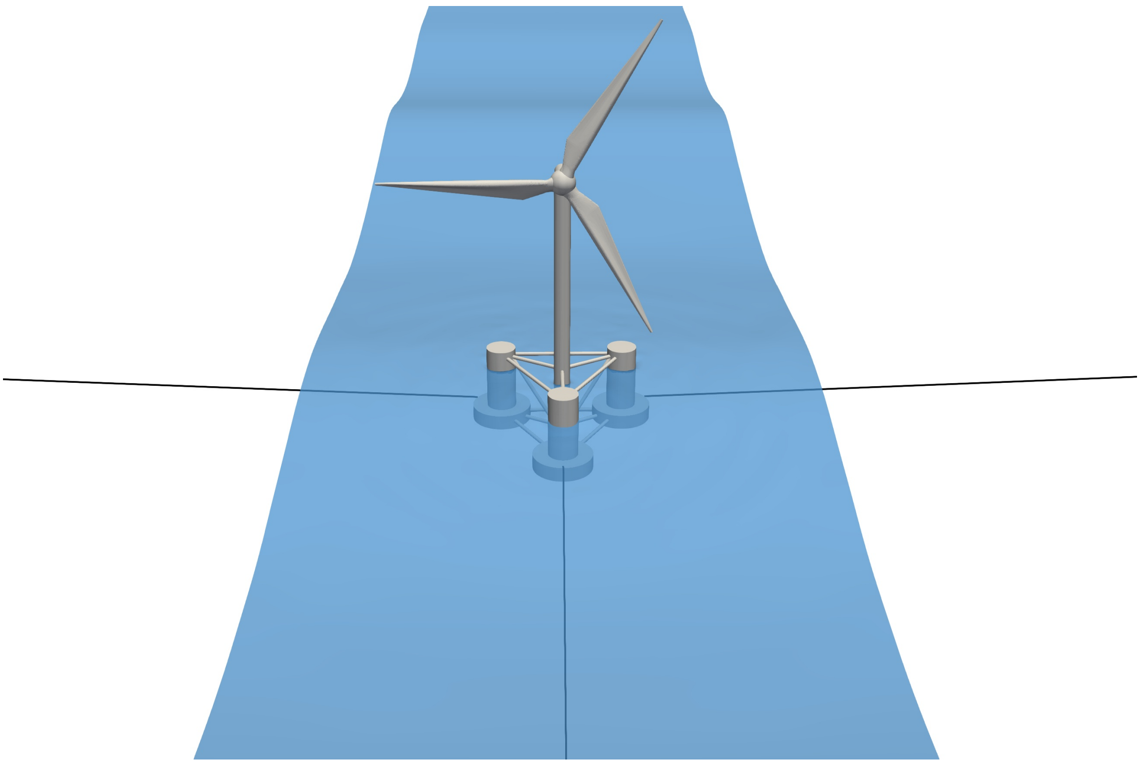
REEF3D :: User Guide



REEF3D 25.01

Contact

Hans Bihs

Professor
Marine Civil Engineering
NTNU Trondheim

hans.bihs@ntnu.no

Contents

1. Introduction

Todays ever increasing computational resources make it possible to compute incredible large flow problems. Every company and research institution interested in performing serious CFD (computational fluid dynamics) simulations has now the real option to buy and maintain supercomputers at reasonable costs. The limiting factor of such simulations becomes less the problem size but rather the time it takes for the engineer to generate grids, start simulations and post-process the results. With the increasing speed of high-performance computer clusters, the amount of grid points is growing. As Peric Peric [2004] writes, in the future the "computational points in a mesh will be counted in billions", the issue of grid quality will become less relevant because the mesh will be very fine in any case.
With these observations in mind, the numerical model REEF3D was designed under the following premises:

1. hydraulic, coastal, marine and environmental engineering
    - level set method for complex free surface flow
    - open channel flow boundary conditions
    - numerical wave tank boundary conditions
    - sediment transport
2. ease of use: grid generation as limiting factor of user productivity
    - immersed boundary
    - STL file input
    - easy natural bathymetry handling
3. increasing computer performance enables larger computations, but can only be exploited in parallel
    - full parallelization based on the domain decomposition strategy and MPI
4. ease of use: stable, accurate and fast numerical simulation
    - staggered grid: tight velocity-pressure coupling
    - very accurate and stable WENO discretization
    - adaptive time-stepping for maximum stability
5. numerical tool should be easy to maintain, changes should be possible
    - C++ Programming
    - object oriented code
    - MPI, an industry standard for high-performance computing

The source code of REEF3D is available at `www.reef3d.com` and is published under the GPL license, version 3. Open-source CFD software has several advantages. All code developments have the potential to benefit a large audience, including students, researchers and engineers working on academic or commercial problems. The usage of open-source CFD programs comes at no cost. It gives the people who work with the software the opportunity to gain insights into how the numerical model works and build up valuable competence and experience in this field. This makes contributions to open-source software a sustainable investment in a lot of ways. Open-source also means that REEF3D is more flexible and more open to innovation.

2. Model Overview

2.1 CFD

REEF3D::CFD is a three-dimensional Navier-Stokes solver. From the start, the focus of this model was to solve complex free surface dynamics through two-phase flow interface capturing. In order to ensure high levels of accuracy while maintaining robustness, REEF3D::CFD is designed on a staggered rectilinear grid. This results in tight pressure-velocity coupling, crucial for stable two-phase flow modeling with its significant density and pressure jumps across the interface.This aspect comes especially into play when modeling waves, where REEF3D's implementation avoids any form of spurious air flow over the wave free surface. The rectilinear mesh enables the use of high-order conservative finite differences for the discretization of the governing equations. As a result, a relatively coarse mesh can be used when using REEF3D::CFD as a numerical wave tank. A potential loss of flexibility due to the mesh arrangement is avoided through the immersed boundary treatment for complex solid geometries. Mesh generation is performed in the accompanying open-source software DIVEMesh. This pre-processing tool handles solid geometries from STL files, bed interpolation from measured data points, domain decomposition and hydrodynamics coupling. The source code is written in modular C++, enabling straightforward expansion and maintenance of the model. The unique characteristics of REEF3D::CFD are:

- High-order discretization in space (5th-order WENO)
- High-order discretization in time (3rd-order TVD Runge-Kutta)
- Full Parallelization with domain decomposition and MPI
- Geometric multigrid preconditioned CG-solver (hyper) for efficient solution of the pressure Poisson equation
- Complex free surface modeling with the level set method
- Immersed boundary for complex solids
- Modular C++ for straightforward expansion and maintenance of the model

The flow solver can be used together with the following multiphysics solvers:

- 6DOF algorithm for floating bodies
- Mooring line dynamics
- Static and dynamic net modeling
- Sediment transport
- Pollutant transport
- Heat transfer

- Turbulence modeling: RANS & LES
- Porous Media: Darcy or VRANS
- Non-Newtownian Rheology
- Wave generation with a rich library of wave theories
- Hydrodynamics coupling (HDC) with REEF3D::FNPF

2.2 NHFLOW

REEF3D::NHFLOW is a three-dimensional non-hydrostatic wave model. This model is implemented in the open-source hydrodynamics framework REEF3D. It uses a Godunov-type scheme for shock-capturing properties, albeit with WENO flux reconstruction. In order to achieve good dispersion characteristics, a pressure correction. As part of the REEF3D framework, MPI parallelization, hypre's multigrid solvers and high-order finite difference WENO discretization methods are readily available In contrast to m the existing fully-nonlinear potential flow solver FNPF, the new model includes the capability to model the dynamic wetting and drying processes in the swash zone.

