational cells when volume fluxes are significant in all three directions and when δt is close to the local Courant stability limit (see section 5.4.6).

More recently developed methods are referred to as the *unsplit* and *split* Lagrangian method in FLOW-3D. These advection methods, suitable for both single and two-phase flow, have been developed to alleviate the aforementioned deficiencies of the standard algorithm. The fluid interface is reconstructed in 3D using a piecewise linear representation, where the interface is assumed to be planar in each control volume (or cell) containing the interface. The fluid volume bounded by the interface and cell faces is then moved according to the local velocity vector in a Lagrangian manner. Finally, the advected volume is overlaid back onto the Eulerian grid to obtain the new values of the F-function. This combination of the Lagrangian and Eulerian methodology gives the method its name. There is no difference in how the interface is reconstructed in both Lagrangian methods. The difference is only in how fluid is moved after the reconstruction. In the 'unsplit' method, it is moved along the 3D velocity vector, while in the 'split' method, it is moved first in x-direction, then in y-direction and finally in z-direction, with the interface being reconstructed after each step (Barkhudarov, 2012).

Generally, the two Lagrangian methods exhibit good accuracy in tracking sharp interfaces in complex 3D motions. The split Lagrangian method typically produces lower cumulative volume error than the other methods, although the volume error may increase when this method is used together with the moving obstacle model. More details on the Lagrangian methods are given by Barkhudarov (2004).

When applying the free-surface boundary conditions and advection of the F-function, it is necessary to determine an approximate direction normal to the free surface. The neighboring cell closest to the direction of the inward normal to the surface is recorded by specifying integer values of cell flags $NF_{i,j,k}$ (see Table 5.1). More details are provided in Flow Science, Inc. (2011).

Methods are available in the code to suppress instabilities related to *misty* fluid regions, i.e. isolated fluid drops due to excessive splashing and free-surface breakup. The adjustment consists in artificially removing the fluid distribution in misty regions, for a cell (and all adjacent cells) with a value of F below the parameter FCLEAN, typically between 0 and 0.1. *Foaming* is another issue related to the extreme deformation of the free surfaces. In this case, an algorithm is available to eliminate the small voids in a flow with significant free-surface breakup. The algorithm, referred to as F-packing and only used in single-fluid flows, works by creating small negative divergences in internal fluid cells in which the fluid fraction is less

than 0.99. The rate of F-packing is proportional to the coefficient CFPK. The default value is 1, when it is equal to 0 no packing will take place.

5.4.6 Stability conditions and time step control

The explicit schemes previously discussed need a limitation to the time step size δt in order to remain stable. Several criteria are applied to the maximum allowable timestep:

1. The fluid must not be permitted to flow across more than one computational cell in one time step. This advective transport depends not only on the velocity but also on the fractional area/volume open to flow. The basic stability condition is a modification of the CFL condition:

$$\delta t < \text{CON} * min\left(\frac{V_F \delta x_i}{A_x u}, \frac{V_F \delta y_j}{A_y v}, \frac{V_F \delta z_k}{A_z w}\right) \tag{5.55}$$

   The default value of CON amounts to 0.45 in case of incompressible flow. A cell with a large open face area and a small volume could restrict the time step to small values if there is significant flow in this cell. Should this happen, it can be determined by monitoring the mesh locations controlling the time step as printed in the output. In this case, it may be necessary to modify the mesh/obstacle arrangement. An algorithm is implemented in the code which automatically adjust these ratios by making small adjustments in volume fractions.

2. Free surfaces also introduce another type of stability condition associated with the propagation of surface waves. If an acceleration ACCN is applied to the fluid in a direction normal to the free surface (in this case ACCN$= g$, the gravitational constant), there can be surface waves with speeds of order $\sqrt{ACCN.h}$, where h is the depth of fluid or length of the wave. In practice, the cell size in the normal direction is used for h, together with an extra safety factor of 0.5 in the stability test. The actual condition is that surface waves should not propagate more than one cell in one time step. For example, if z is the normal direction to the surface and ACCZ is the normal acceleration, then the stability limit reads:

$$\delta t < 0.5 * \frac{min(\delta x_i, \delta y_j)}{\sqrt{\delta z_k ACCZ}} \tag{5.56}$$

   Similar limits must be imposed in the x- and y-directions for each cell containing a free surface.

3. A linear analysis indicates that the time step must be further limited when a non-zero value of dynamic viscosity is used. This condition is

$$\delta t < 0.25 * max \left[RM\ \nu \left(\frac{1}{\delta x_i^2} + \frac{1}{\delta y_j^2} + \frac{1}{\delta z_k^2} \right) \right]^{-1} \tag{5.57}$$

   where RM is the maximum multiplier used on ν for all types of diffusional processes. The restriction physically means that no quantity should diffuse more than approximately one mesh cell in one time step. For safety, an extra factor of 0.5 has been incorporated in the right side of eq. (5.57) because the limit is otherwise marginal.

4. A last stability criterion regards the choice of the parameter ALPHA in the first-order momentum advection approximation. When ALPHA=1.0 is used, the previous stability conditions are sufficient. Generally, a value for ALPHA should satisfy the following condition:

$$\delta t * max \left(\frac{|u|}{\delta x_i}, \frac{|v|}{\delta y_j}, \frac{|w|}{\delta z_k} \right) < \text{ALPHA} \leq 1.0 \tag{5.58}$$

   If the first condition eq. (5.55) is fulfilled, eq. (5.58) implies that ALPHA is larger than CON.

The different options to control the evolution of δt are listed in Table 5.2 below.

Table 5.2: Time step control options

AUTOT	evolution of δt
0	constant
1	controlled by stability limit and number of pressure iterations
2	controlled by stability limit

If a constant time step is used (AUTOT=0), the initial value can be set by specifying δt, defaulted to TWFIN/100 (TWFIN being the total computation duration). However, δt will be changed, even if a constant value has been requested, when the advection of fluid fraction exceeds an amount equal to twice the volume of the cell times the stability factor CON (or 0.85 times the volume of the cell if this is smaller). In this case the solution is returned to its state at the beginning of the cycle before continuing, and the cycle is repeated with the time step cut in half.

If AUTOT is set to 1, the code will adjust the time step to be as large as possible without violating stability conditions or exceeding the user-supplied

maximum time-step size (DTMAX). δt will also be reduced when pressure iterations exceed corresponding nominal values that depend on the iteration options. Generally, the time step will float up or down with 5% changes per cycle unless a stability condition is violated, in which case a larger reduction may occur. The maximum number of pressure iterations before reducing the time step can be specified by means of the parameter ITDTMAX, defaulted to 10 when the GMRES pressure solver is used.

If AUTOT=2 is selected, the time step size will not be reduced if the number of pressure iterations exceeds a certain value. The number of pressure iterations per cycle is limited by ITMAX, defaulted to 100 for the GMRES pressure solver.

5.5 Conclusions

Prerequisites to a numerical model for wave interaction with permeable coastal structures are an effective treatment of the free surface and porous media flow. The CFD code FLOW-3D has been selected to this purpose. In this chapter, the basic model equations are discussed and a synopsis of their numerical implementation is provided, for a better understanding of the work developed subsequently.

FLOW-3D uses a cut-cell method for obstacle representation, together with structured rectangular grids and a staggered mesh topology. The philosophy behind this approach is to automize the grid generation process as far as possible. The definition of volume and area fraction functions enables a generalized approach for obstacle representation, applicable to both solid and porous obstacles.

Modeling of free-surface flows is achieved with the VOF method. In addition to the method based on the original donor-acceptor approach (Hirt and Nichols, 1981), more contemporary methods including a PLIC reconstruction scheme are available.

Flow in coarse granular media can be represented by a Forchheimer drag term in the momentum equations. The impact on the mass continuity and kinematic free-surface boundary condition is automatically fulfilled by the inclusion of volume and area fraction functions.

6 | A numerical wave flume in FLOW-3D®

6.1 Introduction

FLOW-3D is a general multi-purpose CFD code. It disposes of a standard wave boundary condition to generate regular and irregular surface gravity waves. However, to enable its operation as a numerical wave flume, it is necessary to run test series with regular and irregular waves which remain stable for a long period in time. Hence, measures need to be taken to avoid re-reflection at the wave generation boundary, which would lead to an excessive increase of the total wave energy in the flume and disturb the desired incident wave field. Such a feature of the wave boundary condition is called *active absorption*.

The wave boundary condition that comes standard with the code however has no such technique for simultaneous absorption of the reflected waves. This compelled to develop additional techniques for wave generation. Because the code has a standard model for moving objects, an interesting track consists of simulating the movements of a piston wavemaker, as it is employed in a physical wave flume. Existing techniques for controlling wavemakers with active absorption are well validated, and can be directly adopted in a numerical wave flume.

This chapter describes the implementation of a first-order piston wavemaker, which is the most simple type of wavemaker. First, details are given on the implementation of a control system for the piston motion. This is followed by an extensive set of validation tests, using wave conditions with varying nonlinearity. The specific objectives of the tests are to investigate the different parameters that can be selected in the numerical model, affecting the operation of the piston and the propagation of free-surface waves. In brief, the main objectives are to:

- validate the piston wave generation, paying special attention to the time-stability of long-duration wave simulations;
- validate the performance of the active wave absorption;
- identify the key numerical parameters controlling the operation of the piston and the simulation of progressive waves, and obtain optimal settings for those parameters.

6.2 Wave generation methods

The 2D Dirichlet wave boundary condition for regular and irregular waves that comes standard with FLOW-3D, prescribing surface elevations and velocity components at the mesh boundary, is probably the most efficient and widely-used method for numerical wave generation. However, the implementation in FLOW-3D does not include active wave absorption. Although it would be possible to extend the wave boundary condition with a method for active absorption, an alternative path is explored in this work, modeling the movements of a piston wavemaker. The main benefit of this approach is that the total amount of fluid within the wave flume is automatically preserved. This is generally not the case for a wave boundary condition, where a net inflow of fluid mass occurs over one wave period. Particularly for highly-nonlinear waves and long test durations, the accumulation of fluid inflow might eventually cause a significant deviation from the initial water level, hence disturbing the hydraulic boundary conditions of the considered flume test. Additional corrections are often needed in order to compensate for the net inflow of fluid volume through the wave boundary.

In the following section, the implementation of a piston wavemaker for the generation of regular and irregular waves is presented, followed by a description of the active absorption method. A short description of the standard FLOW-3D wave boundary conditions is also included, since both wave generation methods will be used in the validation tests.

6.2.1 Piston wavemaker

The piston wavemaker is implemented in FLOW-3D by means of the General Moving Object (GMO) model, which simulates rigid body motion. The motion of the piston will be prescribed for a *position-controlled* piston, a type of wave generator which is operated in many laboratories. Here, the most simple type of piston wavemaker is implemented, with a vertical face moving horizontally above the flume bed. The motion of the piston is assumed to be independent from fluid interaction.

Generation of regular waves

Biesel and Suquet (1951) provided the theoretical background in order to link a sinusoidal piston displacement $X(t)$=$e$sin(ωt) to the surface elevation $\eta(x,t)$ of the progressive wave component generated by the piston:

$$\eta(x,t) = eK_f \sin(\omega t - kx + \varphi_f) + eK_n \sin(\omega t) \tag{6.1a}$$

$$K_f = \frac{4\sinh^2(kh)}{\sinh(2kh) + 2kh} \tag{6.1b}$$

$$K_n = \sum_{n=1}^{\infty} \frac{2\sin^2(k_n h)\exp(-k_n x)}{\sin(k_n h)\cos(k_n h) + k_n h} \tag{6.1c}$$

where e is half the piston stroke S_0, ω= $2\pi/T$ the pulsation frequency and k and k_n the wavenumbers obtained from the linear dispersion equations:

$$\omega^2 = kg\tanh(kh) \tag{6.2a}$$

$$\omega^2 = -k_n g\tanh(k_n h), n > 0 \tag{6.2b}$$

where k_n is the n^{th} real positive solution of eq. (6.2b). $K_n(k_n h)$ and $K_f(kh)$ are transfer functions for the near and far field, respectively. The first term in eq. (6.1a) describes a progressive wave with amplitude eK_f at large distance from the piston, the *far-field* solution. The far-field surface elevation is shifted in phase with respect to the displacement of the wavemaker. In case of a piston wavemaker, φ_f equals = $\pi/2$.

The second term in eq. (6.1a) refers to the *near-field* solution or *evanescent modes* (Schäffer, 1996), representing a series of standing waves near the piston, in phase with the piston movement. They are caused by the difference between the velocity profile of the progressive wave and the uniform velocity profile created by the piston. The transfer function K_n contains a negative exponential factor to account for the decreasing amplitude of evanescent modes with increasing distance from the piston. In practice, the near-field solution can be discarded at a sufficient distance from the piston, e.g. 3 times the water depth h (Dean and Dalrymple, 1991). According to Frigaard et al. (1993), the disturbance from the near-field solution will be less than 1% of the far-field solution at a distance of approximately one to two wave lengths from the piston.

In order to generate a sinusoidal monochromatic progressive wave with wave height H and wave number k in a water depth h, the piston stroke $S_0 = 2e$ becomes:

$$S_0 = \frac{H}{K_f(kh)} \tag{6.3}$$

Limitations to the operation of the piston wavemaker exist. The limit of K_f in eq. (6.1b) tends to zero when kh approaches 0. Consequently, the piston stroke S_0 would need to become very large in order to generate long waves. Practical limitations, both in the physical and the numerical wave flume restrict the maximum allowable piston stroke, which renders the generation (and absorption) of long waves difficult.

The GMO model in FLOW-3D requires the prescription of the object velocity, which implies that piston control velocities rather than displacements need to be specified. A customizable subroutine *mvbvel_usr.f* is used to pass the velocities to the moving object, in order to generate the desired progressive wave field. Existing LabView® subroutines, used in the piston control software of the wave flume of the Civil Engineering Dept. (Ghent University) were modified to generate the file *piston_left.vel*, containing the horizontal piston velocities derived from the piston displacements. The sampling rate of piston control velocities are specified with the parameter f_s.

Generation of irregular waves

The previous method can be equally used to generate irregular waves. Here, the so-called *random phase method* is used (Tuah and Hudspeth, 1982), a deterministic method producing wave trains of finite durations which match the specified wave characteristics exactly.

Random waves are simulated in the frequency domain and subsequently transferred using a Fast Fourier Transfer (FFT) algorithm in order to obtain the time series of surface elevations. The discrete amplitude wave spectrum corresponding to the target wave energy spectrum is combined with a random phase spectrum synthesized from a random number generator. The several steps to obtain the piston control signal include:

1. defining a target wave energy density spectrum $S_\eta(f)$, according to e.g. Pierson-Moskowitz (Pierson and Moskowitz, 1964) or JONSWAP (Hasselmann et al., 1976);

2. choosing the sample frequency f_s and spectral resolution N (half the number of Fourier components). This yields the frequency domain resolution $\Delta f = f_s/N$. The discrete wave energy spectrum $\sigma_\eta(f_i)$ is computed as:
$$\sigma_\eta^2(f_i) = S_\eta(i\Delta f)\Delta f \tag{6.4}$$

3. computing the discrete piston-displacement energy spectrum $\sigma_x(f_i)$:
$$\sigma_x^2(f_i) = \sigma_\eta^2(f_i)[K_f(k_i h)]^{-2}\ ,\ i = 1..N \tag{6.5}$$

where the Biesel transfer function K_f is a function of the discrete wave number $k_i = k(f_i)$;

4. calculating the N complex Fourier coefficients $C = A + iB$ by picking a random phase φ_f between 0 and 2π, for all frequencies smaller than the Nyquist frequency $f_N = f_s/2$:

$$\begin{aligned} A_i &= \cos(\varphi(f_i))\sqrt{\sigma_x^2(f_i)}/\sqrt{2} \\ B_i &= \sin(\varphi(f_i))\sqrt{\sigma_x^2(f_i)}/\sqrt{2} \end{aligned} \tag{6.6}$$

The N Fourier coefficients are mirrored to the Nyquist frequency f_N in order to obtain a hermitian Fourier Transform, i.e.:

$$C_{N+i} = C_{N-i+1}^* \ , \ i = 1..N \tag{6.7}$$

where * denotes the complex conjugate;

5. applying the inverse Fourier transform and calculating the time series of the control signal $X(t)$ for the piston. The real part of the inverse Fourier transform is the time series, the imaginary part is zero because the Fourier transform is Hermitian.

In the same way as for regular wave generation, the piston displacements $X(t)$ are converted into piston control velocities, which are stored in the file *piston_ left.vel*.

6.2.2 Active wave absorption

Methods for wave generation with active wave absorption were originally developed for physical wave flume experiments (Frigaard and Christensen, 1994). A velocity-meter based method for a VOF model with a Dirichlet wave boundary condition was presented by Troch and De Rouck (1999). A method based on the same principle is implemented here, with the necessary modifications to enable an operation in combination with a piston wavemaker.

Working principle

The principle of the active wave absorption is indicated in Fig. 6.1 and comprises two steps. Firstly, an on-line detection of velocities (u, w) at position (x_1,z_1) in front of the piston is performed. The measurement of these velocities enables the detection of the reflected wave field.

In a second step, a correction signal u_{ref}^* for the piston control velocity is computed, which will cancel out the reflected wave component propagating

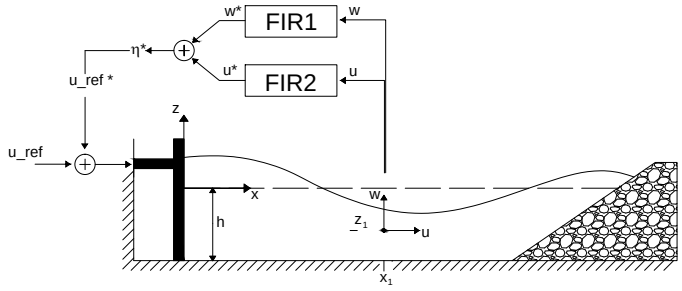


Figure 6.1: Definition sketch of active wave absorption system implemented in FLOW-3D.

towards the piston. The correction signal u^*_{ref} is determined from the two filtered velocity signals u^* and w^*. Digital FIR filters are used to compute a time-domain discrete convolution of the velocities (u, w) and the impulse response h^i (where i=u or w); e.g. for the u-velocity component:

$$u^*[n] = \sum_{j=0}^{J_f-1} h^u[j]u[n-j] \tag{6.8}$$

where J_f is the number of filter coefficients and $u^*[n] = u^*(n\Delta t_f)$ the filter output at time $t = n\Delta t_f$, with Δt_f the filter time interval. The impulse response $h^i(t)$ is determined from inverse Fourier transformation of the complex frequency response function $H^i(f)$, composed of a gain $C^i(f)$ and a phase $\varphi^i(f)$ (i=u or w):

$$Re(H^i(f)) = C^i(f)\cos(\varphi^i(f)) \tag{6.9a}$$

$$Im(H^i(f)) = C^i(f)\sin(\varphi^i(f)) \tag{6.9b}$$

The additional surface elevation $\eta^* = u^* + w^*$ to be generated in order to absorb the reflected wave is equal to $\eta_{-R} = a_R\cos(\omega t + \varphi_R + \pi)$, which is the reflected wave component at the generation boundary, in opposite phase. The derivation of the gain $C^i(f)$ and a phase $\varphi^i(f)$ for a system with a wave boundary condition is given by Troch and De Rouck (1999):

$$C^u(f) = \frac{-\sinh(kh)}{2\omega\ h(k(h+z_1))} \tag{6.10a}$$

$$C^w(f) = \frac{-\sinh(kh)}{2\omega\sinh(k(h+z_1))} \tag{6.10b}$$

$$\varphi^u(f) = \pi - kx_1 \tag{6.10c}$$

$$\varphi^w(f) = \pi - kx_1 + \pi/2 \tag{6.10d}$$

Applying the Biesel transfer function (eq. (6.1b)) and a phase shift φ_f converts η^* into a corrected piston displacement: $X^*(t) = K_f^{-1} a_R \cos(\omega t + \varphi_R + \varphi_f + \pi)$. The time derivation of $X^*(t)$ yields the required correction signal for the control velocity u^*_{ref}, and is simply obtained by amplifying the velocity signal by ω and applying a phase shift $\varphi(f) = \pi/2$. This leads to the following expressions for the gain and phase defining the frequency response eq. (6.9):

$$C^u(f) = \frac{-\sinh(kh)}{2K_f \cosh(k(h+z_1))} \tag{6.11a}$$

$$C^w(f) = \frac{-\sinh(kh)}{2K_f \sinh(k(h+z_1))} \tag{6.11b}$$

$$\varphi^u(f) = \pi - kx_1 + \varphi_f + \pi/2 \tag{6.11c}$$

$$\varphi^w(f) = \pi - kx_1 + \pi/2 + \varphi_f + \pi/2 \tag{6.11d}$$

Design of digital filters

Inverse Fourier transformation of the theoretical complex frequency response function, denoted as $H^i_{theo}(f)$, delivers the theoretical impulse response $h^i_{theo}(t)$. In practice, a finite number (J_f) of filter coefficients is used:

$$h^i_{theo}[j] = J_f^{-1} \sum_{k=0}^{J_f-1} H^i_{theo}[k] \exp(i2\pi kj J_f^{-1}) \tag{6.12}$$

where $h^i_{theo}[j] = h^i_{theo}(j\Delta t_f)$ is the value of the theoretical filter coefficient at discrete time steps $t = j\Delta t_f (j = 0..J_f - 1)$, where $\Delta t_f = T_{0f}/J_f$ is the filter time interval determined from the filter duration T_{0f}.

By using a finite number of filter coefficients in the Fourier transformation, the filter response may deviate from the theoretical response for input frequencies not coinciding with one of the discrete filter frequencies. As a result, it is recommended to verify the filter performance for intermediate frequencies which do not coincide with the discrete theoretical filter frequencies. This can be achieved by means of an oversampling technique and proves particularly useful for the absorption of reflected wave spectra.

It is noticed that the number of filter coefficients J_f is virtually unlimited in a numerical model, since the filter convolution is not exectuted in real

time as it is the case in a physical wave flume (see further). The parameters which define the filter design are:

- **the filter duration** T_{0f}, the most important parameter of the filter design. It defines the filter frequency interval Δf_f=T_{0f}^{-1}, which mainly determines the accuracy of the filter operation. In between discrete values of Δf_f, the realized filter response might deviate from the theoretical gain and phase. In case of regular waves with period T, it is possible to achieve good performance when Δf_f is a multiple of T^{-1}, even with a limited frequency resolution. In case of irregular wave generation, the frequency resolution should be chosen as high as possible, yet in accordance with the length of the wave flume. Practically, the filter should have reached a 'steady state' when the reflected waves reach the wave piston, yielding the following condition:

$$T_{0f} < 2L_{struct}/C \tag{6.13}$$

  where L_{struct} is the distance between the piston and the intersection of SWL and the tested structure, and C the wave celerity.

- for a given filter duration, the **number of filter coefficients** J_f defines the filter time interval $\Delta t_f = T_{0f}/J_f$ or the execution rate of the convolution eq. (6.8). Operation in a physical wave flume requires a real-time response of the piston to the reflected waves. This constrains the number of filter coefficients, since the calculation time of the convolution increases with J_f. Such a restriction does not exist in a numerical model, where the time step is advanced after the completion of the active absorption procedure, including the computation of the convolution. This allows an arbitrary definition of the filter time step, a property that later on will turn out useful in the numerical simulations. The input control and the corrected control velocities are prescribed at equal time intervals Δt_f, as in the physical wave flume. A different execution rate of piston movements (f_s) and filter operation (f_{sf}=Δt_f^{-1}) is possible, but has not been considered in this study.

- the **cut-off frequencies** f_{LC} and f_{HC} define the frequency interval in which the realized frequency response of the filter should match the theoretical response as closely as possible. This interval corresponds to the interval wherein all the energy of the wave spectrum is concentrated.

- the **position** (x_1,z_1) of the point where velocities are detected. By changing x_1, a phase shift in the filter operation is applied. This can be useful to avoid high-amplitude filter components at the initial and final instant of the filter duration, which deteriorates the filter performance. The value of z_1 has little effect on the filter performance. Considering

however that the active absorption method is based on the linear wave theory, a sufficient distance with SWL should be respected ($> h/3$), in order to minimize the performance reduction under nonlinear waves. The latter is due to the deviation from the linear velocity profile, which is largest near SWL. Hereafter, z_1 is always chosen equal to $-0.4h$.

The frequency response (eqs. (6.11)) shows one singularity in the gain $C^w(f)$ at $f = 0$ Hz. For high frequencies, zero gain is prescribed to avoid quick transitions in the phase shift. In addition, a cosine taper is applied to the theoretical gain $C^i_{theo}(f)$, in order to obtain a gradual transition between zero gain and the theoretical value on both sides of the cut-off limits:

$$C^i_{real}(f) = 0.5C^i_{theo}(f_{LC})\left[1+\cos\left(\frac{\pi f}{f_{LC}}+\pi\right)\right] \;;\; f \in [0; f_{LC}] \tag{6.14a}$$

$$C^i_{real}(f) = 0.5C^i_{theo}(f_{HC})\left[1+\cos\left(\frac{\pi(f-f_{HC})}{5\Delta f_f}\right)\right] \;;\; f \in [f_{HC}; f_{HC}+5\Delta f_f] \tag{6.14b}$$

Moreover, a tapering of the filter coefficients h^i can be applied to get a more stable digital filter. Existing LabView® code of the active absorption system operated in the wave flume of Ghent University has been adapted using the theoretical filter responses given by eqs. (6.11). In summary, the several steps in the design of the digital filter include:

1. selecting the filter duration T_{0f}, which should be chosen as large as possible to obtain the highest accuracy (certainly in case of irregular waves). An upper bound can be estimated from eq. (6.13);

2. selecting J_f, which determines the filter time interval and filter sample frequency f_{sf};

3. selecting f_{LC} and f_{HC} according to the generated wave period. In case of regular waves, a smaller frequency interval can be chosen, which simplifies the filter design. Care should be taken however that possible higher-order components in the reflected wave field arising from nonlinear interactions can be properly neutralized by the piston. It is recommended to take f_{HC} not smaller than $2T^{-1}$. In case of irregular waves, the interval $f_p/3 < f < 3f_p$ is generally respected. Choosing a value of f_{LC} slightly larger than the lower bound ($f_p/3$) can improve the filter performance, although the margin to adjust f_{LC} is limited due to the shape of the wave spectrum, which generally shows a steeper slope in the lower frequency range ($f < f_p$);

4. adjusting x_1 in order to achieve small (or zero) values of the filter coefficients at the initial and final instant of the filter duration. This can significantly improve the filter performance.

The control parameters defining the active wave absorption system are included in Table 6.1. They are implemented in the FLOW-3D customizable subroutine *mvbvel_usr.f*.

Table 6.1: Control parameters for active wave absorption.

parameter	**control option**
$paddle\_left$ (0 or 1)	(de)activate piston
awa (0 or 1)	(de)activate active wave absorption
xpistonL [m]	position of the right face of the piston
$dx1$ [m]	distance between right piston face and measurement location
zpos [m]	distance between SWL and vertical position of measurement location
dpaddle [m]	water depth near piston
$f_s = f_{sf}$ [Hz]	sample frequency of piston velocities and filter execution rate

6.2.3 Wave boundary conditions

Linear and nonlinear wave boundary conditions are available in the standard version of FLOW-3D. Nonlinear waves can be generated with a wave boundary based on Fenton's Fourier Series method (Fenton, 1988; Rienecker and Fenton, 1981). In order to prevent the aforementioned problem of fluid accumulation through the wave boundary, the code provides a default option that eliminates the net volume influx through the wave boundary (IRMFLUX=1). Tests have shown however that the efficiency of this remedy is questionable, particularly for highly-nonlinear waves.

Other wave boundary conditions that come standard with FLOW-3D enable the generation of cnoidal, solitary wave and random waves. Standard wave spectra included in the random wave generator are the Pierson-Moskowitz and JONSWAP spectrum. Additionally, it is also possible to employ a user-defined wave energy spectum. For further details on the implemented wave boundary conditions, the interested reader is referred to the FLOW-3D user manual (Flow Science, Inc., 2011).

6.2.4 Considerations with linear generation methods

Nonlinear regular waves propagating with constant form in intermediate or shallow water can be decomposed into free first harmonics and bound higher harmonics. A linear wave generation method, either a piston wavemaker or a wave boundary condition, does not include the natural bound higher harmonics. This leads to the release of parasitic higher harmonics which will

propagate as free wave components. The superposition of the free and bound higher harmonics leads to a spatial modulation of the wave amplitude. In a first estimate of the resulting surface elevation (Madsen and Sørensen, 1993), the second harmonic amplitude varies between 0 and 2 times the bound wave amplitude with a repetition or beat length L_B:

$$L_B = \frac{2\pi}{k_2 - 2k_1} \tag{6.15}$$

where k_1 and k_2 are the wave numbers corresponding to the free first and second bound harmonic, respectively. It is noticed that in this simplified description, subsequent sub- and superharmonic wave-wave interactions are neglected, as well as near-resonant triad interactions.

In the following, it will be verified whether and to which extent the piston wave generation is affected by harmonic generation.

6.3 Validation : 2D wave propagation over a horizontal bed

The propagation of 2D progressive nonbreaking waves over a horizontal bed is studied in detail in this section. Results obtained with the piston wavemaker will be compared with the nonlinear wave boundary and a theoretical solution based on Fenton's Fourier series theory. In a first step, a grid convergence study will be performed with the simplest spatial and time discretization options. Next, further testing of more advanced numerical options will be carried out. The goals of this study are to validate the operation of the piston wavemaker, to determine the numerical settings affecting the solution and to derive an optimal set of numerical parameters.

6.3.1 Test setup

The following study is limited to progressive waves. Reflection is prevented by constructing a wave flume with sufficient length, in agreement with the given test duration. Fig. 6.2 shows a definition sketch of the test setup. The initial piston position x_0 and flume length x_R are specified in Table 6.2. In case a wave boundary condition is used, the boundary coincides with $x = 0$. All other mesh boundaries are modeled as a free-slip condition.

Uniform cell dimensions are used in the entire study, using one single mesh block. Local mesh refinement by creating multiple mesh blocks is not considered. In this way, interpolation errors associated with block boundaries are avoided. The choice of uniform grid cells moreover delivers the highest possible accuracy. This higher accuracy originates from the

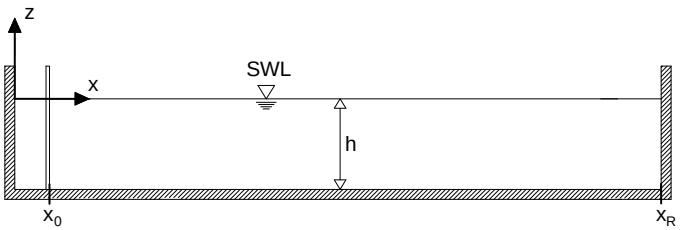


Figure 6.2: Definition sketch of the computational domain with indication of reference system.

way the governing equations are discretized (see Chapter 5), since in the evaluation of the change between quantities on either side of a cell, higher order terms cancel by symmetry in uniform cells (Flow Science, Inc., 2011).

The moving piston is represented by a solid obstacle. Care must be taken that the edges of the box-shaped piston in initial position coincide with the cell edges. Otherwise, the FAVOR obstacle representation, which is limited by the mesh resolution, introduces discretization errors in the contact area between the bed and the piston, allowing fluid to 'leak' into the dry area behind the piston. Tests show that a minimum obstacle thickness of 2 cells is required, again to avoid leakage of fluid caused by discretization errors. Fig. 6.3 illustrates the initial position of the piston in the mesh. A tapering of the piston movements is applied at the start and end of the test. In this way, surface elevations gradually build up or decrease, minimizing possible disturbances caused by excessive fluid acceleration.

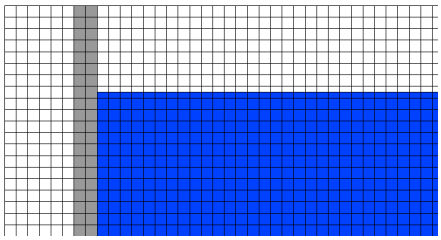


Figure 6.3: Initial position of the piston in the mesh.

Wave conditions are adopted from GWK tests, which are used in the validation study in Chapter 7. Four tests cases are selected (Table 6.2), in such way that practically the entire range of wave conditions is covered. Only regular waves are simulated at this stage, aiming to compare the results of surface elevations and velocity profiles with the linear and nonlinear wave theory.

Table 6.2: Definition of 2D wave propagation test cases.

case	h [m]	T [s]	H_0 [m]	L^1[m]	kh^1[-]	x_0[m]	x_R[m]	duration [s]
1	4.5	4.0	0.25	21.6	1.31	1.6	541.0	100
2	4.5	4.0	1.00	21.6	1.31	1.6	541.0	100
3	4.5	8.0	0.25	50.6	0.56	1.6	950.0	150
4	4.5	8.0	1.00	50.6	0.56	1.6	950.0	150

[1] according to linear wave theory

Test wave conditions are shown together with the limits of linear and nonlinear wave theories in Fig. 6.4. Theoretically, case 1 and 3 can be described by second-order Stokes wave theory, showing a mildly nonlinear character, close to linear waves. Waves in case 2 can be classified in between second- and third-order Stokes wave theory. Waves in case 4 exhibit the most pronounced nonlinear character, and are to be described by the fifth-order Stokes theory.