- High-order discretization in space (5th-order WENO) for the free surface boundary conditions
- High-order discretization in time (3rd-order TVD Runge-Kutta)
- Full Parallelization with domain decomposition and MPI
- Robust and accurate breaking waves through shock-absorbing solver
- Wetting and drying algorithm for complex coastlines
- Wave propagation from deep water to shallow water
- Geometric multigrid preconditioned CG-solver (hyper) for efficient solution of the Laplace equation
- Varying bathymetry with geometry input form DIVEMesh

2.3 FNPF

REEF3D::FNPF is a fully nonlinear potential flow model. The starting point for this model are the unique characteristics of the Norwegian coast: The water depth changes rapidly from deep to shallow water, the coastline is highly irregular and an immense number of small islands complicate meshing. REEF3D::FNPF is tailor-made to incorporate solutions to the Norwegian coastal features, and consequently the model can be used for wave propagation from deep to shallow water while taking care of complex coastlines in an easy to use way. In addition, REEF3D::FNPF has proved to serve as an highly accurate and efficient tool to generate typical 3-hour sea states used for designing offshore and coastal structures. Here, the selected numerical approach has shown to be so robust that extreme sea states with steep nonlinear waves including breaking wave kinematics can be handled. Up to now, REEF3D::FNPF has been used for a range of coastal and offshore industry projects. The unique characteristics of REEF3D::FNPF are:

- High-order discretization in space (5th-order WENO) for the free surface boundary conditions
- High-order discretization in time (3rd-order TVD Runge-Kutta)
- Full Parallelization with domain decomposition and MPI
- Robust and accurate breaking wave model
- Coastline algorithm for efficient inclusion of complex topographies
- Wave propagation from deep water to shallow water
- Geometric multigrid preconditioned CG-solver (hyper) for efficient solution of the Laplace equation
- Varying bathymetry with geometry input form DIVEMesh
- Hydrodynamic coupling to REEF3D::CFD

2.4 SFLOW

REEF3D::SFLOW is a non-hydrostatic shallow water equations (SWE) model. This means that all flow quantities are depth-averaged, resulting in a two-dimensional solver. The model can be use for wave propagation and open channel flow. Sediment transport is incorporated through bedload based transport mode. The unique characteristics of REEF3D::SLOW are:

- High-order discretization in space (5th-order WENO)
- High-order discretization in time (3rd-order TVD Runge-Kutta)
- Full Parallelization with domain decomposition and MPI, delivering very good scalability
- Quadratic dynamic pressure assumption, leading to correct phase velocities for shallow and intermediate wave conditions
- Geometric multigrid preconditioned CG-solver (hyper) for efficient solution of the dynamic pressure equation
- Varying bathymetry with geometry input form DIVEMesh

3. Functions

A file called “ctrl.txt” is used to give the necessary data to the program. The structure of this file is as follows: a capital letter and a number are used to enable different algorithms followed by the corresponding variables. The “ctrl.txt” file needs to be stored in the same location as the executable REEF3D. In front of the variables in the definition of the algorithms you find either an “int” or a “float”. For “int” an integer needs to be given, for “float” a floating-point number is expected by the program. It is possible to enter comments in the “ctrl.txt” by entering the followed by the comment, but take care not to include capital letters as that will cause the program to look for a valid command option.
Most of the options in this section (A) will determine the discretization and other parameters for the first three models listed above. The validity of the functions for the particular models is denoted by the filled circle beside the function under the columns S (SFLOW), F (FNPF), N (NHFLOW) and C (CFD) in all the sections for functions in this manual.
Commands with an asterix ($*$) can be issued multiple times to create multiple instances of the call.

3.1 A :: Hydrodynamic models

The commands in this section are used to select the hydrodynamic model that is to be activated for the simulation. This will determine the governing equations that will be used for the simulations. The different models available in REEF3D are:

- **S** : SFLOW: A depth-averaged shallow water equations model, that solves the non-hydrostatic pressure.
- **F** : FNPF: This model solves the Laplace equation.
- **N** : NHFLOW: This model solves the Navier-Stokes in three dimensions on a sigma grid, and is suitable for large scale wave modeling with a single-value free surface.
- **C** : CFD : This solves the 3D RANS equations with two-phase flow.

Valid for:
S F N C

3.1.1 SFLOW

● ● ● ●

A 10 int Hydrodynamic Model

- **2** SFLOW
- **3** FNPF
- **4** PTF
- **5** NHFLOW
- **6** CFD
- **default:** 6

● ○ ○ ○

A 210 int Time scheme for SFLOW momentum equations

- **2** 2nd-order Runge-Kutta
- **3** 3rd-order Runge-Kutta
- **default:** 3

● ○ ○ ○

A 211 int Convection scheme for SFLOW momentum equations

- **1** FOU
- **2** CDS2
- **4** WENO FLUX
- **5** WENO HJ
- **default:** 4

● ○ ○ ○

A 212 int Turn on diffusion for SFLOW velocities

- **0** OFF
- **1** explicit
- **2** implicit
- **default:** 0

● ○ ○ ○

A 214 int Turn on convection for vertical SFLOW velocity w_s

0 OFF
1 ON
default: 1

● ○ ○ ○

A 217 int Boundary conditions at walls

1 slip
2 no-slip
default: 2

● ○ ○ ○

A 218 int Turn on roughness

0 OFF
1 ON
default: 0

● ○ ○ ○

A 220 int Pressure scheme for SFLOW dynamics pressure w_s

0 hydrostatic
1 linear non-hydrostatic pressure
2 quadratic non-hydrostatic pressure
default: 2

● ○ ○ ○

A 221 int Type of hydrostatic pressure gradient

0 OFF
1 standard
2 bounded
default: 1

● ○ ○ ○

A 223 double Blending of previous and current pressure for gradient calculation

default: 0.5

● ○ ○ ○

A 240 int Free surface scheme for SFLOW

0 OFF
1 ON
default: 1

● ○ ○ ○

A 241 int Discretization scheme for SFLOW water levels

1 FOU
2 CDS2
4 WENO FLUX
default: 1

● ○ ○ ○

A 242 int Hydrostatic pressure for shallow areas

0 OFF
1 ON
default: 0

● ○ ○ ○

A 243 int Turn on wetting and drying algorithm

0 OFF
1 ON
default: 1

● ○ ○ ○

A 244 double Use absolute wetting-drying threshold value

default: 0.00005

● ○ ○ ○

A 245 double Use relative wetting-drying threshold value (dx multiplier)

default: 0.001

● ○ ○ ○

A 246 int Turn on breaking wave algorithm (turns off dynamics pressure for breaking waves)

0 OFF
1 ON
default: 1

● ○ ○ ○

A 247 double Breaking wave threshold parameter

default: 0.6

● ○ ○ ○

A 248 int Turn on breaking wave persistance algorithm (turns off dynamics pressure for breaking waves)

0 OFF
1 ON
default: 1

● ○ ○ ○

A 249 double Breaking wave persistence parameter

default: 0.3

● ○ ○ ○

A 260 int Turbulence model

0 OFF
1 k-ϵ model
3 Prandtl length scale model
4 Parabolic eddy-viscosity model
default: 0

● ○ ○ ○

A 261 double Length scale factor for length scale turbulence model

default: 0.267

● ○ ○ ○

A 262 double Factor for parabolic eddy-viscosity model

default: 0.0667

3.1.2 FNPF

○ ● ○ ○

A 310 int Time scheme for FNPF algorithm

3 3rd-order TVD Runge-Kutta
4 4th-order Runge-Kutta
default: 3

○ ● ○ ○

A 311 int Spatial discretization for KFSFBC and DFSFBC

0 OFF
1 CDS2
2 CDS4
4 WENO5
5 WENO5 with wetting-and-drying
6 CDS6
7 WENO7
default: 5

○ ● ○ ○

A 312 int Spatial discretization for sigma coordinate system

2 CDS2
3 CDS4
default: 2

○ ● ○ ○

A 320 int Order of spatial discretization for Laplace equation
The 4th-order discretization requires the HYPRE AI solvers and preconditioners.