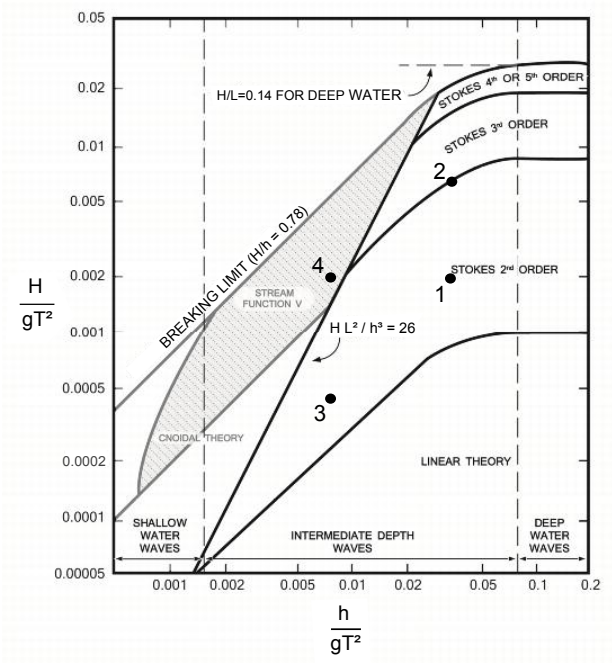


Figure 6.4: Location of wave conditions case 1 - 4 (Table 6.2) in diagram of wave theories (after Le Méhauté, 1976).

6.3.2 Basic grid convergence study

In the basic study, the first-order upwind momentum advection scheme will be employed. In this case, truncation errors are proportional to:

- the first power in time increment dt
- the first power in space increments dx, dy, dz if ALPHA $\neq 0$

Notwithstanding the first-order scheme generally shows good properties in terms of robustness and stability, it possibly produces larger numerical dissipation. In the grid convergence study, the consistency of the solution is verified by gradually reducing the cell dimensions. The aim of the convergence study is thus to verify whether and to what extent the simulations are sensitive to numerical dissipation. This dissipation can become visible in terms of wave height reduction and phase lag.

In the basic approach, an explicit treatment of convective terms and viscous stresses is applied. This imposes restraints to the time step size dt. Therefore, the time step control algorithm will be used, which automatically adjusts the time step size within the stability limits specified in 5.4.6.

When using the piston wavemaker, the sampling frequency f_s of the piston control velocities should be chosen in accordance with the resulting time step. The impact of this parameter on the solution will be discussed hereafter. Unless otherwise stated, the piston sample frequency f_s employed in all test cases is 40 Hz.

Since the studied problem deals with progressive nonbreaking waves, the flow is assumed to be laminar. All computations are performed with the split Lagrangian VOF advection method. The standard donor-acceptor algorithm produces very similar results in terms of free-surface motion, however also a considerably larger convective volume error. A list of relevant numerical parameter settings of the basic convergence study is given in Table 6.3. Other parameters not included in this table are set to default values. The solver executable *hydr3d* version 10.0.3 is used in all tests.

Case 1

Convergence is tested using 4 different cell sizes dx ranging between 0.4 and 0.05 m. Fig. 6.5 shows the results of surface elevations η and depth-profile of horizontal u-velocities under the passage of a wave crest and trough. Numerical results are plotted together with the theoretical solution obtained with Fenton's Fourier series method (Fenton, 2012).

Fig. 6.5(a,b) show the results obtained with the piston wavemaker at locations x=21.5 m ($\approx 1L$ from the piston) and x=146.5 m ($\approx 7L$ from the piston), respectively. The surface elevations correspond with an instant near the start of the wave train, after the wave generation is fully

Table 6.3: Parameter settings in basic convergence study.

Option	Setting	Parameter
fluid	fresh water (20° C), incompressible	ICMPRS=0
viscosity	Newtonian fluid	IFVISC=1
turbulence	laminar calculation	IFVIS=0
pressure solver	GMRES	IGMRES=1
	standard convergence criterion	EPSADJ=1
momentum advection	explicit	IMPADV=0
	first-order, upwind	IORDER=1, ALPHA=1
viscous stress	explicit	IMPVIS=0
VOF advection	split Lagrangian method	IFVOF=6
F-packing	default	CFPK=1
time step control	automatic (stability and convergence)	AUTOT=1
maximum dt	default	DTMAX=10^{10} s

established. In order to enhance the comparison of surface elevations, a time synchronization of wave crests at t/T=0.5 is applied. The small phase lag between results is a consequence of a different time step size associated with different cell sizes.

Surface elevations and velocity profiles show very limited differences between runs with cell size dx 0.4 and 0.05 m. They are in almost perfect agreement with the theoretical solution. The comparison between locations at different distance from the piston points to spatially stable results.

Fig. 6.5(c,d) show the results obtained with the nonlinear wave boundary generation based on Fenton's Fourier series method. Again, good correspondence with the theoretical surface elevation and velocity profiles is obtained. However, slightly larger deviations from the theoretical solutions are noticed at further distance from the piston, as seen in Fig. 6.5(d).

The stability of surface elevations and velocities in time is further explored in Fig. 6.6, showing velocity profiles under the passage of a wave crest and trough at the beginning and end of the wave train, for cell size dx=0.05 m. Fig. 6.6(a,b) at respectively closer and larger distance from the piston, show that the surface elevations and velocities are stable, both in time and space. Fig. 6.6(c,d) show the same for the nonlinear wave boundary. At larger distance from the wave boundary, larger variations in surface elevation and velocity profile are noticed, albeit still limited.

The dependency of the wave height on the numerical mesh is verified by calculating characteristic wave heights $H_{m,123}$ and $H_{m,456}$, obtained by averaging the mean wave height H_m in 3 locations x_i, indicated in Table 6.4. The obtained averages are specified in Table 6.5, expressed as the relative difference Δ with H_0 (= 0.25 m), the value specified at the generation

boundary:

$$\Delta = \frac{H_m - H_0}{H_0} \tag{6.16}$$

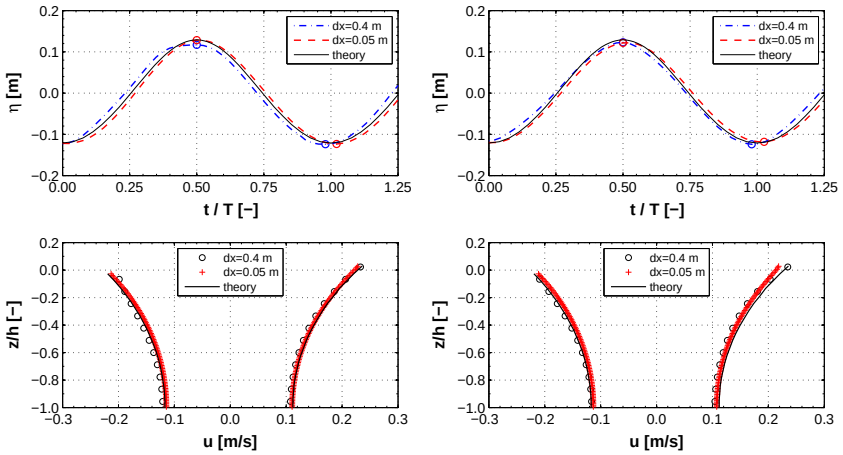
Limited differences in averaged H_m values are noticed between different cell sizes in Table 6.5. A more pronounced difference appears between H_{123} and H_{456}, showing a slight reduction of wave height with increasing distance from the piston or wave boundary. In all cases however, the obtained wave height is within a range of 6% of the target value (Table 6.5). It seems that the impact of the first-order momentum advection, and the expected numerical diffusion resulting from this approximation is limited. Even with the largest cell size tested (dx=0.4 m), a relatively accurate wave generation is achieved. However, tests with cell sizes larger than 0.4 m show increasing instabilities in surface elevations and velocities. A value of 0.4 m seems to be a practical upper limit for the cell size dx in this case.

Table 6.4: Positions x_i (in m) used in spatial averaging of H_m, case 1 and 2.

x_1	x_2	x_3	x_4	x_5	x_6
21.5	27	32.5	135.5	141	146.5

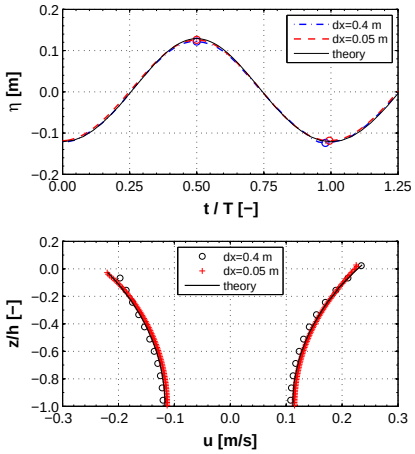
Table 6.5: Difference Δ [%] between target wave height H_0 (=0.25 m) and spatially-averaged H_m, case 1.

	piston		wave boundary	
dx [m]	Δ_{123}	Δ_{456}	Δ_{123}	Δ_{465}
0.40	-2.00	-3.68	-2.52	-4.50
0.20	-1.54	-4.79	-2.01	-4.79
0.10	-2.34	-5.35	-1.85	-4.63
0.05	-0.05	-3.12	-2.31	-4.69

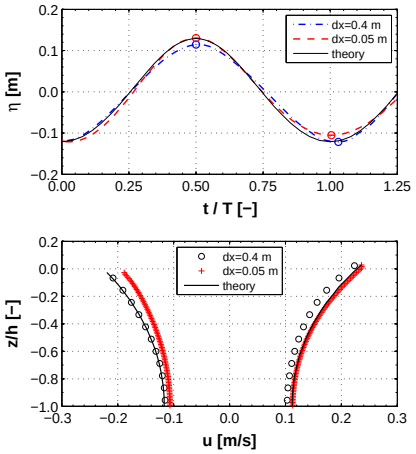


(a) Piston, x=21.5 m

(b) Piston, x=146.5 m

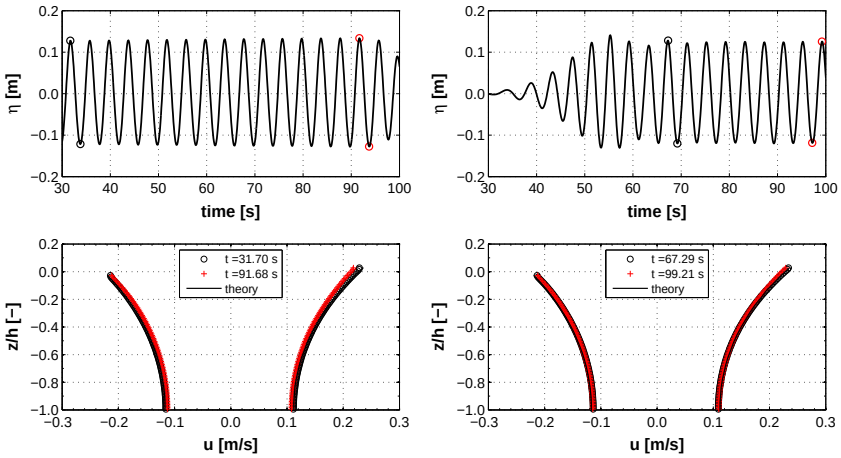


(c) Wave boundary, x=21.5 m

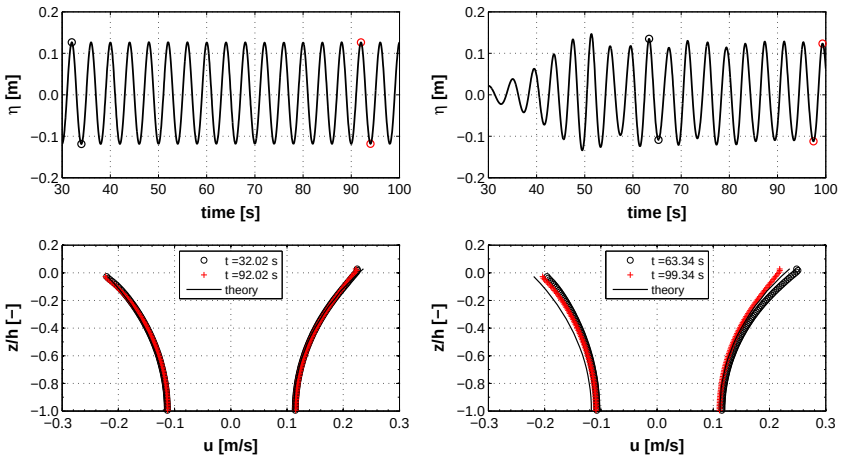


(d) Wave boundary, x=146.5 m

Figure 6.5: Close-up on surface elevation η and horizontal velocity $u(z)$ at an instant near the start of the wave train. Numerical results generated with the piston wavemaker and nonlinear wave boundary, at two different x-positions and for two different cell sizes dx, case 1.



(a) Piston, x=21.5 m (b) Piston, x=146.5 m



(c) Wave boundary, x=21.5 m (d) Wave boundary, x=146.5 m

Figure 6.6: Verification of time-dependent stability of surface elevation η and horizontal velocity $u(z)$ generated by the piston wavemaker and nonlinear wave boundary, at two different x-positions, cell size dx=0.05 m, case 1.

Case 2

The spatial variability of surface elevations is checked first, as the nonlinear waves (Fig. 6.4) might be affected by harmonic generation (see 6.2.4). Free-surface elevations resulting from both wave generation methods are plotted at several distances in between x=65 m ($\approx 3L$ from the wave boundary) and x=80 m in Fig. 6.7. The distance covered (15 m) is approximately equal to the theoretical beat length predicted by eq. (6.15), using the linear wave theory to compute the wave numbers. Given the very limited spatial variation of η in Fig. 6.7, it can be concluded that no significant harmonic generation occurs. Generally, good agreement is observed between the theoretical and numerical surface elevations.

The grid-dependency of the numerical solution is shown in Fig. 6.8, providing a detailed view of the surface elevation over one wave period, together with the corresponding horizontal velocities under the passage of a wave crest and trough. The surface elevations correspond to an instant near the start of the wave train, after waves have fully developed. As in the previous case, wave crests are synchronized at t/T=0.5 in this comparison.

Fig. 6.8(a,b) show the results obtained with the piston wavemaker at locations x=21.5 m ($\approx 1L$ from the piston) and x=146.5 m ($\approx 7L$ from the piston), respectively. The accuracy of the solution clearly improves by increasing the mesh resolution. For cell size dx of 0.05 m, numerical results agree almost perfectly with the theoretical solution. The solution obtained with dx=0.4 m clearly suffers from numerical dissipation, to an increasing degree with increasing distance from the piston. Fig. 6.8(c,d) show similar results, yet obtained with the nonlinear wave boundary. Slightly larger deviations from the theoretical solution are noticed here.

In order to evaluate the time-dependent stability of the solution, a large part of the total time series is shown in Fig. 6.9, together with depth-profiles of u-velocities corresponding with a crest and trough, at two instants near the beginning and the end of the wave train. The solution obtained with the piston wavemaker (dx=0.05 m) is shown in Fig. 6.9(a,b) at location x=21.5 m, and x=146.5 m, respectively. Clearly, a stable wave train is obtained, both in time and space. Fig. 6.9(c,d) show similar results, yet obtained with the nonlinear wave boundary. Again, as in case 1, the solution seems to show a larger variability in time than with to the piston wavemaker.

Spatially-averaged values of H_m are given in Table 6.6, expressed as the relative difference with H_0 (=1.0 m). As expected, wave heights tend to the target value when the mesh resolution increases. Simulations with cell sizes 0.40 and 0.20 m are clearly affected by numerical dissipation. In addition, a significant difference between H_{123} and H_{456} appears, showing a clear reduction of wave height with increasing distance from the piston or wave boundary. With a sufficient mesh resolution however, wave heights

generated by the piston are within ± 2 % of the target value.