1 CDS2

2 CDS4

default: 1

○ ● ○ ○

A 321 int Order of KBEDBC for 4th-order Laplace equation

1 2nd-order

2 4th-order

default: 1

○ ● ○ ○

A 329 int Order of Dirichlet wave generation and active wave absorption velocity potential extrapolation

1 1st-order

2 2nd-order

default: 1

○ ● ○ ○

A 341 double coastline relaxation zone factor times dx

default: 0.0

○ ● ○ ○

A 342 double coastline relaxation zone absolute distance

default: 0.0

○ ● ○ ○

A 343 int Turn on wetting-and-drying

0 OFF

1 ON

default: 1

○ ● ○ ○

A 344 double Use absolute wetting-drying threshold value

default: 0.001

○ ● ○ ○

A 345 double Use relative wetting-drying threshold value (dx multiplier)

default: 0.01

○ ● ○ ○

A 346 double Damping viscosity within the coastline, modulated by the coastline relaxation zone.

default: 1.86

○ ● ○ ○

A 347 int Coastline relaxation is used for:

1 eta and Fi

2 eta

3 Fi

default: 2

○ ● ○ ○

A 348 int Numerical beach relaxation is used for:

1 eta and Fi

2 eta

3 Fi

default: 2

○ ● ○ ○

A 350 int Breaking wave algorithm

0 OFF

1 viscosity based

2 filter based

default: 0

○ ● ○ ○

A 351 int Breaking wave detection

0 OFF

1 shallow water

2 deep water

3 shallow and deep water

default: 0

○ ● ○ ○

A 352 int Added filtering for breaking when viscosity breaking algorithm is used for additional robustness, for breaking in:

0 OFF

1 shallow water

2 deep water

3 shallow and deep water

default: 1

○ ● ○ ○

A 354 double Breaking wave threshold parameter shallow water

default: 0.6

○ ● ○ ○

A 355 double Breaking wave slope threshold deep water

default: 1.25

○ ● ○ ○

A 357 int Breaking wave algorithm is used for:

1 eta and Fi

2 eta

3 Fi

default: 1

○ ● ○ ○

A 361 int Breaking wave filter algorithm: outer iterations

default: 5

○ ● ○ ○

A 362 int Breaking wave filter algorithm: inner iterations

default: 2

○ ● ○ ○

A 365 double Breaking wave viscosity

default: 1.86

○ ● ○ ○

A 368 int Breaking wave viscosity based algorithm inside numerical beach relaxation zone

0 OFF
1 ON
default: 0

3.1.3 NHFLOW

○ ○ ● ○

A 510 int Time scheme for FLOW momentum equations

2 2nd-order Runge-Kutta
3 3rd-order Runge-Kutta
default: 2

○ ○ ● ○

A 511 int Discretization scheme for NHFLOW momentum equations

1 HLL
2 HLLC
default: 1

○ ○ ● ○

A 514 int Reconstruction scheme for NHFLOW momentum equations

1 van Leer TVD

2 Superbee TVD

4 WENO reconstruction

default: 4

○ ○ ● ○

A 518 int Bed boundary condition

1 slip

2 no-slip

default: 2

○ ○ ● ○

A 519 int Bed roughness for velocities

0 OFF

1 ON

default: 1

○ ○ ● ○

A 520 int Pressure scheme for NHFLOW dynamics pressure w_s

0 hydrostatic

1 PJM for non-hydrostatic pressure

2 Pressure correction for non-hydrostatic pressure

default: 2

○ ○ ● ○

A 531 double Froude number limiter

default: 3.0

○ ○ ● ○

A 544 double Use absolute wetting-drying threshold value

default: 0.001

● ○ ○ ○

A 560 int Turbulence model

0 OFF

2 k-ω model

default: 0

○ ○ ● ○

A 570 int Type of wind forcing for NHFLOWw_s

0 OFF

2 Wu

default: 0

● ○ ○ ○

A 571 double Wind velocity U_{10}, double Wind direction (degree)

0 OFF

2 k-ω model

default: 0

○ ○ ● ○

A 573 int Area of wind forcing w_s

1 only wave crests

2 everywhere

default: 2

3.2 B :: Boundary Conditions

Valid for:
S F N C
○ ○ ● ●

B 10 int Wall functions for the velocities

0 OFF

1 ON

default: 1

○ ○ ● ●

B 20 int Slip or no-slip boundary conditions for velocities

> **1** slip
> **2** no-slip
> **default:** 2

○ ○ ● ●

B 23 int Extrapolation or reflection for ghostcell updates

> **1** extrapolation
> **2** reflection
> **default:** 1

○ ○ ○ ●

B 30 int Type of pressure reference point

> **0** OFF
> **1** fixed location (see B 32)
> **2** moves with free surface location at inlet
> **3** atmospheric pressure at lid
> **4** map zero-pressure contour to free surface
> **default:** 0

● ○ ● ●

B 31 double Pressure reference value

> **default:** 0.0

● ○ ● ●

B 32 double x-, y-, and z-location for fixed pressure reference value location

> If not given when used together with B 30 1, the inlet location with the smallest x- and y-coordinates are used.
>
> **default:** 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

B 33 int Type of pressure reference gage normalization For certain configurations, inline pressure normalization causes instabilities.

1 virtual

2 inline

default: 1

● ○ ● ●

B 50 double Global wall roughness

default: 0.001

○ ○ ○ ●

B 51 double Wall 1 roughness

default: nan

○ ○ ○ ●

B 52 double Wall 2 roughness

default: nan

○ ○ ○ ●

B 53 double Wall 3 roughness

default: nan

○ ○ ○ ●

B 54 double Wall 4 roughness

default: nan

○ ○ ○ ●

B 55 double Wall 5 roughness

default: nan

○ ○ ○ ●

B 56 double Wall 6 roughness

default: nan

● ○ ● ●

B 60 int Enable ioFlow for open channel flow

This parameter is important for open channel flow calculations. The ioFlow module makes sure, that the inflow discharge as given in W10 is constant throughout the computation even though the inflow water level may change. This is the counterpart to B90 for open channel flow. When selecting the hdyrograph inflow option, make sure that a file with the name "hydrograph.dat" is present in the simulation folder. When selecting the hdyrograph outflow option, make sure that a file with the name "hydrograph_out.dat" is present in the simulation folder. The data format of the hydrograph file consists of two columns, the first one gives the time in seconds and the second column the discharge in m^3/s.

0 OFF

1 constant inflow

2 hydrograph inflow

3 hydrograph outflow

4 hydrograph inflow and outflow

default: 0

○ ○ ○ ●

B 61 int Inflow profile for ioFlow

1 constant velocity

2 logarithmic profile bed

3 constant velocity only for phase 1

4 logarithmic profile only for phase 1, bed

5 logarithmic profile only for phase 1, sidewalls and bed

default: 2

○ ○ ○ ●

B 71 * double value, double distance, double line angle, double line x-origin, double line y-origin; for use of relaxation method for fixed water level for initialization only

default: na

○ ○ ○ ●

B 75 int Outflow boundary condition

- **1** zero gradient
- **2** convection condition

default: 1

● ○ ○ ●

B 76 int Pressure inflow boundary conditions

- **0** hydrostatic pressure
- **1** zero gradient pressure

default: 1

● ○ ○ ●

B 77 int Pressure outflow boundary conditions

- **1** controlled outflow with pressure condition
- **2** controlled outflow with free surface condition
- **10** free stream outflow (zero pressure)

default: 1

○ ○ ○ ●

B 81 double x-location for focus, y-location for focus, time of wave focusing (For a 2D simulation set y-location to 0.0)

default: 0.0 ;0.0; 0.0

● ● ● ●

B 82 int Type of focusing amplituted calculation

- **1** NewWave
- **2** Spectrum
- **3** Constant wave steepness spectrum
- **4** Constant wave amplitude

default: 1

● ● ● ●

B 83 double Gain factor per wave component for constant wave steepness spectrum focused waves. The amplitudes of the wave components A_i are calculated using $A_i = \frac{B83}{k_i}$. The frequency distribution is calculated as defined in B 84.

default: 0.0025

● ● ● ●

B 84 int Frequency spectrum discretisation method

1 Peak enhance method

2 Equal energy method

3 Uniform distribution

default: 1

● ● ● ●

B 85 int Wave spectrum for irregular waves In case of the spectrum file, provide a "spectrum-file.dat", with the two columns "ω" and "S".