Table 6.6: Difference Δ [%] between target wave height H_0 (=1.0 m) and spatially-averaged H_m, case 2.

	piston		**wave boundary**	
***dx* [m]**	$\mathbf{\Delta_{123}}$	$\mathbf{\Delta_{456}}$	$\mathbf{\Delta_{123}}$	$\mathbf{\Delta_{465}}$
0.40[1]	-5.91	-14.60	-5.85	-14.67
0.20[1]	+0.03	-5.34	+0.35	-8.84
0.10[1]	+1.95	-3.54	-2.37	-6.13
0.05[2]	+2.08	-1.65	-3.19	-4.49

[1] obtained with f_s=40 Hz,
[2] obtained with f_s=100 Hz

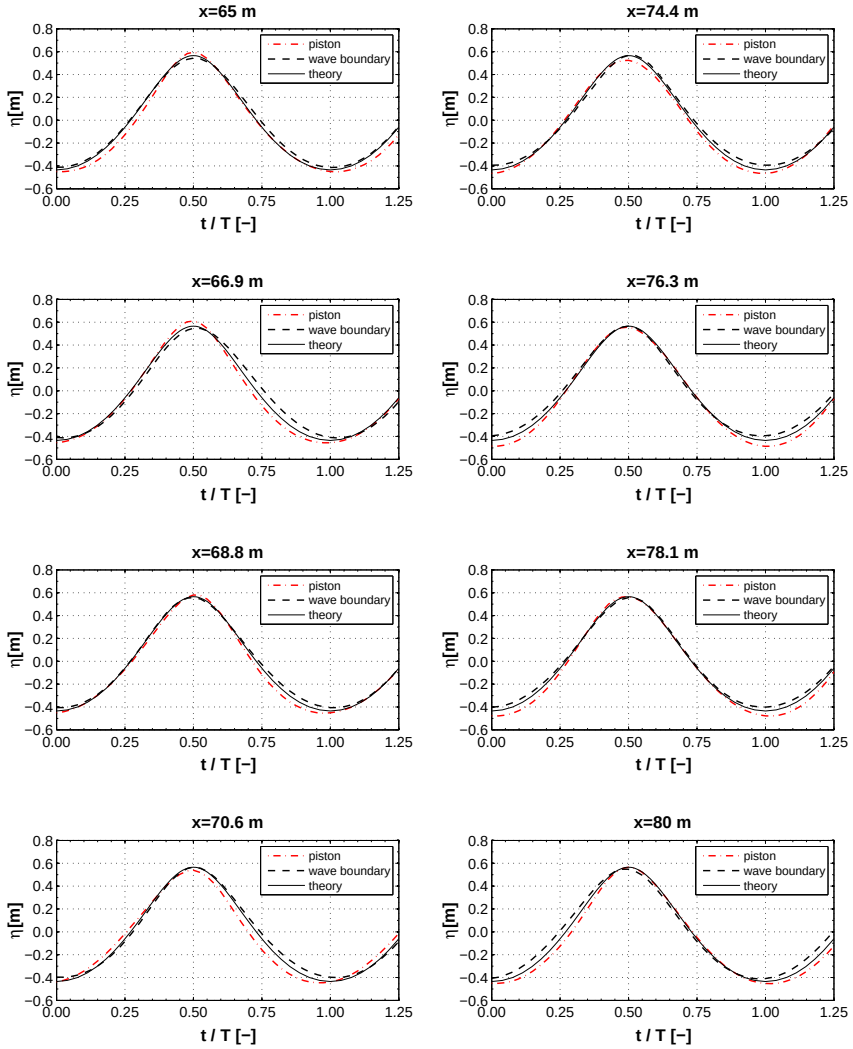
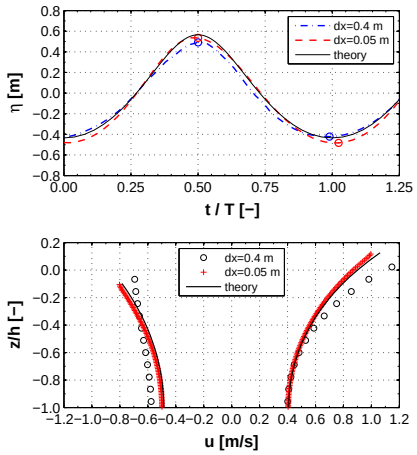
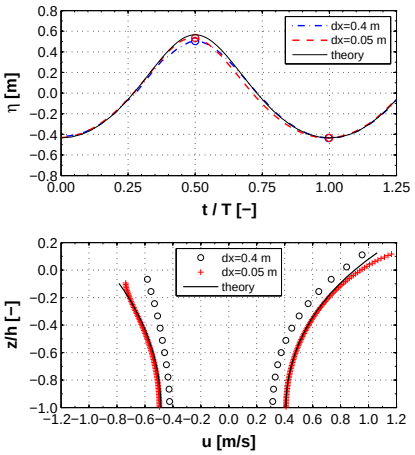


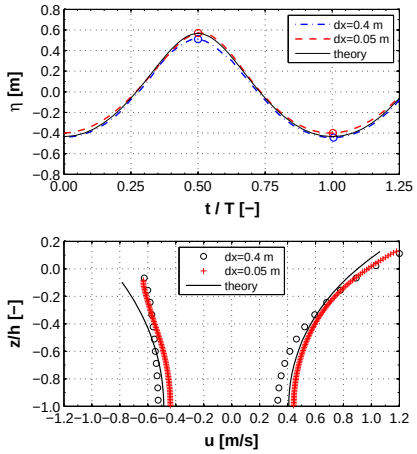
Figure 6.7: Variation of surface elevation η in between x=65 and 80 m, corresponding with an instant near the end of the wave train, for test case 2. Numerical surface elevations obtained with piston wavemaker and nonlinear wave boundary condition ($dx = 0.05$ m), theoretical values with Fenton's Fourier series method.



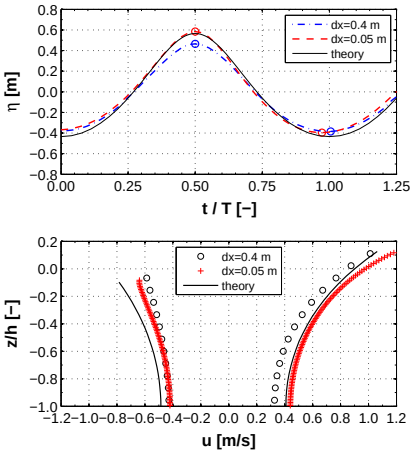
(a) Piston, x=21.5 m



(b) Piston, x=146.5 m

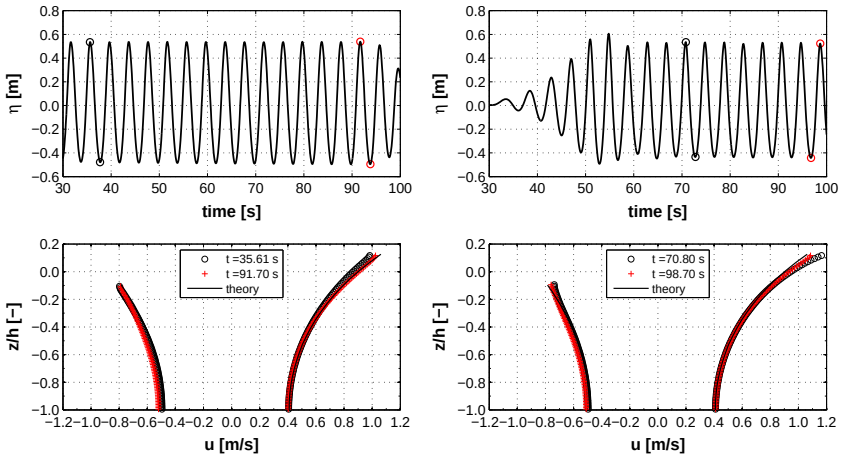


(c) Wave boundary, x=21.5 m



(d) Wave boundary, x=146.5 m

Figure 6.8: Close-up on surface elevation η and horizontal velocity $u(z)$ at an instant near the start of wave train. Numerical results generated with the piston wavemaker and nonlinear wave boundary, at two different x-positions and for two different cell sizes dx, case 2.



(a) Piston, x=21.5 m (b) Piston, x=146.5 m

(c) Wave boundary, x=21.5 m (d) Wave boundary, x=146.5 m

Figure 6.9: Verification of time-dependent stability of surface elevation η and horizontal velocity $u(z)$ generated by the piston wavemaker and nonlinear wave boundary, at two different x-positions, cell size dx=0.05 m, case 2.

Case 3

Fig. 6.10 shows the surface elevation and velocity profiles in detail. No significant differences between both generation methods are observed. It is noticed that the surface profile obtained with the wave boundary condition in Fig. 6.10(d) shows a more pronounced asymmetry with respect to the vertical plane. Nonetheless, surface elevations and velocity profiles generally agree very well with the theoretical solution. The comparison between results obtained with dx=0.4 and 0.1 m indicate that convergence is already obtained with the lowest mesh resolution.

Fig. 6.11 shows a large part of the entire time series where velocity profiles are taken at instants corresponding with the start and end of the wave train. It can be concluded that the results are stable in time, for both wave generation methods.

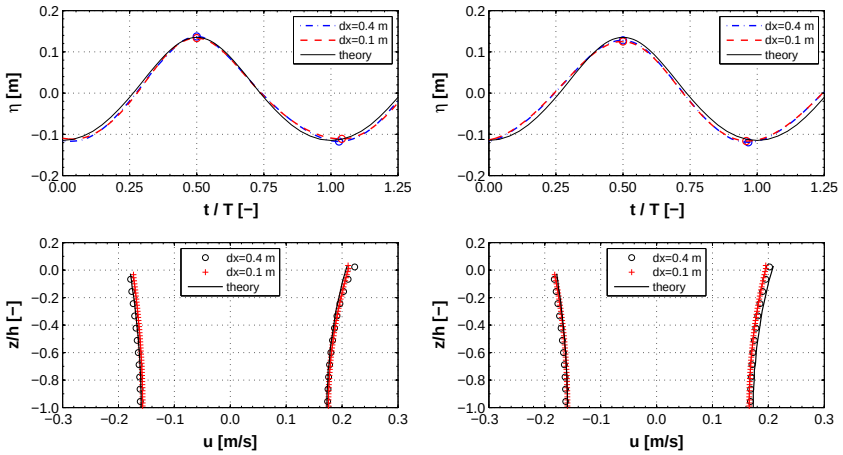
Analogous to the previous cases, wave heights $H_{m,123}$ and $H_{m,456}$ are obtained by averaging the mean wave height H_m in 3 locations x_i, indicated in Table 6.7. Table 6.8 shows the spatially-averaged wave heights, expressed as the relative difference with H_0(=0.25 m). It is noticed that a larger wave height is obtained with the largest cell size (dx=0.4 m), in contrast to the previous cases. If numerical dissipation were to have an effect, it should decrease with increasing mesh resolution. One would then rather expect the wave height to increase with increasing mesh resolution. Deviations from the target wave height are limited however, within ± 5 %.

Table 6.7: Positions x_i (in m) used in spatial averaging of H_m, case 3 and 4.

x_1	x_2	x_3	x_4	x_5	x_6
51.0	63.5	76.0	253.5	266.0	278.0

Table 6.8: Difference Δ [%] between target wave height H_0 (=0.25 m) and spatially-averaged H_m, case 3.

	piston		wave boundary	
dx [m]	Δ_{123}	Δ_{456}	Δ_{123}	Δ_{465}
0.40	+0.36	-0.81	+0.12	+0.55
0.20	-1.89	-3.52	-0.80	-0.69
0.10	-2.32	-3.68	-1.76	-2.48
0.05	-1.12	-4.04	-3.21	-4.44



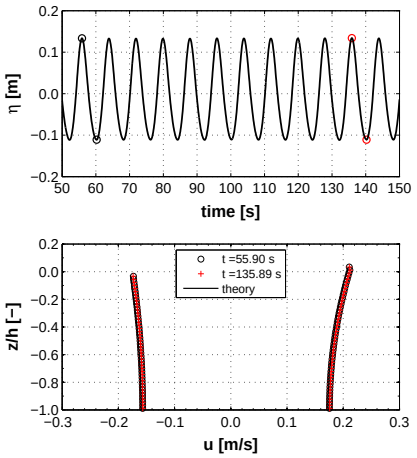
(a) Piston, x=51.0 m

(b) Piston, x=278.0 m

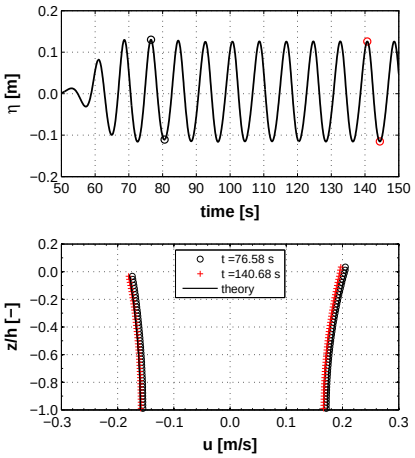
(c) Stokes V, x=51.0 m

(d) Stokes V, x=278.0 m

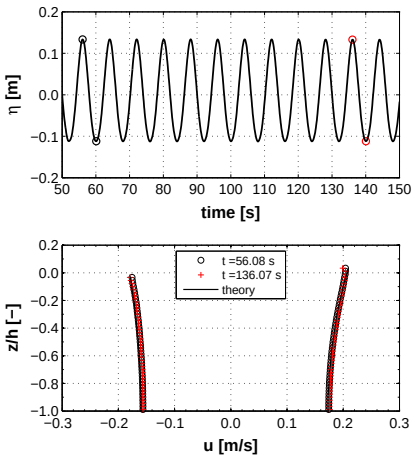
Figure 6.10: Close-up on surface elevation η and horizontal velocity $u(z)$ at an instant near the start of the wave train. Numerical results generated with the piston wavemaker and nonlinear wave boundary, at two different x-positions and for two different cell sizes dx, case 3.



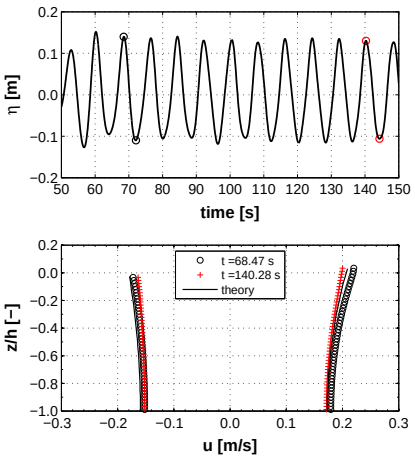
(a) Piston, x=51.0 m



(b) Piston, x=278.0 m



(c) Wave boundary, x=51.0 m



(d) Wave boundary, x=278.0 m

Figure 6.11: Verification of time-dependent stability of surface elevation η and horizontal velocity $u(z)$ generated by the piston wavemaker and nonlinear wave boundary, at two different positions x, cell size dx=0.10 m, case 3.

Case 4

Of all four test cases in Fig. 6.4, waves in case 4 exhibit the highest nonlinear character. The theoretical beat length according to eq. (6.15) amounts to 146 m. Fig. 6.12 shows the free-surface elevation at eight different locations in between one beat length. Computations with dx=0.1 m are shown, obtained with the piston wavemaker and the nonlinear wave boundary condition. The comparison between different locations reveals a clear variation of surface elevation for the linear piston wavemaker. At some locations, secondary peaks in the trough appear. The spatial η-variation caused by the release of higher harmonics is not observed with the nonlinear wave boundary.

The surface profile generated by the piston wavemaker at location x=150 m is very similar to x=296 m, an indication that the repetition of the spatial variation corresponds well with the theoretical beat length predicted by eq. (6.15). The harmonic generation under linear piston generation results in larger wave heights compared to the waves generated by the nonlinear wave boundary, as indicated in Table 6.9.

Fig. 6.13 shows the velocity profile at different locations in between x=150 m and x=241.2 m, obtained with the piston wavemaker. Notwithstanding the spatial variations in η, the wave train appears to be stable in time in each location. The velocity profiles corresponding with the passage of a crest and trough clearly deviate from the theoretical solution. The discrepancy between numerical and theoretical profiles of u-velocities increases when the local amplitude of the second harmonic reaches its maximum value, at approximately half the beat length. In Fig. 6.13(c), the sharp, peaked crest and flattened trough correspond with an increase in u-velocities near the free surface.

Table 6.9: Difference Δ [%] between target wave height H_0 (=1.0 m) and spatially-averaged H_m, case 4.

	piston		**wave boundary**	
$\boldsymbol{dx}$ **[m]**	$\boldsymbol{\Delta_{123}}$	$\boldsymbol{\Delta_{456}}$	$\boldsymbol{\Delta_{123}}$	$\boldsymbol{\Delta_{465}}$
0.20[1]	+15.63	+4.13	-1.72	-1.55
0.10[1]	+14.37	+3.41	-2.29	-1.60
0.05[2]	+11.67	+1.46	-2.60	-1.78