1 Pierson-Moskowitz

2 JONSWAP

3 Torsethaugen

4 Wavepackets

5 Wavepackets for steep waves

6 Gaussian Wavepackets

10 spectrum file

21 Goda JONSWAP

22 TMA

default: 1

● ● ● ●

B 86 int Number of regular waves for irregular wave generation

default: 10

● ● ● ●

B 87 double ω_s and ω_e for irregular wave generation When not given, the model will calculate the values automatically.

default: 0.0 ; 0.0

• • • •

B 88 double γ for irregular wave generation with JONSWAP spectrum

default: 3.3

• • • •

B 89 int Turn on wave generation optimization through space-time decomposition for relaxation method. Currently supported for the following wave theory options:

B 92 5, B 92 31, B 92 41, B 92 51

0 OFF

1 ON

default: 0

• • • •

B 90 int Enable ioWave for numerical wave tank This parameter turns on the numerical wave tank. It is the counterpart to B60 for wave simulations. Different options for generating and dissipating waves exist.

0 OFF

1 ON

default: 0

• • • •

B 91 double wave height, wave length

The main wave parameters are given here. For regular waves the first values is H, whereas for irregular waves, the first value is the significant wave height. Alternatively B93 can be used, when the wave period is known.

default: 0.0 ; 0.0

• • • •

B 92 int Wave type Different wave theories can be used as input for the numerical wave tank. It is important, to check beforehand whether the selected theory fits the given wave conditions consisting of wave height, wave length/wave period and still water level.

For the piston wavemaker theory with η timeseries input, provide a "wavemaker_eta.dat" input file with two columns "t" and "η"

For the piston and flap wavemaker theories, provide a "wavemaker.dat" input file with two columns "t" and "X" or "t" and "β", depending on input from B 116.

For the double-hinged flap wavemaker, two "X" or two "β" columns are expected. For giving two "X" columns, this will be the total exitation at the end of each flap. For giving two "β" columns, the second angle is the additional exitation of the second flap.

For the wave reconstruction, provide a "waverecon.dat" input file with three columns giving "A" and "ω" and phase "ϕ" for each of the harmonic wave components.

Wave theories which prescribe the potential directly are marked with a $*$ and are suitable for relaxation zone wave generation in FNPF. Those wave theories without $*$ require either B 98 3 or B 98 4 for FNPF.

1 Shallow Water Waves

2 Linear Waves $*$

3 Deep Water Waves $*$

4 $2nd$-order Stokes Waves $*$

5 $5th$-order Stokes Waves (Fenton) $*$

6 Shallow Water Cnoidal Waves

7 $1st$-order Cnoidal Waves

8 $5th$-order Cnoidal Waves

9 $1st$-order Solitary Wave

10 $3rd$-order Solitary Wave

11 $5th$-order Stokes Waves (Skjelbreia)

20 Piston Wavemaker based on η timeseries

21 Piston Wavemaker

22 Flap Wavemaker

23 Double-hinged Flap Wavemaker

31 $1st$-order Irregular Waves $*$

32 $2nd$-order Irregular Waves (Longuet-Higgins)

33 $2nd$-order Irregular Waves (Schäffer) $*$

41 $1st$-order Focused Waves $*$

42 $2nd$-order Focused Waves (Longuet-Higgins)

43 $2nd$-order Focused Waves (Schäffer) $*$

51 Wave Reconstruction with $1st$-order Irregular Waves $*$

52 Wave Reconstruction with $2nd$-order Irregular Waves (Longuet-Higgins)

53 Wave Reconstruction with $2nd$-order Irregular Waves (Schäffer) $*$

61 Hydrodynamic Coupling HDC (FNPF to CFD or SFLOW to CFD)

70 Steady Surface Gravity Wave SSGW $*$ (currently FNPF relaxation wavegen only)

default: 0

● ● ● ●

Wave Generation Compatibility					
B 92	Wave Type	SFLOW	FNPF	NHFLOW	CFD
1	Shallow Water Waves	D + R	D	D + R	D + R
2	Linear Waves	D + R	D + R	D + R	D + R
3	Deep Water Waves	D + R	D + R	D + R	D + R
4	$2nd$-order Stokes Waves	D + R	D + R	D + R	D + R
5	$5th$-order Stokes Waves (Fenton)	D + R	D + R	D + R	D + R
6	Shallow Water Cnoidal Waves	D + R	D	D + R	D + R
7	$1st$-order Cnoidal Waves	D + R	D	D + R	D + R
8	$5th$-order Cnoidal Waves	D + R	D	D + R	D + R
9	$1st$-order Solitary Wave	D + R	D	D + R	D + R
10	$3rd$-order Solitary Wave	D + R	D	D + R	D + R
11	$5th$-order Stokes Waves (Skjelbreia)	D + R	D	D + R	D + R
20	Piston Wavemaker, η timeseries	D	D	D	D
21	Piston Wavemaker	D	D	D	D
22	Flap Wavemaker	D	D	D	D
23	Double-hinged Flap Wavemaker	D	D	D	D
31	$1st$-order Irregular Waves	D + R	D + R	D + R	D + R
32	$2nd$-order Irregular Waves (Longuet-Higgins)	D + R	D + R	D + R	D + R
33	$2nd$-order Irregular Waves (Schäffer)	D + R	D + R	D + R	D + R
41	$1st$-order Focused Waves	D + R	D + R	D + R	D + R
42	$2nd$-order Focused Waves (Longuet-Higgins)	D + R	D + R	D + R	D + R
43	$2nd$-order Focused Waves (Schäffer)	D + R	D + R	D + R	D + R
51	Wave Reconstruction with $1st$-order Irregular Waves	D + R	D + R	D + R	D + R
52	Wave Reconstruction with $2nd$-order Irregular Waves (Longuet-Higgins)	D + R	D + R	D + R	D + R
53	Wave Reconstruction with $2nd$-order Irregular Waves (Schäffer)	D + R	D + R	D + R	D + R
61	Hydrodynamic Coupling HDC	D + R	D + R	D + R	D + R
70	Steady Surface Gravity Wave SSGW		R		

Table 3.1: Wave generation compatibility for B 92, D = Dirichlet/Neuman generation; R = Relaxation generation.

B 93 double wave height, wave period

The main wave parameters are given here. For regular waves the first values is H, whereas for irregular waves, the first value is the significant wave height. Alternatively B91 can be used, when the wavelength is known.

default: 0.0 ; 0.0

B 94 double water depth d for wave theory

If not given, d will be set to F 60, which requires the bed to be located at $z = 0.0$ m.

default: na

● ● ● ●

B 96 double Wave Generation and Beach parameters, dist1, dist2

dist1 ist the distances from the beginning of the wave flume (default, can be changed, see B 106), dist2 is the distance measured from the end of the wave flume (default, can be changed, see B 107).

default: 0.0 ; 0.0

● ● ● ●

B 98 int Wave Generation Method Relaxation Method 2 ramps up the values for the input wave within one zone following the function by Jacobsen et al. [2012]. Relaxation Method 1 uses two zones to generate the waves Afshar [2010]. For the Dirichlet type, waves are generated by assigning the values for the free surface and the velocity in the inflow boundary only.