[1] obtained with f_s=40 Hz,
[2] obtained with f_s=100 Hz

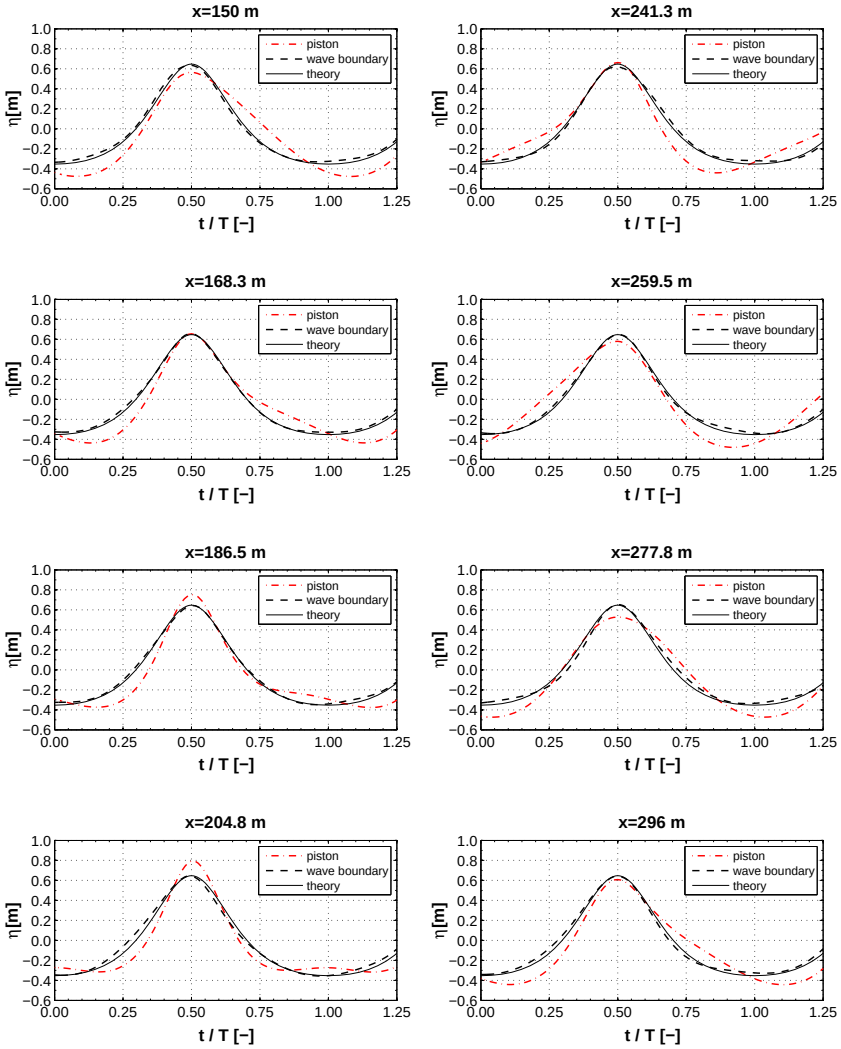
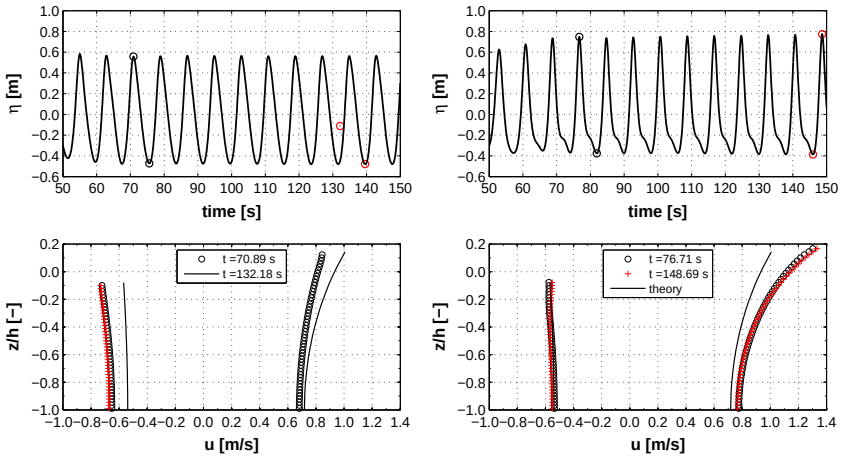
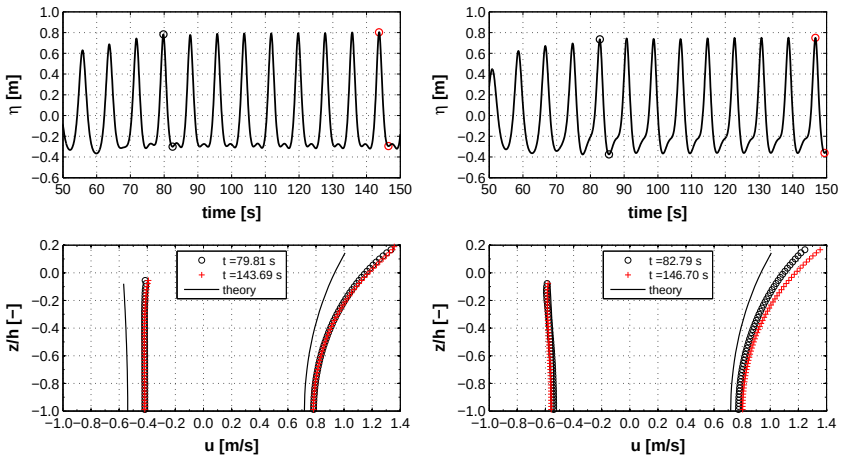


Figure 6.12: Variation of surface elevation η in between x=150 and 296 m, corresponding with an instant near the end of the wave train, for test case 4. Numerical surface elevations obtained with piston wavemaker and nonlinear wave boundary condition ($dx = 0.10$ m), theoretical values with Fenton's Fourier series method.



(a) x=150.0 m (b) x=186.5 m



(c) x=204.8 m (d) x=223.0 m

Figure 6.13: Close-up on surface elevation η and horizontal velocity $u(z)$ generated by the piston wavemaker, at different locations x in between approximately half the beat length, cell size dx=0.1 m, case 4.

6.3.3 Further testing of numerical options

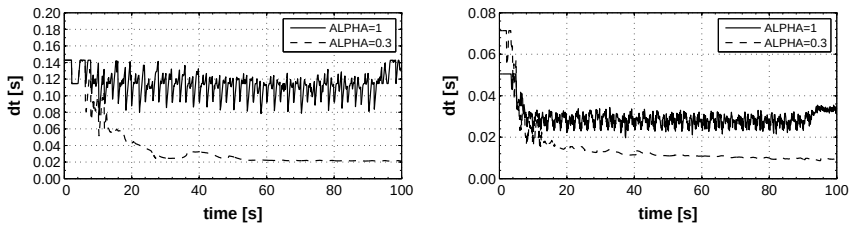
In the previous section, a first-order upwind discretization scheme for the momentum advection terms was used in all simulations. In the following it is tested whether a mixed first-order or second-order scheme yields better results in terms of computational accuracy and/or efficiency. A discussion on the time step control (section 5.4.6) is included in the following, since it appears that the time step evolution is closely connected with the spatial discretization of momentum advection.

Because of the larger wave height and absence of harmonic generation in case 2, this particular case will be used to illustrate the impact of the momentum discretization scheme and time step control on the solution. All results are obtained with piston wave generation, with f_s=40 Hz (unless otherwise stated).

Momentum advection approximations

A variation to the first-order upwind scheme consists of a mixture of upwind and centered differences (see 5.4.3) and is obtained by setting the weighting factor ALPHA to a value between 0 and 1. Liu and Lin (1997) suggest to take ALPHA in the range of 0.3 to 0.5. The impact of the weighting factor ALPHA is studied by comparing the standard upwind scheme (ALPHA=1) to a mixed scheme with ALPHA=0.3. Use is made of the automatic time step algorithm, determined by stability constraints and the number of pressure iterations (AUTOT=1).

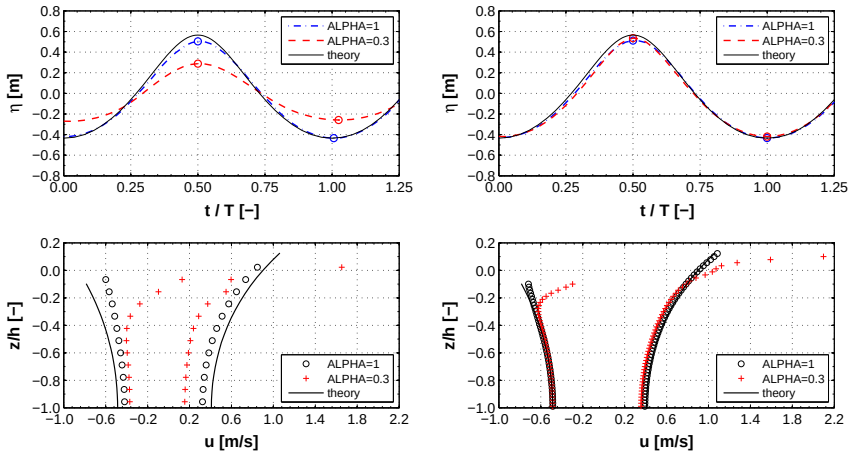
Fig. 6.14 shows the impact of ALPHA on the time step size. Clearly, the mixed scheme results in a reduction of the time step size. In both cases, the advection in the x-direction is the restraining limit to the time step size. It is noticed that the value of ALPHA is smaller than the default value (0.45) of the CFL stability limit (see section 5.4.6), hence eq. (5.58) yields the most stringent condition to the time step limit.



(a) dx=0.40 m (b) dx=0.10 m

Figure 6.14: Impact of ALPHA on time step size dt, case 2.

Surface elevations and corresponding velocity profiles under a passage of a crest and trough are shown in Fig. 6.15, at location x=146.5 m, taken at an instant corresponding to the start of the wave train. Comparison of results between different cell sizes in Fig. 6.15 show that the mixed scheme is more prone to numerical dissipation than the fully-upwind scheme (ALPHA=1). For both mesh resolutions, a disturbance of the velocity profile near the free surface is noticed with the mixed scheme. This could be the result of the significant reduction in time step size.



(a) dx=0.40 m (b) dx=0.10 m

Figure 6.15: Impact of ALPHA on surface elevation η and horizontal velocity $u(z)$, shown at x=146.5 m, at an instant near the start of the wave train, case 2.

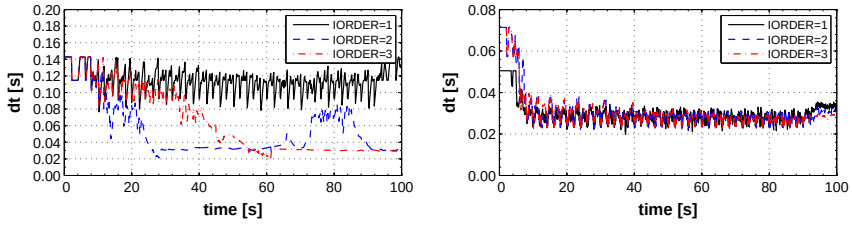
In the previous section, simulations with the first-order upwind scheme proved to yield accurate results with limited numerical dissipation, when the mesh resolution is sufficiently high. It was noticed that a higher mesh resolution is needed for the cases with larger wave height. In the following, it is verified whether a second-order advection scheme yields higher accuracy for lower mesh resolutions. Both the regular (IORDER=2) and monotonicity-preserving (IORDER=3) second-order scheme are used.

Fig. 6.16 shows the impact of the momentum discretization on the time step size. A standard upwind differencing (ALPHA=1) is used in the first-order scheme. Employing the automatic time step control with both second-order schemes results in a significant reduction of the time step size in case of a relatively coarse grid (Fig. 6.16(a)). On a finer grid, all time step sizes result to be very similar (Fig. 6.16(b)).

Contrary to what one would assume, the second-order schemes do not

improve accuracy on a relatively coarse grid (dx=0.40 m), when comparing to the first-order method. A decrease in wave height is noticed in Fig. 6.17. Velocity profiles are less stable, showing spurious deviations from the theoretical profile near the free surface in Fig. 6.17(a). This local, unphysical increase in velocity leads to a reduction of dt due to the application of the CFL stability limit, as observed in Fig. 6.16(a).

For a higher mesh resolution, differences between the different schemes diminish, see Fig. 6.17(b). It is remarked that in the latter case, resulting values of time step size for different momentum schemes are very similar, suggesting that the difference in dt could be a cause for the deviations observed with dx=0.4 m. This is further explored hereafter.

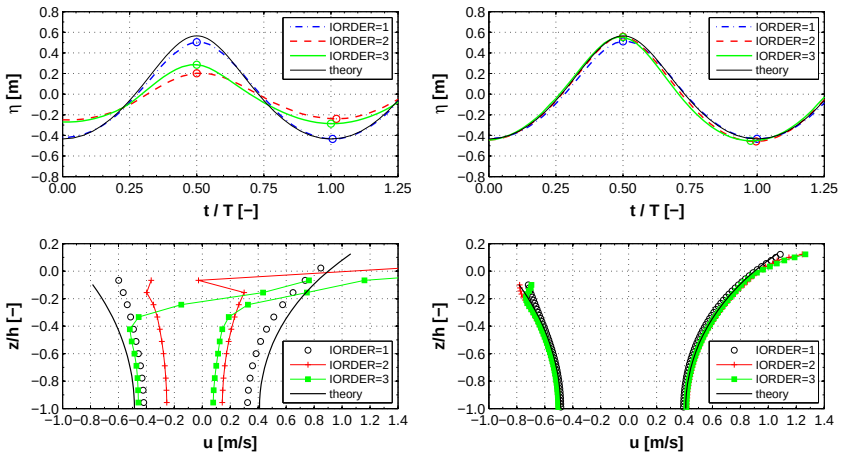


(a) dx=0.40 m (b) dx=0.10 m

Figure 6.16: Impact of IORDER on time step size dt, case 2.

Time step control

The previous simulations suggest that deviations in surface elevation and velocity profiles in the mixed and second-order schemes are associated with a reduction of dt, resulting from the stability limits applied by the automatic time step control. In the following, the impact of the time step control (variable or fixed time step size) and the magnitude of dt are tested with the first-order upwind scheme. In order to exclude the effect a variable tolerance in eq. (5.53), the automatic pressure convergence criterion is replaced by a fixed value of EPSI, to be specified in accordance with the maximum expected dt and below the limit predicted by eq. (5.53). The automatic time step control is set to AUTOT=2, leaving the resulting dt unaffected by the number of pressure iterations. In order to investigate the effect of the time step size, values for dt are estimated from Fig. 6.14. Upper bounds of 0.08 and 0.02 s are selected for dx respectively 0.4 and 0.10 m, close to the smallest time step size resulting from the application of stability constraints. Both values of dt are reduced by a factor 4 in order to check the effect of the magnitude of dt on the solution.

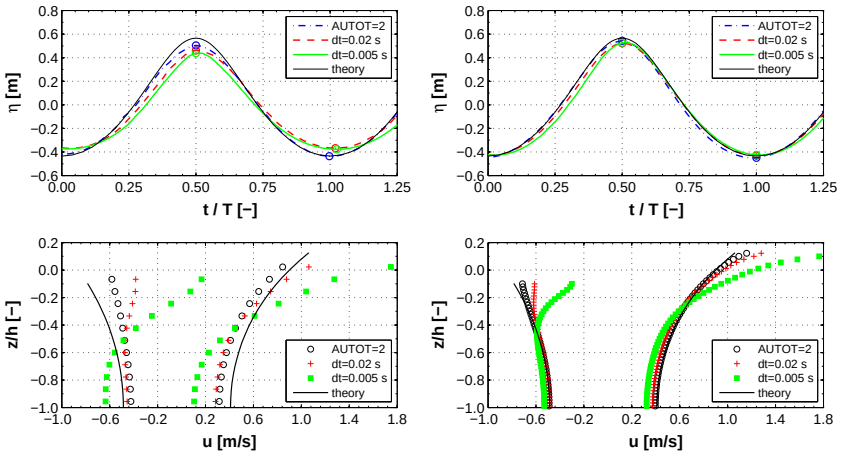


(a) dx=0.40 m (b) dx=0.10 m

Figure 6.17: Impact of IORDER on surface elevation η and horizontal velocity $u(z)$, shown at x=146.5 m, at an instant near the start of the wave train, case 2.

Fig. 6.18 shows the surface elevations and corresponding velocity profiles under the passage of a crest and trough at x=146.5 m, taken at an instant corresponding to the start of the wave train. The magnitude of dt clearly affects the velocity profiles, and to a lesser extent also the surface elevations. Results seem to deteriorate when dt is reduced, apparently to an increasing degree with increasing difference between the specified value of dt and the value resulting from the automatic time step control. This will have an implication on the operation of the piston wavemaker using the active wave absorption method, as will be further discussed in section 6.5.

In addition to the effect on the surface elevations and fluid velocities, the time step size also has an effect on the modeled phase. Tests with different values of dt show different values of phase lag, to an increasing degree with increasing distance from the piston. Reducing dt results in an improved accuracy of the modeled phase.



(a) dx=0.40 m, EPSI= $5 * 10^{-4}$ (b) dx=0.10 m, EPSI= $1 * 10^{-3}$

Figure 6.18: Impact of time step control and time step size dt on surface elevation η and horizontal velocity $u(z)$, shown at x=146.5 m, at an instant near the start of the wave train. Simulations case 2 with IORDER=1, ALPHA=1.

Selecting the piston sample frequency f_s

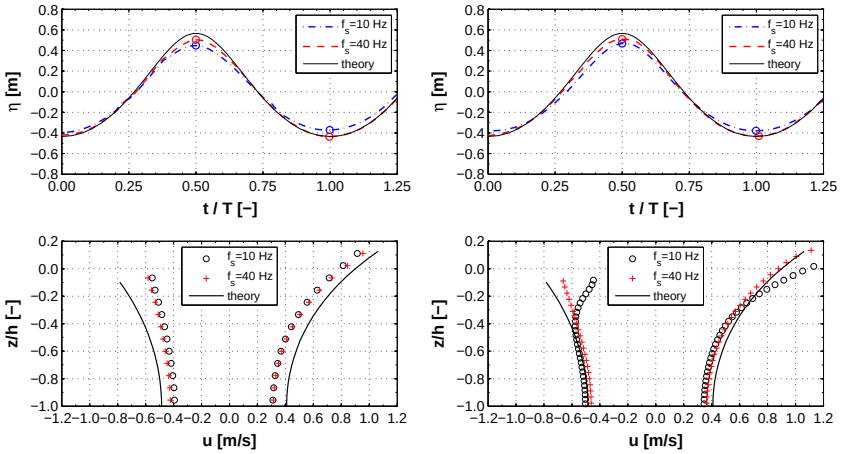
In this previous section, the rate f_s at which piston control velocities are specified was set to 40 Hz in all cases, except for the smallest cell size (dx=0.05 m). In fact, it is noticed that the piston time step $dt_p = f_s^{-1}$ should be chosen in accordance with the time step size dt resulting from the application of stability constraints. For instance, when dt_p is much larger than the realized time step, the input signal of the piston control velocities looks like a step function. Tests with different values of dt_p show that this can affect the surface elevation and particularly the u-velocities, which tend to show an increase (a *drift*) in a confined region near the free surface.