0 OFF

2 Relaxation Method 2

3 Dirichlet

4 Dirichlet with Active Absorption, based on shallow water theory

default: 0

● ● ● ●

B 99 int Numerical Beach Method Both relaxation methods dissipate the wave energy by ramping the free surface down to the still water level, the velocity to the zero and the pressure to its hydrostatic distribution for still water conditions. Typically the length of the dissipating relaxation zone should be two wavelengths. The Active Wave Absorption avoids reflections by generating a wave opposite of the reflected wave, thus canceling it out. The method is more efficient in computational terms, as it does not require additional space.
In order to use the Active Wave Absorption beach, keep in mind to use the active beach boundary condition in the C data set in DIVEMesh.

0 OFF

1 Relaxation Method 1

2 Relaxation Method 2

3 Active Wave Absorption, based on shallow water theory

4 Active Wave Absorption, based on intermediate water theory

5 Active Wave Absorption, using a flap wavemaker (see B 123)

default: 0

● ● ● ●

B 101 int Ramp function for wave generation

0 OFF

1 On, based on wave period

2 On, based on simulation time

default: 0

● ● ● ●

B 102 double Ramp function for duration in wave periods or simulation time (see B 101)

default: 1.0

● ● ● ●

B 105 double angle β, x and y coordinates for the distance origin for the generated wave

default: 0.0 ; 0.0 ; 0.0

● ● ● ●

B 106 * double angle β, x and y coordinates for the distance origin of wave generation relaxation zone

default: 0.0 ; 0.0 ; 0.0

● ● ● ●

B 107 * double $x_{start}, x_{end}, y_{start}, y_{end}$ and acting distance d of numerical beach relaxation zones.

As default, without any user input for B 107, a relaxation zone for the numerical beach is created at the end of the domain, i.e. at the maximum x-coordinates of the domain, with an orientation perpendicular to the x-axis and with infinite length using the relaxation distance from B 96. If user input is provided for B 107, the default relaxation zone will not be created and the input from B 96 will not be used. Instead the start, end and acting distance of the relaxation zone of the numerical beach are provided through B 107. Multiple numerical beach generation zones are possible.

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

● ● ● ●

B 108 double $x_{start}, x_{end}, y_{start}, y_{end}$ and acting distance d of wave generation relaxation zones.

As default, without any user input for B 108, a relaxation zone for the wave generation is created at the beginning of the domain, i.e. at the minimu x-coordinates of the domain, with an orientation perpendicular to the x-axis and with infinite length using the relaxation distance from B 96. If user input is provided for B 108, the default relaxation zone will not be created and the input from B 96 will not be used. Instead the start, end and acting distance of the relaxation zone of the wave generation are provided through B 108.

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

● ● ● ●

B 110 double z_{start} z_{end} for the piston wavemaker

default: 0.0 ; 0.0

● ● ● ●

B 111 double z_{start} z_{end} for the flap wavemaker

default: 0.0 ; 0.0

● ● ● ●

B 112 double z_{start} z_{hinge_2}, z_{end} for the double-hinged flap wavemaker

default: 0.0 ; 0.0 : 0.0

○ ○ ○ ●

B 115 int Vertical velocity component for flap wavemaker kinematics

0 OFF

1 ON

default: 0

● ● ● ●

B 116 int Flap wavemaker kinematics input as paddle motion or angle

1 $X(t)[m]$

2 $\beta(t)[rad]$

default: 1

● ● ● ●

B 117 double Starting time shift for wavemaker kinematics input

default: 0.0

● ● ● ●

B 120 double phase angle shift for periodic regular waves

default: -90°

○ ○ ○ ●

B 123 double Z-coordinate for active wave absorption hinge location

default: 0.0

○ ○ ○ ●

B 125 double y-coordinate for hydrodynamic coupling slice selection for HDC input

default: na

○ ○ ○ ●

B 127 int remove y-component from hydrodynamic coupling

0 OFF

1 ON

default: 0

● ● ● ●

B 130 int Enable directional spreading for irregular waves

0 OFF

1 Cosine squared

2 Mitsuyasu

default: 0

● ● ● ●

B 131 double Main direction for multidirectional irregular waves

default: 0.0°

● ● ● ●

B 132 double Start and end angle for directional spreading

default: -90.0° ; 90.0°

● ● ● ●

B 133 int Number of angle intervals for directional spreading

default: 1

● ● ● ●

B 136 int Type of multi-directional irregular wave calculation

1 DSM - double summation method

2 PSSM - Pascal's Single Summation Method

3 YSSM - Yu's Single Summation Method

4 EEM - equal energy method, to be used together with B 84 2

default: 1

● ● ● ●

B 138 int Seed number 1 and int Seed number 2 for the random phase generation of multi-directional irregular waves

If not given, the random numbers are based on the computer clock.

default: na

● ● ● ●

B 139 int Seed number for the random phase generation of irregular waves

IMPORTANT: seed numbers are not machine transferable. In order to reliably repeat a given wave spectrum, the wave component file (see REEF3D_Log-Wave folder) needs to be re-used and renamed to 'waverecon.dat'.

0 time based seed number

>**0** input based seed number

default: 0

● ○ ○ ○

B 160 int Number of vertical layers for obtaining depth-averaged velocities in the wave generation for REEF3D::SFLOW

default: 5

○ ● ○ ○

B 170 int Number of Fourier modes for calculating the steady surface gravity wave (B 92 70)

default: 1024

○ ○ ○ ●

B 181 double x-motion amplitude, frequency, phase change

default: 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

B 182 double y-motion amplitude, frequency, phase change

default: 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

B 183 double z-motion amplitude, frequency, phase change

default: 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

B 191 double angle for rotation around x-axis, rotation frequency, y-coordinate, z-coordinate

default: 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

B 192 double angle for rotation around y-axis, rotation frequency, x-coordinate, z-coordinate

default: 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

B 194 double start time rotation, end time rotation

default: 0.0 ; 0.0

○ ○ ○ ●

B 210 int patch boundary condition: inflow or outflow condition

1 inflow
2 outflow
default: 2

○ ○ ○ ●

B 240 * double Darcy porous media as rectangular box: $C, D, x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

B 241 int Porous media in x-direction for B240

0 OFF
1 ON
default: 1

○ ○ ○ ●

B 242 int Porous media in y-direction for B240

0 OFF
1 ON
default: 1

○ ○ ○ ●

B 243 int Porous media in z-direction for B240

0 OFF

1 ON

default: 1

○ ○ ○ ●

B 260 double c factor for VRANS porous media

default: 0.0

○ ○ ○ ●

B 264 double KC number VRANS porous media force source terms for the momentum equations

default: $1.0 \cdot 10^{20}$

○ ○ ○ ●

B 270 * double VRANS porous media as rectangular box $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}, n, d_{50}, \alpha, \beta$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ;

○ ○ ○ ●

B 274 * double VRANS porous media as vertical cylinder: $x_{center}, y_{center}, z_{start}, z_{end}, radius, n, d_{50}, \alpha, \beta$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ;

○ ○ ○ ●

B 281 * double VRANS porous media as wedge in x-direction: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}, n, d_{50}, \alpha, \beta$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ;

○ ○ ○ ●

B 282 * double VRANS porous media as wedge in y-direction: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}, n, d_{50}, \alpha, \beta$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ;

○ ○ ○ ●

B 291 * double VRANS porous media as plate in x-direction: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}, d_{thickness}, n, d_{50}, \alpha, \beta$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ;

○ ○ ○ ●

B 295 int VRANS source treatment for turbulence model

0 OFF

1 VRANS source terms

2 turn off turbulence inside porous media

default: 1

○ ○ ○ ●

B 308 int consider vegetation porosity effects on fluid acceleration

0 OFF

1 ON

default: 1

○ ○ ○ ●

B 309 double C_M for vegetation

default: 2.0

○ ○ ○ ●

B 310 * double VRANS vegetation as rectangular box. N is the number of cylinders per unit area, D the diameter of the cylinders and C_d the drag coefficient.

$x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}, N, D, C_d$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ;

○ ○ ○ ●

B 321 * double VRANS vegetation as wedge in x-direction. N is the number of cylinders per unit area, D the diameter of the cylinders and C_d the drag coefficient.

$x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}, N, D, C_d$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ;

○ ○ ○ ●

B 322 * double VRANS vegetation as wedge in y-direction. N is the number of cylinders per unit area, D the diameter of the cylinders and C_d the drag coefficient.

$x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}, N, D, C_d$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ;

● ○ ○ ●

B 411 * patchBC discharge int ID, double $discharge$

default: na

● ○ ○ ●

B 412 * patchBC pressure int ID, double $pressure$

default: na

● ○ ○ ●

B 413 * patchBC waterlevel int ID, double $waterlevel$

default: na

● ○ ○ ●

B 414 * patchBC velocity perpendicular to face int ID, double $velocity$

default: na

● ○ ○ ●

B 415 * patchBC velocity components int ID, double U, V, W

default: na ; na ; na

● ○ ○ ●

B 416 * patchBC horizontal inflow angle int ID, double α

default: na

● ○ ○ ●

B 418 int ID, int free stream outflow

0 OFF

1 ON

default: 0

● ○ ○ ●

B 421 int ID, int discharge hydrograph

Requires a file named e.g. 'hydrograph_Q_1.dat', where the number is the patchBC ID.

0 OFF

1 ON

default: 0

● ○ ○ ●

B 422 int ID, int waterlevel hydrograph

Requires a file named e.g. 'hydrograph_FSF_1.dat', where the number is the patchBC ID.

0 OFF

1 ON

default: 0

● ○ ○ ●

B 440 * patchBC as line:

int $ID, face$, double $x_{start}, x_{end}, y_{start}, y_{end}$

The patch boundary condition will convert solid boundaries to inflow or outflow conditions. Patches are addressed with the ID, with which it is possible to prescribe different flow conditions (see B 411 - B 418). When flow values are not prescribed, the default are zero-gradient boundary conditions.

default: 0 ; 0; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

B 441 * patchBC as rectangular box:

int $ID, face$, double $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

The patch boundary condition will convert solid boundaries to inflow or outflow conditions. Patches are addressed with the ID, with which it is possible to prescribe different flow conditions (see B 411 - B 417).

default: 0 ; 0; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

B 442 * patchBC as circle: int $ID, face$, double$x_M, y_M, z_M, radius$

The patch boundary condition will convert solid boundaries to inflow or outflow conditions. Patches are addressed with the ID, with which it is possible to prescribe different flow conditions (see B 411 - B 417).

default: 0 ; 0; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

3.3 C :: Concentration

Valid for:
S F N C
○ ○ ○ ●

C 1 double Additional density from concentration in phase 1

default: 10.0

○ ○ ○ ●

C 2 double Additional viscosity from concentration in phase 1

default: 0.0

○ ○ ○ ●

C 3 double Additional density from concentration in phase 2

default: 10.0

○ ○ ○ ●

C 4 double Additional viscosity from concentration in phase 2

default: 0.0

○ ○ ○ ●

C 5 double Additional viscosity from concentration in phase 2

default: 0.0

○ ○ ○ ●

C 10 int Time scheme concentration function

0 OFF
1 2nd-order Adams-Bashfort
2 2rd-order TVD Runge-Kutta
3 3rd-order TVD Runge-Kutta
default: 0

○ ○ ○ ●

C 15 int convection discretization for concentration function

1 FOU
2 CDS2
3 QUICK
4 WENO FLUX
5 WENO HJ
6 CDS4
10 High Resolution TVD scheme with Minmod limiter
11 High Resolution TVD scheme with van Leer limiter
12 High Resolution TVD scheme with Umist limiter
13 High Resolution TVD scheme with Sweby limiter
14 High Resolution TVD scheme with Superbee limiter
15 High Resolution scheme with Smart limiter
16 High Resolution TVD scheme with Limo3 limiter
42 High Resolution TVD scheme with Weller limiter
default: 0

○ ○ ○ ●

C 20 int Diffusion discretization

0 OFF

1 explicit

2 implicit (automatic for implicit convection discretization)

default: 0

○ ○ ○ ●

C 50 double Fill ratio concentration area 1 and 2

default: 1.0 ; 0.0

○ ○ ○ ●

C 51 double Area 1 start, x-direction

default: 0.0 m

○ ○ ○ ●

C 52 double Area 1 start, y-direction

default: 0.0 m

○ ○ ○ ●

C 53 double Area 1 start, z-direction

default: 0.0 m

○ ○ ○ ●

C 54 double Area 1 end, x-direction

default: $1.0 \cdot 10^7$ m

○ ○ ○ ●

C 55 double Area 1 end, y-direction

default: $1.0 \cdot 10^7$ m

○ ○ ○ ●

C 56 double Area 1 end, z-direction

default: $1.0 \cdot 10^7$ m

○ ○ ○ ●

C 57 double Area 1 as 3D-plane: $ax + by + cy + d = 0$

default: 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

C 58 double Area 1 as sphere, center-coordinates: x_0, y_0, z_0, and radius r

default: 0.0 ; 0.0 ; 0.0 ; 0.0

3.4 D :: Discretization

Here the discretization of the convection terms in the momentum equations is chosen. All methods are implemented in terms of conservative finite differences. The exception is WENO HJ, which is the non-conservative form of the WENO scheme.
Valid for:
S F N C
○ ○ ● ●

D 10 int convection discretization

0 OFF

1 FOU

2 CDS2

3 QUICK

4 WENO5 FLUX

5 WENO5 HJ

6 CDS4

7 WENO3 FLUX

8 WENO3 HJ

10 High Resolution TVD scheme with Minmod limiter

11 High Resolution TVD scheme with van Leer limiter

12 High Resolution TVD scheme with Umist limiter

13 High Resolution TVD scheme with Sweby limiter

14 High Resolution TVD scheme with Superbee limiter

15 High Resolution scheme with Smart limiter

16 High Resolution TVD scheme with Limo3 limiter

42 High Resolution TVD scheme with Weller limiter

default: 4

○ ○ ● ●

D 11 int convection velocities for momentum equations

0 OFF

1 FOU

2 CDS2

default: 2

○ ○ ● ●

D 20 int diffusion discretization

Selecting the implicit treatment of the diffusion term in the momentum equations for explicit time stepping has the advantage of removing diffusion from the CFL criterion, which determines the time step size. As the diffusion term enters the CFL criterion in the order of $1/dx^2$, effectively larger times steps for D20 2 will be used on especially finer grids. When RANS turbulence models are used, the effective diffusion can be several magnitudes higher than the molecular diffusion. Then implicit diffusion treatment will increase the time step significantly.

0 OFF

1 explicit

2 implicit

default: 2

○ ○ ● ●

D 21 int print out diffusion solver time and iteration for implicit diffusion and explicit convection discretization

0 OFF

1 ON

default: 0

○ ○ ○ ●

D 30 int pressure algorithm The projection method (PJM) works only together with explicit time advancement schemes, while all SIMPLE type methods work with implicit timestepping.