Table 6.10 contains time step sizes dt for different combinations of dx and dt_p, computed for test cases 1 and 2. The results in Table 6.10 indicate that the drift in u-velocity is connected with a relatively large ratio of and dt_p to dt. It is also noticed that the value of dt_p has a small impact on the resulting time step since the latter is being adapted by the automatic time step control algorithm, in order to properly resolve the movement of the piston. Using a smaller piston time step dt_p generally results in a slightly larger numerical time step dt.

Fig. 6.19 shows the impact of f_s on the surface elevations and velocity profiles for case 2, computed with different mesh resolutions. For cell sizes dx=0.2 and 0.1 m, the observed drift in u-velocity vanishes when dt_p is reduced. For the highest mesh resolution (dx=0.05 m) however, it seems that the drift is very limited, in spite of the large difference between dt_p and dt in case f_s=10 Hz. In Fig. 6.19, the deviations in the velocity profile reach till a depth of about $-0.4h$, which corresponds with the vertical position of the single-point velocity measurement in the active wave absorption procedure. If the disturbance grows beyond this point, it could affect the performance of the active wave absorption.

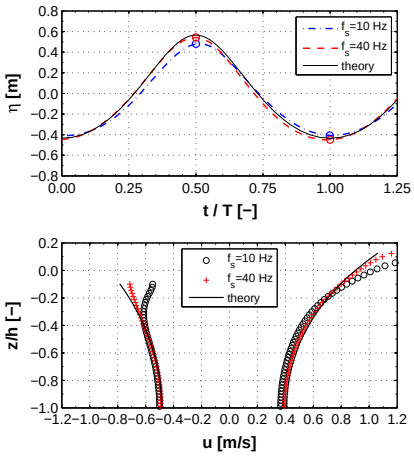
Table 6.10: Difference between dt_p and realized dt and indication of drift in $u(z)$, for case 1 and 2.

	case 1 (H_0 = 0.25 m)			**case 2 (H_0 = 1.0 m)**		
***dx* [m]**	***dt_p* [s]**	***dt* [s]**	**drift?**	***dt_p* [s]**	***dt* [s]**	**drift?**
0.4	0.10	0.101	no	0.10	0.079 - 0.101	no
0.4	0.025	0.101	no	0.025	0.078 - 0.101	no
0.2	0.10	0.06 - 0.071	no	0.10	0.028 - 0.045	**yes**
0.2	0.025	0.071	no	0.025	0.045 - 0.065	no
0.1	0.10	0.010 - 0.051	**yes**	0.10	0.015 - 0.025	**yes**
0.1	0.025	0.046 - 0.051	no	0.025	0.025 - 0.033	no
0.05	0.10	0.015 - 0.031	**yes**	0.10	0.010 - 0.015	**yes**
0.05	0.025	0.028 - 0.032	no	0.01	0.012 - 0.017	no

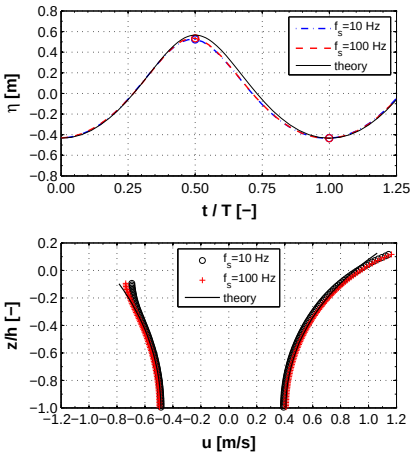


(a) dx=0.40 m

(b) dx=0.20 m



(c) dx=0.10 m



(d) dx=0.05 m

Figure 6.19: Impact of f_s on surface elevation η and horizontal velocity $u(z)$ shown at x=146.5 m, at an instant near the start of the wave train. Simulations case 2 for different cell sizes dx.

6.3.4 Conclusions

The observations drawn from the simulation of 2D progressive nonbreaking waves over a horizontal bed lead to the following conclusions and recommendations:

- The test cases show that the minimum mesh resolution required for an accurate solution of surface elevations and velocities is mainly determined by the wave height. A minimum ratio H/dx of 0.5 to 1 appears sufficient when the wave height is relatively small (case 1 and 3). For larger waves, H/dx needs to be raised (to about 20 for case 2 and 4). The wavelength has a smaller impact on the required mesh resolution, though a minimum ratio L/dx about 50 to 100 seems reasonable. Note that uniform cell sizes dx were used in all simulations. A further optimization with nonuniform meshes is not undertaken in the present study;

- An accurate modeling of progressive waves can be achieved with the first-order upwind scheme, which has been shown to yield the most stable solutions. A comparison with higher-order momentum advection schemes does not indicate a larger numerical dissipation associated with the first-order method, even on relatively low mesh resolutions;

- The automatic time step control (AUTOT=1/2) provides the most efficient solution. A user-specified constant time step much below the implemented stability constraints can lead to deviations in the velocity profile near the free surface, and to a minor extent in the surface elevation. The critical minimum time step associated with the occurrence of a drift in near-surface velocities depends on the mesh resolution;

- A maximum piston time step dt_p should be selected, which should not largely exceed dt resulting from the automatic time step control. Otherwise, control velocities behave like a step function which can affect the results. The sensitivity of the solution to the value of dt_p however is observed to be mesh-dependent, diminishing with increasing mesh resolution. No specification of a maximum allowable time step size is needed when the piston operates in single-generation mode (i.e. without active absorption).

6.4 Validation : long-duration wave test

2D wave propagation over a horizontal bed was modeled in the previous section, where zero wave reflection was obtained by constructing a long wave flume and a relatively short test duration (about 20 wave periods). In most wave flume studies however, the simulation of long-duration test series is required, e.g. in tests with irregular waves or tests where certain aspects of wave-structure interaction take a considerable number of wave cycles in order to fully establish; e.g. the wave-induced set-up of MWL in permeable structures.

In the following section, the stability in time of a long series of piston-generated waves (about 100 wave periods) is investigated. Tests in section 6.3.3 revealed that results can become unstable when the time step size is reduced. This is of particular importance, since a reduction of dt is most likely to occur when waves interact with a structure. The following tests aim to the investigate the effect of a variable time step size on the long-term stability of the generated wave train, using the different momentum discretization schemes available in FLOW-3D.

6.4.1 Test setup

The simulation of an undisturbed progressive wave field with long duration in combination with a wave flume of limited length requires an adequate technique for absorbing the waves. The standard nonreflecting outflow boundary condition available in the code (a Sommerfeld-type condition) does not require additional space in the computational domain and therefore is the most efficient method in terms of computational cost. However, tests with this boundary condition show unacceptable degrees of reflection in most cases. Therefore, use is made of a passive absorption technique, a so-called *sponge* layer. The sponge is the numerical counterpart of an absorbing beach applied in a physical wave flume, where flow motions are gradually dampened out over a gentle slope, reducing the reflection as much as possible.

Fig. 6.20 presents the setup of the numerical flume, with indication of the piston and the sponge. In FLOW-3D, the sponge has been implemented in the customizable subroutine *qsadd.f*, by gradually damping the individual fluid velocity components (u, v, w) over a distance L_{sponge} toward zero values at $x = x_R$. The mathematical description of the sponge takes the form of a

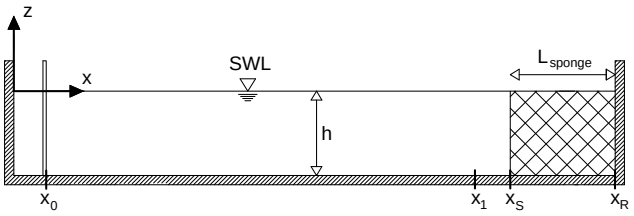


Figure 6.20: Definition sketch of the computational domain for long-duration wave test.

power function[1], which e.g. for the u-velocity reads as:

$$u^*(x_i) = u(x_S)\left[1 - \left(\frac{x_i - x_R + L_{sponge}}{L_{sponge}}\right)^{n_s}\right] \tag{6.17}$$

where $u^*(x_i)$ and $u(x_s)$ are the velocity components at position x_i and x_s, respectively ($x_S \leq x_i \leq x_R$). The power n_s in eq. (6.17) determines the shape of the sponge function. Tests with the sponge function eq. (6.17) indicate a recommended value of n_s between 3 and 7. Additionally, the absorption performance improves with increasing L_{sponge}. Tests with varying sponge lengths show that a minimum sponge length of about one wave length is required to achieve acceptable absorption (i.e. CR<0.20). A sponge length of about $3L$ provides optimum results in terms of absorption capacity and computational cost. For smaller sponge lengths, a smaller value of n_s in eq. (6.17) yields better absorption. For larger values of L_{sponge} ($> 2L$), the value of n_s has a limited impact on the absorption performance.

As in section 6.3.3, case 2 will be employed as a reference case in the following tests. Waves are generated using the piston wavemaker in single-generation mode. All simulations are performed with dx=0.1 m and f_s=40 or 100 Hz (depending on dx and dt). A fixed sponge length of 64.8 m ($\approx 3L$) is used, with n_s=3. A wave flume with length 108 m ($\approx 5L$) is constructed between the initial piston position x_0 and the starting point x_S of the sponge.

6.4.2 First-order upwind momentum advection

Tests are run with a constant time step size of 0.3 and 0.15 s, set by AUTOT=0. A value of 0.3 s closely corresponds to the smallest time step size resulting from the automatic time step control (AUTOT=2). Fig. 6.21

[1]Other formulations than eq. (6.17) exist (e.g. elliptic or cosine sponge functions) but are not tested here.

shows the resulting surface elevations and velocity profiles at a location $x_1 = 100$ m, near the end of the wave flume. The reduced time step size clearly leads to a dissipation of incident wave height, shown in detail in Fig. 6.21(b). The dissipation increases with time but apparently stabilizes after about 200 s. The reduction of wave height corresponds with a drift of u-velocity in Fig. 6.21(d): in the lower half of the cross section, a decrease of horizontal velocities occurs, whereas an increase is noticed in the upper half.

It is known that a reduction in time step size increases the numerical dissipation with the first-order upwind scheme, since the diffusive error ϵ_{diff} for e.g. a flow with velocity u in the x-direction is proportional to:

$$\epsilon_{diff} \propto \left(1 - u\frac{dt}{dx}\right) \tag{6.18}$$

Tests with increasing mesh resolution do not yield significant improvement in the velocity profile. It is thus unlikely that the increase in velocity observed in Fig. 6.21(d) is caused by numerical diffusion, since a diffusive effect should decrease noticeably with increasing mesh resolution. It can only be concluded that a plausible argument is still lacking at the moment, explaining for the erroneous behavior of the first-order upwind scheme in case the time step size is reduced considerably below the stability limit resulting from the automatic time step control.

6.4.3 Second-order momentum advection

A drawback of the second-order momentum advection is a reduced stability compared to the upwind scheme, manifesting as spurious velocities near the free surface. The local increase of velocity causes a reduction of dt, due to the application of the CFL stability constraint. The following tests with the second-order schemes are therefore run with AUTOT=2 and the specification of DTMAX.

Fig. 6.22 shows the resulting surface elevations and velocity profiles at x_1. A detailed view on the surface elevations shows that a reduced DTMAX leads to slightly larger wave heights. However, both simulations show excellent stability in time. The velocity profiles in Fig. 6.22(c,d) remain stable and agree very well with the theory, except for the spurious velocities near the free surface.

6.4.4 Second-order monotonicity-preserving momentum advection

Similar tests are performed as with the second-order scheme. Fig. 6.23 shows that this algorithm does not yield stable results, in spite of the methodology

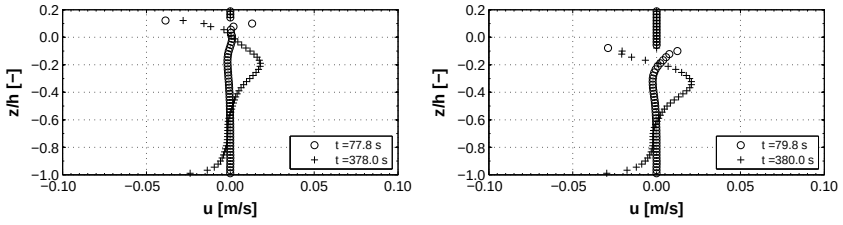
applied in the algorithm which aims for enhanced stability.

6.4.5 Impact of spatial discretization

In an attempt to further explore the instable behavior of the first-order upwind scheme, the influence of the grid is investigated. Therefore, the simulation with dt=0.015 s, which showed to become unstable, is repeated with the number of cells ny in transversal direction increased to 2 and 4.

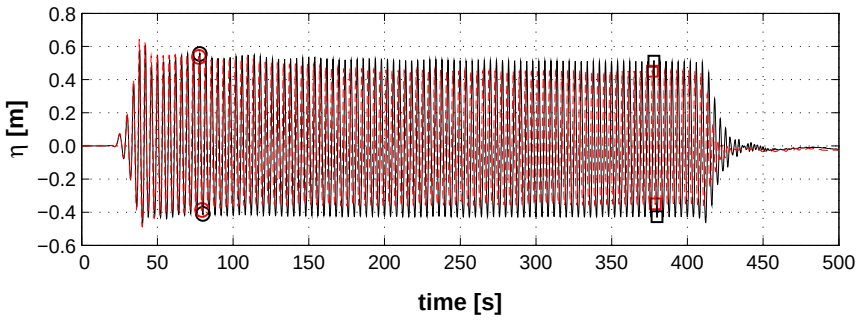
Fig. 6.24 shows the resulting surface elevations and velocity profiles of both simulations. Small differences are observed between different values of ny. However, both simulations with ny equal to 2 and 4 do not exhibit the damping of wave height and drift in fluid velocities, associated with the time step reduction with ny=1 (Fig. 6.21). This suggests that other factors than numerical diffusion are causing the instability under a reduction of time step size.

The slight deviations in the velocity profiles in Fig. 6.24(c,d) are most probably caused by disturbances in the transversal direction, as shown in Fig. 6.25. Theoretically, the velocity component v should be zero over the entire fluid depth for a purely 2D flow. Disturbances in the v-profile are small, limited to about 5% of the u-profile, but seem to increase with time.

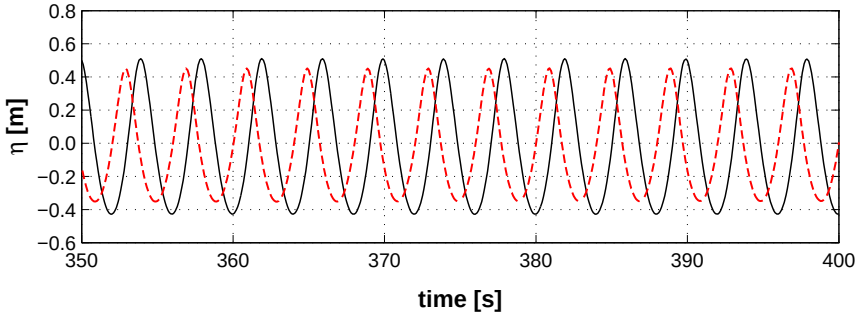


(a) crest (b) trough

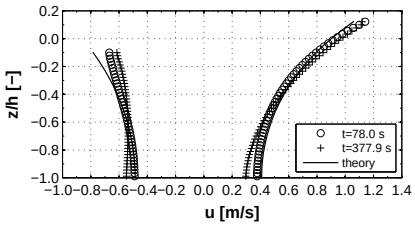
Figure 6.25: Cross-direction v-velocity profile, shown at $x = x_1$ and half the cross section width (y=0.2 m). Velocity profiles corresponding with the passage of (a) crest and (b) trough, corresponding with the start and end of the wave train. Simulations case 2 with dx=0.1 m, ny=4, IORDER=1, ALPHA=1, AUTOT=2, DTMAX=0.015 s, EPSI= $1 * 10^{-3}$.



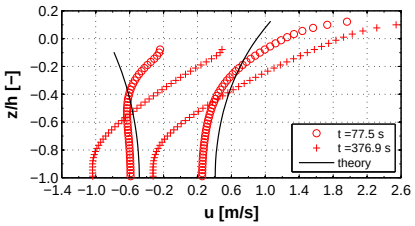
(a)



(b)

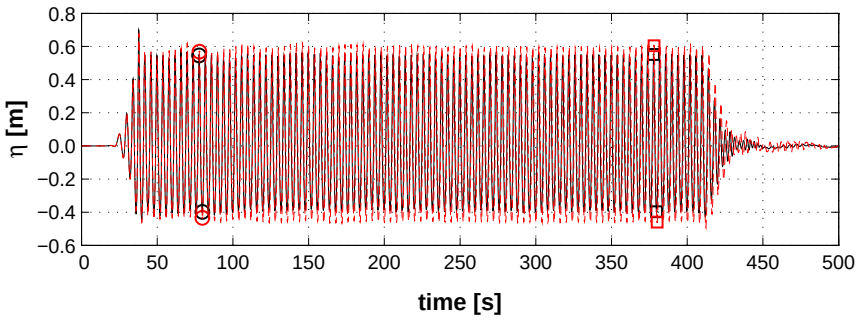


(c) dt=0.03 s

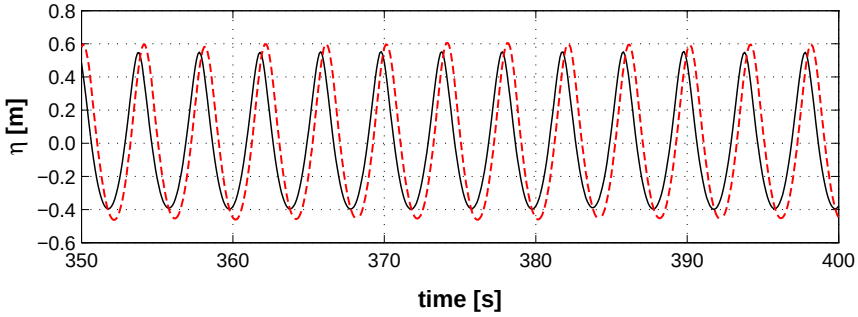


(d) dt=0.015 s

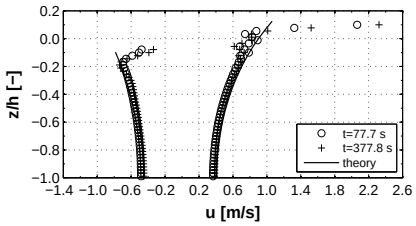
Figure 6.21: Impact of dt on surface elevations and horizontal velocities, shown at $x = x_1$. Total time series (a) and close-up (b) of surface elevations for dt=0.03 s (black solid) and dt=0.015 s (red dashed). Velocity profiles corresponding with the start and end of the wave train for (c) dt=0.03 s and (d) dt=0.015 s. Simulations case 2 with dx=0.1 m, IORDER=1, ALPHA=1, AUTOT=0, EPSI= $1 * 10^{-3}$.



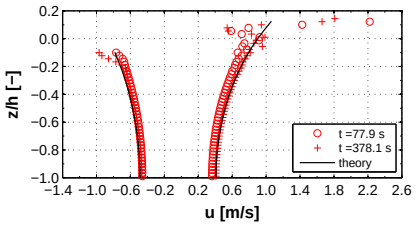
(a)



(b)

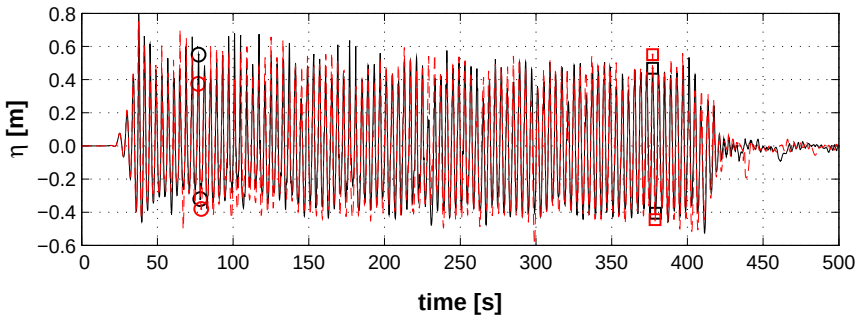


(c) DTMAX=0.015 s (f_s=40 Hz)

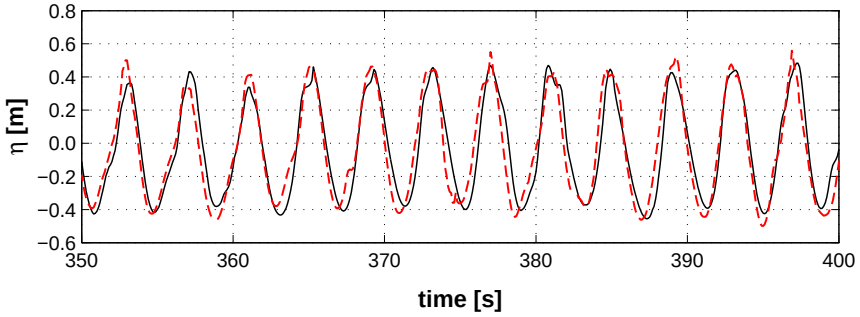


(d) DTMAX=0.005 s (f_s=100 Hz)

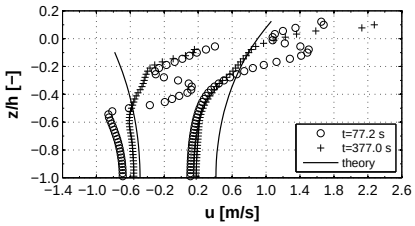
Figure 6.22: Impact of dt on surface elevations and horizontal velocities, shown at $x = x_1$. Surface elevations in panel (a) and (b) shown for DTMAX=0.015 s (black solid) and DTMAX=0.005 s (red dashed). Velocity profiles corresponding with the start and end of the wave train for (c) DTMAX=0.015 s and (d) DTMAX=0.005 s. Simulations case 2 with dx=0.1 m, IORDER=2, AUTOT=2, EPSI= $1 * 10^{-3}$.



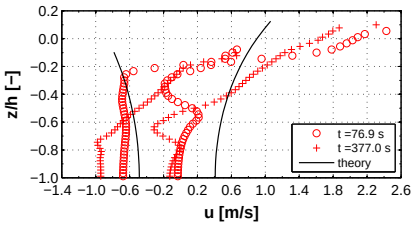
(a)



(b)

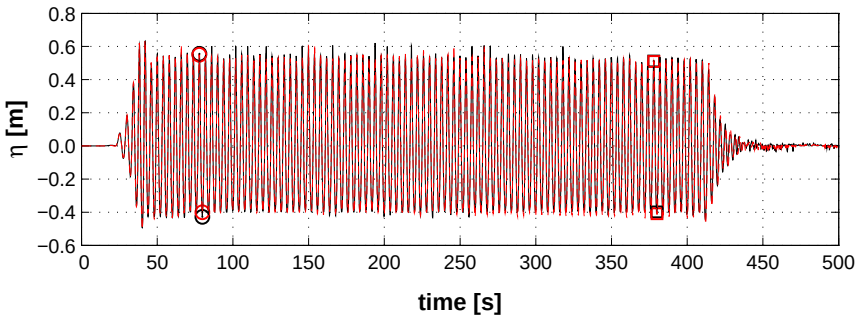


(c) DTMAX=0.015 s

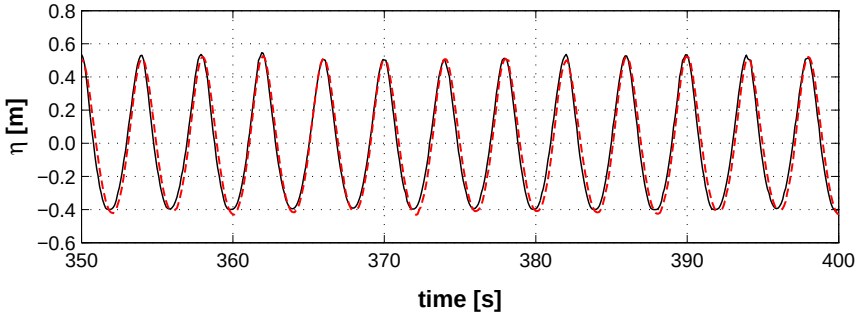


(d) DTMAX=0.005 s

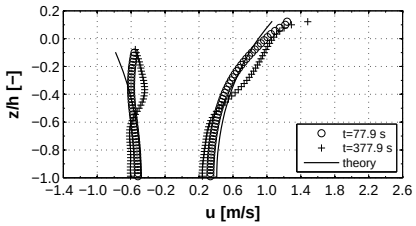
Figure 6.23: Impact of dt on surface elevations and horizontal velocities, shown at $x = x_1$. Surface elevations in panel (a) and (b) shown for DTMAX=0.015 s (black solid) and DTMAX=0.005 s (red dashed). Velocity profiles corresponding with the start and end of the wave train for (c) DTMAX=0.015 s and (d) DTMAX=0.005 s. Simulations case 2 with dx=0.1 m, IORDER=3, AUTOT=2, EPSI= $1 * 10^{-3}$.



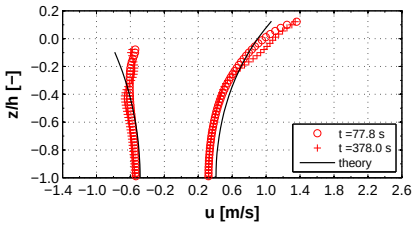
(a)



(b)



(c) ny=2



(d) ny=4

Figure 6.24: Impact of ny on surface elevations and horizontal velocities, shown at $x = x_1$. Surface elevations in panel (a) and (b) shown for ny=2 (black solid) and ny=4 (red dashed). Velocity profiles corresponding with the start and end of the wave train for (c) ny=2 and (d) ny=4. Simulations case 2 with dx=0.1 m, IORDER=1, ALPHA=1, AUTOT=2, DTMAX=0.015 s, EPSI= $1 * 10^{-3}$.

6.4.6 Conclusions

Testing the stability of long-duration wave tests with the first-order upwind momentum advection shows a considerable decrease of wave height and nonphysical drift in fluid velocities, in case dt decreases considerably, below the stability limits specified in section 5.4.6. A plausible reason for this deficiency is lacking, and simulations with multiple grid cells in the cross direction suggest that mere numerical diffusion is unlikely to be a cause, since the drift is not observed in those cases.

Tests with the second-order scheme do not exhibit the instabilities observed with the first-order upwind scheme. Generally, excellent stability of surface elevations and velocity profiles is obtained, except for the occurrence of spurious velocities near the free surface. The second-order monotonicity-preserving momentum advection scheme yields the poorest results of all three schemes, showing large instabilities in both surface elevation and fluid velocity.

6.5 Validation : active wave absorption

6.5.1 Test setup

The operation of the piston wavemaker has been verified for purely progressive waves, i.e. without compensation for reflected waves. In this section, the performance of the active wave absorption system is tested by operating the piston in pure absorption mode. This corresponds to a case where one would expect maximum (100%) reflection from a structure placed within the wave flume.

A piston wavemaker is positioned near the right boundary of the computational domain, with initial position of the left piston face at $x = x_0$, as depicted in Fig. 6.26. The same test cases as in section 6.3 are used, except case 4. In the latter case, the spatial variability of velocity profiles due to harmonic generation complicates the proper evaluation of the absorption performance. The piston near the right boundary operates in simple-generation mode, i.e. without active wave absorption. The input control velocity for the generating piston is denoted by $u_{ref,2}$.

Near the left boundary, a piston working in absorbing mode generates the wave that absorbs the incident wave. The initial position of the right face of the absorbing piston equals to $x = x_L$. The piston velocity $u^*_{ref,1}$ is calculated from the superposition of the filtered velocity signals at location (x_1, z_1). The parameters defining the FIR filters are given in Table 6.11. By due selection of the filter duration, care is taken that the discrete filter frequencies coincide with the ground frequency of the generated wave train (i.e. 0.25 and 0.125 Hz). Tests are carried out with different piston sample

frequencies f_s. The filter frequency f_{sf} is always taken equal to f_s by adjusting the number of filter coefficients.

The length of the wave flume between the piston faces is taken as approximately 5 wavelengths, specified in Table 6.12. With the given flume length, the time for the waves to reach the absorbing piston is approximately 20 s for cases 1 and 2, compared to 40 s in case 3. Given the filter duration of 40 s, this implies that the filter is not yet fully operational when the waves reach the piston in cases 1 and 2. However, the test results show that this has no significant effect on the performance of the absorbing piston.

Other numerical parameters defining the model setup are adopted from the basic test setup defined in section 6.3.2. One important exception concerns the specification of the maximum allowable time step size DTMAX. In the case of simple wave generation, no restrictions to DTMAX were needed. This no longer holds when using the active wave absorption, since it is then necessary to employ a maximum value of DTMAX equal to the piston time step size $dt_p = f_{sf}^{-1}$. The latter is due to the execution of the filter convolution is required at regular times steps f_{sf}^{-1}, which is not guaranteed when dt exceeds $dt_p = f_{sf}^{-1}$. Tests show that the computation becomes unstable when dt exceeds dt_p, leading to excessive fluid motion or even the abortion of the simulation due to a continued time step reduction caused by the piston motion.

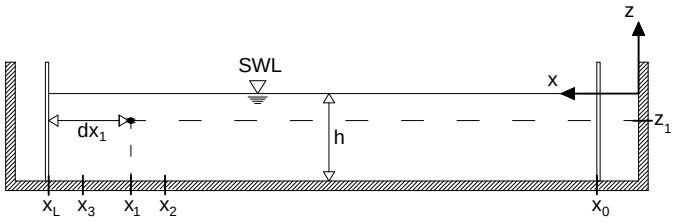


Figure 6.26: Set up of the computational domain for test with absorbing piston.

Table 6.11: Absorbing-piston test: specifications of filter design.

	dx_1 [m]	z_1 [m]	T_{0f} [s]	f_{LC}[Hz]	f_{HC} [Hz]
case 1,2	30	-1.8	40	0.10	0.75
case 3,4	65	-1.8	40	0.045	0.375

The performance of the absorbing piston is verified in 2 ways. The first method computes the amount of wave reflection in front of the piston, which should tend to zero when the left piston is perfectly absorbing the

Table 6.12: Absorbing piston test: specifications of computational domain.

	L[m]	x_0 [m]	x_L[m]	x_1 [m]	x_2 [m]	x_3 [m]
case 1,2	21.58	1.6	109.6	79.6	77.44	82.84
case 3,4	50.62	2.0	255.2	190.2	185.14	197.79

incident waves. The reflection coefficient $CR(f)$ is computed with the 3-gauge-method of Mansard and Funke (1980). Wave gauge locations x_i (i = 1..3) are indicated in Table 6.12.

In a second method, the error associated with the absorption of the incident waves is computed as:

$$\epsilon_{abs} = \frac{\overline{u}_{ref,2} - \overline{u}^*_{ref,1}}{\overline{u}_{ref,2}} \tag{6.19}$$

where $\overline{u}^*_{ref,1}$ and $\overline{u}_{ref,2}$ are the mean amplitudes of the control velocities of the left and right piston, respectively. Under perfect absorption, the corrected signal $\overline{u}^*_{ref,1}$ equals the input control signal $\overline{u}_{ref,2}$, and ϵ_{abs} reduces to zero.

6.5.2 Case 1

Simulations are performed with different cell size dx, piston time step dt_p and maximum time step DTMAX, summarized in Table 6.13. The results show that satisfactory absorption of the incident waves (e.g. $CR(f)< 0.10$ or $|\epsilon_{abs}| < 0.05$) can be achieved already with the largest cell size dx=0.4 m, in case an appropriate piston time step is selected (dt_p=0.025 s). Very little difference is observed between cell sizes 0.1 and 0.05 m. The solution shown in Fig. 6.27 for dx=0.05 m demonstrates that waves are almost perfectly absorbed by the piston.

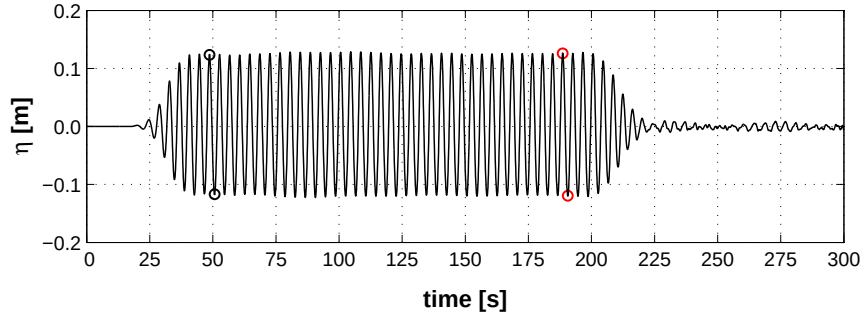
As it is noticed in the results in Table 6.13, the choice of dt_p and DTMAX can have a considerable impact on the performance of the absorbing piston. The simulations of progressive waves in section 6.3.3 already showed that reducing dt relatively far below the stability limit size can lead to a drift in the velocity profile near the free surface, when using the first-order upwind scheme. The restriction DTMAX=dt_p may have a similar effect. The incipient value of dt_p causing the drift will depend on the cell size dx. In Table 6.13, it is indicated whether or not the reduction of dt leads to a drift in the velocity profile. Except for the largest cell size dx=0.4 m, this explains why reducing dt_p below a certain limit does not improve the absorption performance.

In addition to the value of dt_p, the specification of DTMAX can have a considerable impact on the obtained wave height, provided that the value is

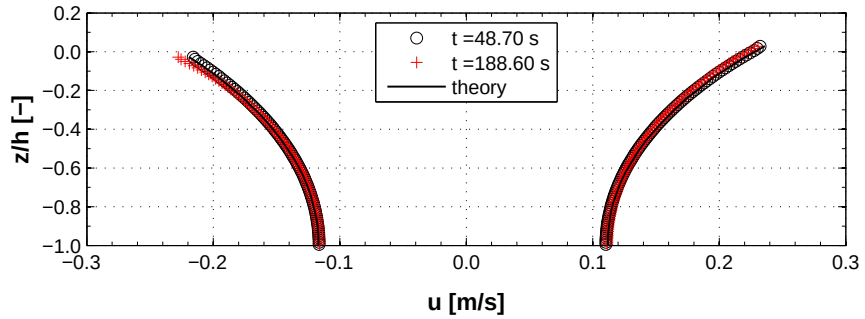
larger than the critical value which leads to a drift in the velocity profile. In the tests, DTMAX was reduced to half the piston time step size dt_p, leading to a significant increase in $H_{m,inc}$ and a reduction of reflection for dx=0.1 and 0.05 m. It is noticed that a smaller time step size yields an improved accuracy of the modeled phase, which can explain the improved absorption performance in those cases.