0 OFF

1 PJM CORR

default: 1

3.5 F :: Free Surface

Valid for:
S F N C
○ ○ ○ ●

F 10 int free surface mode for interface capturing

1 One-Phase

2 Two-Phase

default: 2

○ ○ ○ ●

F 30 int time scheme level set

0 OFF

1 2nd-order Adams-Bashfort

2 2rd-order TVD Runge-Kutta

3 3rd-order TVD Runge-Kutta

default: 3

○ ○ ○ ●

F 31 int particle Level Set

0 OFF

1 PLS with tri-linear interpolation

2 PLS with tri-cubic interpolation

default: 0

○ ○ ○ ●

F 32 int number of particles per cell

default: 64

○ ○ ○ ●

F 33 double factor for particle array allocation

default: 0.4

○ ○ ○ ●

F 34 int Printout iteration of particles

default: 1000

○ ○ ○ ●

F 35 int convection discretization for level set

1 FOU
2 CDS2
3 QUICK
4 WENO5 FLUX
5 WENO5 HJ
6 CDS4
7 WENO3 FLUX
8 WENO3 HJ
10 High Resolution TVD scheme with Minmod limiter
11 High Resolution TVD scheme with van Leer limiter
12 High Resolution TVD scheme with Umist limiter
13 High Resolution TVD scheme with Sweby limiter
14 High Resolution TVD scheme with Superbee limiter
15 High Resolution scheme with Smart limiter
16 High Resolution TVD scheme with Limo3 limiter
42 High Resolution TVD scheme with Weller limiter
default: 5

○ ○ ○ ●

F 39 double Relaxation factor for reini volume constraint

default: 0.5

○ ○ ○ ●

F 40 int Reinitialization time scheme

0 OFF

3 3rd-order TVD Runge-Kutta

11 Geometric reinitialization with 3rd-order TVD Runge-Kutta

default: 3

○ ○ ○ ●

F 42 double length for level set initial reinitialization, overrides maximum domain length

default: -1.0

○ ○ ○ ●

F 43 double Factor for reinitialization time step size

default: 0.55

○ ○ ○ ●

F 44 int Number of Reinitialization time steps

default: 3

○ ○ ○ ●

F 45 double Factor for the calculation of the interface thickness ϵ

default: 2.1

○ ○ ○ ●

F 46 int Type of Picard iterations after reinitialization

0 OFF

1 Volume correction for reinitialization step using volume from previous time step

2 Volume correction for level set and reinitialization step using volume from previous time step

3 Volume correction for level set and reinitialization step using total volume balance

default: 0

○ ○ ○ ●

F 47 int Number of Picard iterations after reinitialization

default: 0

○ ○ ○ ●

F 49 int Reinitialization for interface nodes

0 OFF

1 ON

default: 1

○ ○ ○ ●

F 50 int Fix Level Set for inflow or outflow

1 Inflow fixed

2 Outflow fixed

3 Fix Both

4 Fix None

default: 2

○ ○ ○ ●

F 51 double Phase 1 start, x-direction

default: 0.0 m

○ ○ ○ ●

F 52 double Phase 1 start, y-direction

default: 0.0 m

○ ○ ○ ●

F 53 double Phase 1 start, z-direction

default: 0.0 m

○ ○ ○ ●

F 54 double Phase 1 end, x-direction

default: $1.0 \cdot 10^7$ m

○ ○ ○ ●

F 55 double Phase 1 end, y-direction

default: $1.0 \cdot 10^7$ m

○ ○ ○ ●

F 56 double Phase 1 end, z-direction

default: $1.0 \cdot 10^7$ m

○ ○ ○ ●

F 57 double Interface as 3D-plane: $ax + by + cy + d = 0$

default: 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

F 58 double Interface as sphere, center-coordinates: x_m, y_m, z_m, and radius r

default: 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

F 59 double Interface as vertical cylinder, center-coordinates: $x_m, y_m, z_{start}, z_{end}$ and radius r

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

● ● ● ●

F 60 double initial still water level for the whole domain

default: $-1.0 \cdot 10^2 0$ m

○ ○ ○ ●

F 61 double Phase 1 end, z-direction for inflow boundary

default: $-1.0 \cdot 10^2 0$ m

● ○ ● ●

F 62 double Phase 1 end, z-direction for outflow boundary

default: $-1.0 \cdot 10^2 0$ m

○ ○ ○ ●

F 63 double x_{start} for level set interpolation

default: $-1.0 \cdot 10^{2}0\ m$

○ ○ ● ●

F 64 int number of iterations for outflow water level ramp up (from F 60 to F 62)

default: 0

○ ○ ○ ●

F 70 double* Phase 1 ini as rectangular box: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

F 71 double* Phase 2 ini as rectangular box: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

F 72 double* Regions for free surface elevation $x_{start}, x_{end}, y_{start}, y_{end}, h_{waterlevel}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

F 80 int time scheme VOF Currently, the VOF implementation is in beta and for testing purposes only. The algorithms for open channel flow and wave generation only work for the level set method.

1 2nd-order Adams-Bashfort

3 3rd-order TVD Runge-Kutta

default: 0

○ ○ ○ ●

F 84 double c_α factor for VOF compression

default: 1.0

○ ○ ○ ●

F 85 int convection discretization VOF Recommended scheme for VOF convection is HRIC modified.

1 FOU

2 CDS2

3 QUICK

4 WENO FLUX

5 WENO HJ

6 CDS4

10 High Resolution TVD scheme with Minmod limiter

11 High Resolution TVD scheme with van Leer limiter

12 High Resolution TVD scheme with Umist limiter

13 High Resolution TVD scheme with Sweby limiter

14 High Resolution TVD scheme with Superbee limiter

15 High Resolution scheme with Smart limiter

16 High Resolution TVD scheme with Limo3 limiter

42 High Resolution TVD scheme with Weller limiter

51 HRIC

52 HRIC modified

53 CICSAM

default: 0

○ ○ ○ ●

F 150 int Benchmarks This will initialize the level set funciton and the velocity field according to the benchmark cases

0 OFF

1 Vortex

2 Slotted disc

3 3D Deformation

4 3D Taylor-Green vortex

default: 0

3.6 H :: Heat Transfer

Valid for:
S F N C
○ ○ ○ ●

H 1 double Thermal Diffusivity Phase 1

default: $1.4 \cdot 10^{-7}$ m^2/s (water)

○ ○ ○ ●

H 2 double Thermal Diffusivity Phase 2

default: $2.216 \cdot 10^{-5}$ m^2/s (air)

○ ○ ○ ●

H 3 int Fluid property calculation

1 interpolation for water and air (density and viscosity)
2 generic Boussinesq Approximation (density)
default: 1

○ ○ ○ ●

H 4 double Thermal expansion coefficients double β_1 and β_2

default: 0.0 ; 0.0

○ ○ ○ ●

H 9 int Switch Temperature 1 and temperature 2 fluid for the two phases

1 1: water 2: air
2 1: air 2: water
default: 1

○ ○ ○ ●

H 10 int Time Scheme Heat Transfer

0 OFF
1 2nd-order Adams-Bashfort
2 2nd-order TVD Runge-Kutta

3 3rd-order TVD Runge-Kutta

default: 0

○ ○ ● ●

H 15 int convection discretization for heat transfer

0 OFF

1 FOU

2 CDS2

3 QUICK

4 WENO5 FLUX

5 WENO5 HJ

6 CDS4

7 WENO3 FLUX

8 WENO3 HJ

10 High Resolution TVD scheme with Minmod limiter

11 High Resolution TVD scheme with van Leer limiter

12 High Resolution TVD scheme with Umist limiter

13 High Resolution TVD scheme with Sweby limiter

14 High Resolution TVD scheme with Superbee limiter

15 High Resolution scheme with Smart limiter

16 High Resolution TVD scheme with Limo3 limiter

42 High Resolution TVD scheme with Weller limiter

default: 5

○ ○ ○ ●

H 50 double Temperature 1 and Temperature 2

default: 0.0 C° ; 0.0 C°

○ ○ ○ ●

H 51 double Temperature 1 start, x-direction

default: 0.0 m

○ ○ ○ ●

H 52 double Temperature 1 start, y-direction

default: 0.0 m

○ ○ ○ ●

H 53 double Temperature 1 start, z-direction

default: 0.0 m

○ ○ ○ ●

H 54 double Temperature 1 end, x-direction

default: $1.0 \cdot 10^7$ m

○ ○ ○ ●

H 55 double Temperature 1 end, y-direction

default: $1.0 \cdot 10^7$ m

○ ○ ○ ●

H 56 double Temperature 1 end, z-direction

default: $1.0 \cdot 10^7$ m

○ ○ ○ ●

H 57 double Interface as 3D-plane: $ax + by + cy + d = 0$

default: 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

H 58 double Interface as sphere, center-coordinates: x_0, y_0, z_0, and radius r

default: 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

H 61 double Temperature boundary condition on surface 1

default: na $C°$

○ ○ ○ ●

H 62 double Temperature boundary condition on surface 2

default: na C°

○ ○ ○ ●

H 63 double Temperature boundary condition on surface 3

default: na C°

○ ○ ○ ●

H 64 double Temperature boundary condition on surface 4

default: na C°

○ ○ ○ ●

H 65 double Temperature boundary condition on surface 5

default: na C°

○ ○ ○ ●

H 66 double Temperature boundary condition on surface 6

default: na C°

3.7 I :: Initialization

Valid for:
S F N C
○ ○ ○ ●

I 10 int Initialize Everything Turning this parameter on will invoke I11, I12 and I13.