Table 6.13: Performance of the absorbing piston, for test case 1.

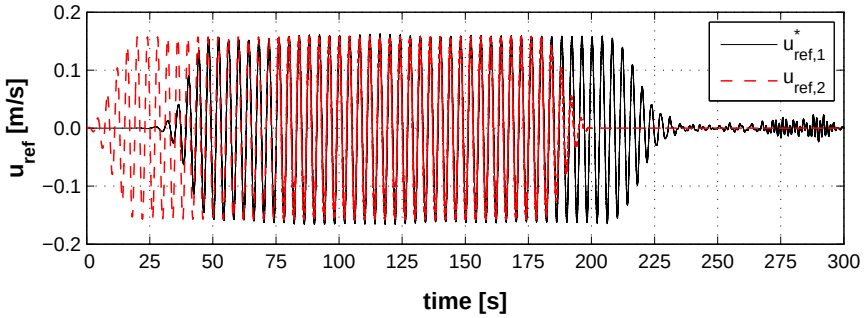
dx [m]	dt_p [s]	DTMAX/dt_p	dt [s]	$CR(f)$ [-]	$H_{m,inc}$ [m]	ϵ_{abs} [-]	drift?
0.4	0.10	1	0.09 - 0.10	0.28	0.224	+0.11	y
0.4	0.10	1/2	0.05	0.09	0.238	+0.09	n
0.4	0.05	1	0.05	0.11	0.238	+0.05	n
0.4	0.05	1/2	0.025	0.10	0.239	+0.00	n
0.4	0.025	1	0.025	0.08	0.242	+0.02	n
0.4	0.025	1/2	0.0125	0.08	0.241	+0.01	n
0.2	0.10	1	0.08 - 0.10	0.14	0.249	+0.06	n
0.2	0.10	1/2	0.05	0.11	0.250	-0.02	n
0.2	0.05	1	0.05	0.04	0.247	+0.01	n
0.2	0.05	1/2	0.025	0.12	0.257	-0.05	y
0.2	0.025	1	0.025	0.08	0.250	+0.01	y
0.2	0.025	1/2	0.0125	0.13	0.255	-0.02	y
0.1	0.10	1	0.03 - 0.07	0.12	0.231	-0.01	n
0.1	0.10	1/2	0.025	0.08	0.252	-0.02	n
0.1	0.05	1	0.05	0.05	0.241	+0.06	n
0.1	0.05	1/2	0.025	0.03	0.248	+0.00	n
0.1	0.025	1	0.025	0.08	0.234	+0.06	n
0.1	0.025	1/2	0.0125	0.13	0.249	+0.01	y
0.05	0.10	1/2	0.020 - 0.05	0.07	0.229	+0.03	n
0.05	0.05	1	0.025 - 0.05	0.06	0.229	+0.03	n
0.05	**0.05**	**1/2**	**0.025**	**0.02**	**0.247**	**-0.02**	**n**
0.05	0.025	1	0.023 - 0.025	0.13	0.221	+0.10	n
0.05	0.025	1/2	0.0125	0.07	0.237	+0.02	n
0.05	0.01	1	0.01	0.18	0.217	+0.11	y
0.05	0.01	1/2	0.005	0.08	0.248	+0.02	y



(a)

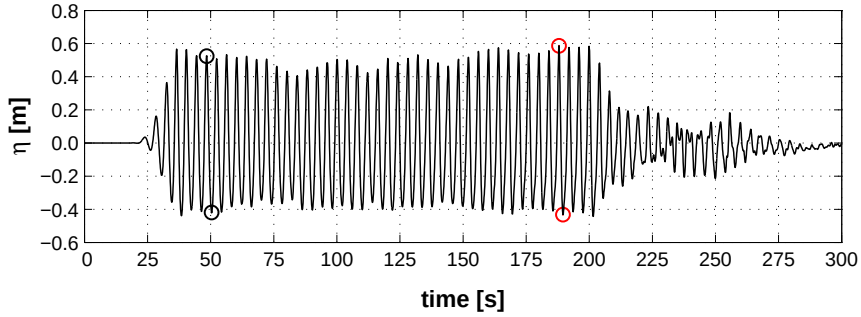


(b)

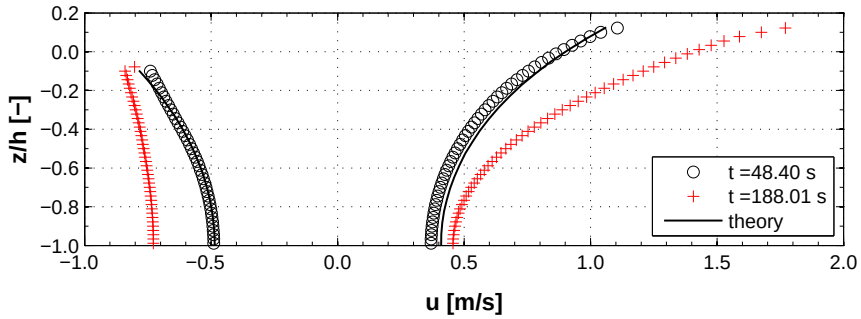


(c)

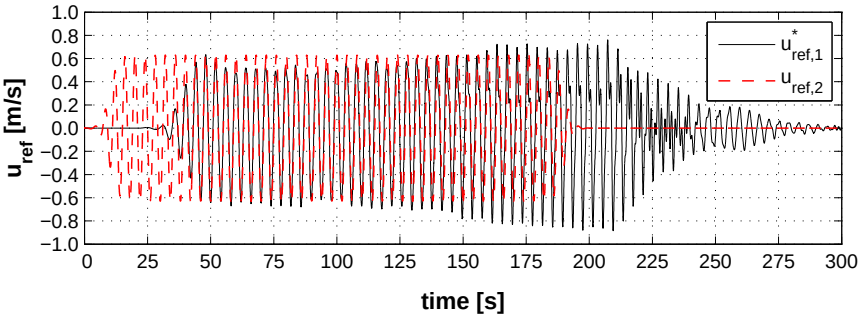
Figure 6.27: Performance of the absorbing piston: (a) time series of surface elevation at $x = x_1$; (b) profile of $u(z)$ at $x = x_1$; and (c) time series of piston control velocities. Simulation case 1 with $dx = 0.05$ m, $dt_p = 0.05$ s and DTMAX=$dt_p/2$.



(a)



(b)



(c)

Figure 6.28: Performance of the absorbing piston: (a) time series of surface elevation at $x = x_1$; (b) profile of $u(z)$ at $x = x_1$; and (c) time series of piston control velocities. Simulation case 2 with $dx = 0.10$ m, $dt_p = 0.025$ s and DTMAX=dt_p.

6.5.3 Case 2

Table 6.14 shows the simulation results obtained with different values of dx, dt_p and DTMAX. As can be expected, the performance of the absorption piston deteriorates with increasing wave nonlinearity. In optimum conditions, the reflection by the piston can be reduced to approximately 0.20, which is significantly higher than in case 1. The best solution is shown in Fig. 6.28. In the time series of surface elevation in Fig. 6.28(a), the various re-reflections at the generating piston are visible. The stability of surface elevations and velocity profiles in time is not as good as in case 1, but the piston is capable of preventing the simulation to become fully instable due to repeated reflection. Again, it is noticed that selecting a proper value for dt_p and DTMAX can improve the absorbing performance, as long as a drift of fluid velocities is avoided.

Table 6.14: Performance of the absorbing piston, for test case 2.

dx [m]	dt_p [s]	DTMAX/dt_p	dt [s]	$CR(f)$ [-]	$H_{m,inc}$ [m]	ϵ_{abs} [-]	drift?
0.2	0.10	1/2	0.028 - 0.050	0.28	1.043	-0.05	n
0.2	0.05	1	0.015 - 0.050	0.26	1.038	-0.08	n
0.2	0.05	1/2	0.025	0.37	1.049	-0.10	y
0.2	0.025	1	0.015-0.025	0.37	1.038	-0.07	y
0.1	0.10	1/2	0.015 - 0.050	0.30	1.064	-0.11	n
0.1	0.10	1/4	0.015 - 0.025	0.27	1.061	-0.12	n
0.1	0.05	1	0.008 - 0.050	0.32	1.037	-0.10	n
0.1	0.05	1/2	0.012 - 0.025	0.24	1.033	-0.09	n
0.1	**0.025**	**1**	**0.009 - 0.025**	**0.20**	**1.036**	**-0.02**	**n**
0.1	0.025	1/2	0.010 - 0.0125	0.33	1.090	-0.15	y
0.05	0.10	1	0.008 - 0.036	0.24	1.136	+0.01	n
0.05	0.05	1	0.006 - 0.036	0.24	1.072	-0.12	n
0.05	0.025	1	0.005 - 0.025	0.17	1.065	-0.07	n
0.05	0.025	1/2	0.008 - 0.0125	0.17	1.066	-0.05	n
0.05	0.010	1	0.008 - 0.010	0.17	1.053	-0.08	n
0.05	0.010	1/2	0.004 - 0.005	0.31	1.098	-0.15	y

6.5.4 Case 3

Table 6.15 shows the simulation results obtained with different values of dx, dt_p and DTMAX. In optimum conditions, the obtained wave reflection varies between 0.15-0.20, which is considerably larger than in case 1. An optimal solution is shown in Fig. 6.29. As in case 1 and 2, reducing DTMAX improves the absorbing performance in most cases, as long as a drift in the velocity profile is avoided.

Table 6.15: Performance of the absorbing piston, for test case 3.

dx [m]	dt_p [s]	DTMAX/dt_p	dt [s]	$CR(f)$ [-]	$H_{m,inc}$ [m]	ϵ_{abs} [-]	drift?
0.4	0.10	1	0.10	0.16	0.258	-0.00	n
0.4	0.10	1/2	0.05	0.15	0.261	-0.02	n
0.4	0.05	1	0.05	0.15	0.263	-0.01	n
0.4	0.05	1/2	0.025	0.15	0.263	-0.01	n
0.4	0.025	1	0.025	0.15	0.262	-0.01	n
0.4	0.025	1/2	0.0125	0.14	0.265	-0.02	n
0.2	0.10	1	0.081-0.10	0.16	0.261	+0.00	n
0.2	0.10	1/2	0.05	0.15	0.260	+0.01	n
0.2	0.05	1	0.05	0.19	0.234	+0.10	n
0.2	0.05	1/2	0.025	0.18	0.243	+0.07	y
0.1	0.10	1	0.045 - 0.071	0.16	0.253	+0.05	n
0.1	0.10	1/2	0.047 - 0.05	0.17	0.260	+0.02	n
0.1	0.05	1	0.043 - 0.05	0.22	0.224	+0.13	n
0.1	0.05	1/2	0.025	0.20	0.233	+0.11	n
0.1	0.025	1	0.024 - 0.025	0.27	0.200	+0.21	n
0.1	0.025	1/2	0.0125	0.20	0.240	+0.08	y
0.05	0.10	1	0.023 - 0.051	0.17	0.240	+0.09	n
0.05	**0.10**	**1/4**	**0.022 - 0.025**	**0.19**	**0.250**	**+0.05**	**n**
0.05	0.05	1	0.024 - 0.05	0.22	0.215	+0.19	n
0.05	0.05	1/2	0.025	0.22	0.211	+0.19	n
0.05	0.025	1	0.025	0.33	0.182	+0.26	n
0.05	0.025	1/2	0.0125	0.31	0.192	+0.23	n

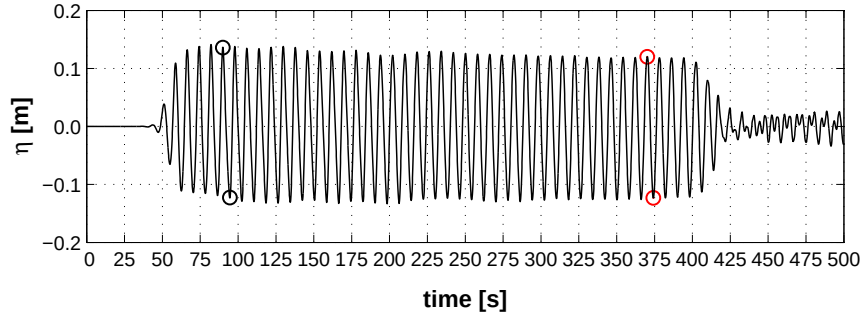
6.5.5 Optimal time step control

As in the case of purely progressive waves, $dt_p = f_{sf}^{-1}$ should be selected in accordance with the mesh resolution. An important difference however concerns the specification of DTMAX, which needs to be restricted to dt_p. The previous test cases lead to the following considerations that should be taken into account when specifying dt_p and DTMAX:

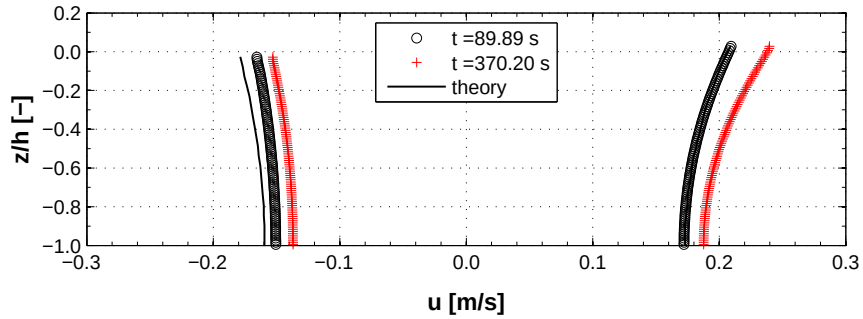
1. A maximum piston time step size dt_p in order to limit the difference between dt_p and dt set by the automatic time step control. This avoids the piston control velocities to look like a 'step' signal;

2. A minimum dt_p because of the occurrence of drift of the fluid velocities near the free surface;

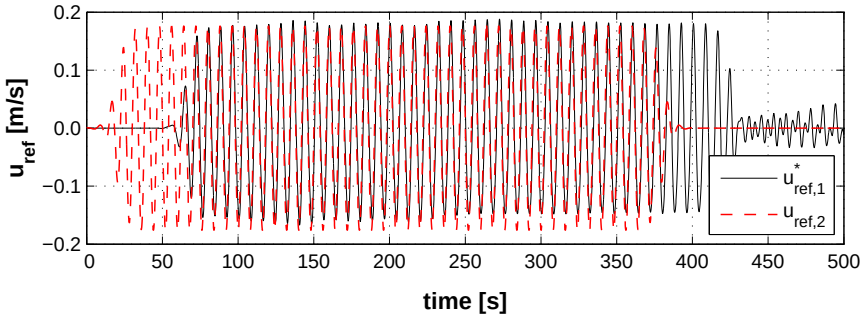
3. The stability of velocity profiles and absorption performance improves when the ratio of dt to dt_p decreases. This can be achieved by reducing $dtmax$, however not below the limit which would lead to a drift in the velocity profile.



(a)



(b)



(c)

Figure 6.29: Performance of the absorbing piston: (a) time series of surface elevation at $x = x_1$; (b) profile of $u(z)$ at $x = x_1$; and (c) time series of piston control velocities. Simulation case 3 with $dx = 0.05$ m, $dt_p = 0.10$ s and DTMAX=$dt_p/4$.

6.6 Conclusions

A first-order piston wavemaker with active wave absorption has been implemented in FLOW-3D, using the GMO model to represent the piston motion. The operation of the piston has been tested based on a selected number of wave conditions, with varying wave nonlinearity. A basic grid convergence study with purely progressive waves shows that a stable and accurate wave generation and propagation can be achieved with the first-order upwind momentum advection (IORDER=1, ALPHA=1). The accuracy of the solution may vary significantly according to the time step control and specification of the piston time step dt_p. Guidelines to select these numerical parameters are included in section 6.3.

Additional long-duration test series were performed to verify the stability of the wave train in time. The first-order upwind scheme proves to be stable, provided the time step size is not reduced below the lower stability limit set by the automatic time step control (AUTOT=1/2). This might be an issue when modeling wave-structure interaction, where a reduction of dt cannot be avoided. A satisfactory explanation for the occurrence of a drift in the velocity profile under these circumstances is lacking to date. Results obtained with an increased number of grid cells in the transverse direction do not show the deviations associated with the time step reduction. In addition, tests with the second-order scheme do not exhibit the instabilities observed with the first-order upwind scheme. The previous considerations suggest that the cause of this problem may be found in the numerical implementation.

Finally, the performance of the linear active wave absorption system has been validated by means of a piston wavemaker in pure absorption mode, for short and long waves with varying wave height. Results show excellent absorption capacity for low-amplitude waves and a lower performance with larger wave heights, although still acceptable. Notwithstanding the active absorption is clearly bound by the limitations of the linear wave absorption system, it seems that its performance is as least as high as in a physical wave flume.