0 OFF
1 ON
2 ON (include free surface for potential flow ini)
default: 0

○ ○ ○ ●

I 11 int Initialize Velocities with Potential Flow Solver

0 OFF
1 ON
2 ON (include free surface)
default: 0

○ ○ ○ ●

I 12 int Initialize Pressure

0 OFF
1 Hydrostatic based on vertical coordinate
2 Hydrostatic based on level set values
default: 0

○ ○ ○ ●

I 13 int Initialize Turbulence Model

0 OFF
1 ON
default: 0

○ ○ ○ ●

I 21 int Phase 2 velocities after potential flow initialization

0 use potential solver result
1 set to zero
default: 0

● ● ● ●

I 30 int Full initialization of Numerical Wave Tank

0 OFF
1 ON
default: 0

○ ○ ○ ●

I 40 int Full initialization from state file (hotstart) See option P 40 for state file print out.

0 OFF

1 ON

default: 0

○ ○ ○ ●

I 41 int Number of state file iteration (hotstart)

default: 0

○ ● ○ ○

I 44 int Read FNPF state file with 3D velocity potential

0 OFF

1 ON

default: 0

● ● ● ●

I 50 int Simtime start

default: 0.0

○ ○ ○ ●

I 56 int pressure above F56 set to zero

default: 0

○ ○ ○ ●

I 58 double Initialize Vertical Velocity around Sphere from F 58, double radius for initialization

default: 0.0 ; 0.0

○ ○ ○ ●

I 230 int Read FlowFiles for inflow boundary condition. The number gives the ID of the FlowFile gage. For iowave activated, B 98 should be set to 0.

default: 0

○ ○ ○ ●

I 231 double Vertical offset of FlowFile inflow.

default: 0.0

○ ○ ○ ●

I 232 double Time offset of FlowFile inflow.

default: 0.0

○ ○ ○ ●

I 241 double Delta t for FlowFile inflow and hydrodynamic coupling.

default: 0.0

3.8 M :: MPI

Valid for:
S F N C
● ● ● ●

M 10 int Number of processors for parallel computations This value needs to be consistent with the grid generation and the console input for starting REEF3D through the mpirun command.

default: 0

3.9 N :: Numerical Options

Valid for:
S F N C
○ ● ● ●

N 10 int Iterative solver for the Poisson equation The default BiCGStab solver uses Jacobi Scaling as the default preconditioner, which is implemented in the compressed diagonal storage format (CDS). Jacobi Scaling is currently the most effective preconditioner for the Poissson equation with a jump in the matrix coefficients, as is the case for multiphase flow. SIP is an optimized implementation of ILU.

The fastest combination of HYPRE solvers is the BiCGSTAB struct solver with PFMG preconditioning. From the HYPER AIJ solvers, the combination PCG + BoomerAMG is fastest. For more information on the HYPRE solvers and preconditioners, have a look at the HYPRE user manual Cen [2015].

3 BiCGStab with Jacobi Preconditioning (internal)

11 PCG (HYPRE Struct)

12 GMRES (HYPRE Struct)

13 LGMRES (HYPRE Struct)

14 BiCGStab (HYPRE Struct)

15 Hybrid-PCG (HYPRE Struct)

16 Hybrid-GMRES (HYPRE Struct)

17 Hybrid-BiCGStab (HYPRE Struct)

18 PFMG (HYPRE Struct, Geometric Multigrid)

19 SMG (HYPRE Struct, Geometric Mulitgrid)

21 PCG (HYPRE AIJ)

22 GMRES (HYPRE AIJ)

23 LGMRES (HYPRE AIJ)

24 BiCGStab (HYPRE AIJ)

25 AMG (HYPRE AIJ, BoomerAMG)

31 PCG (HYPRE SStruct)

32 GMRES (HYPRE SStruct)

33 LGMRES (HYPRE SStruct)

34 BiCGStab (HYPRE SStruct)

35 Hybrid-PCG (HYPRE SStruct)

36 Hybrid-GMRES (HYPRE SStruct)

37 Hybrid-BiCGStab (HYPRE SStruct)

38 PFMG (HYPRE SStruct, Geometric Multigrid)

39 SMG (HYPRE SStruct, Geometric Mulitgrid)

default: 14

○ ● ● ●

N 11 int Preconditioner for the Poisson Equations Keep in mind to use only options, that are compatible with N 10.

0 OFF

11 PFMG (for HYPRE Struct Krylov solvers)

12 SMG (for HYPRE Struct Krylov solvers)

21 AMG (for HYPRE AIJ Krylov solvers, BoomerAMG)

31 PFMG (for HYPRE SStruct Krylov solvers)

32 SMG (for HYPRE SStruct Krylov solvers)

default: 11

○ ○ ○ ●

N 40 int Time scheme for the momentum equations

1 2nd-order Adams-Bashforth

2 2rd-order TVD Runge-Kutta

3 3rd-order TVD Runge-Kutta

4 3rd-order TVD Runge-Kutta low storage

22 2rd-order momentum-free surface coupled TVD Runge-Kutta

23 3rd-order momentum-free surface coupled TVD Runge-Kutta

default: 3

● ● ● ●

N 41 double Maximum modeled time Out of N41 and N45, whichever criterion is fulfilled first will finalize the simulations.

default: $1.0 \cdot 10^{19}$

● ● ● ●

N 44 double Stopping criteria iterative solver

default: $1.0 \cdot 10^{-8}$

● ● ● ●

N 45 int Maximum number of outer iterations Out of N41 and N45, whichever criterion is fulfilled first will finalize the simulations

default: $1 \cdot 10^{8}$

● ● ● ●

N 46 int Maximum number of solver iterations

default: 250

● ● ● ●

N 47 double Relaxation factor for time step size This factor is used when determining the time step size based on the CFL criterion for adaptive timestepping.

default: 0.3

● ● ● ●

N 48 int Adaptive timestepping

0 OFF

1 ON

2 ON, using velocity around the interface

default: 1

● ● ● ●

N 49 double Timestep size for fixed timestepping

default: 1.0 sec

● ● ● ●

N 50 int Adaptive time stepping method

1 standard

2 component wise (less conservative)

default: 1

● ● ● ●

N 61 double Stopping criterion for critical velocities. The code will exit and write a final .vtu paraview file

default: 500.0

3.10 P :: Printer

Valid for:
S F N C
● ● ● ●

P 10 int Print ParaView binary format

0 OFF

1 ON

default: 1

○ ○ ○ ●

P 11 int Log print frequency

default: 10

● ● ● ●

P 12 int Terminal print frequency

default: 1

● ● ● ●

P 15 int Print out file numbering

1 print out based
2 iteration based
default: 1

○ ○ ○ ●

P 18 int Algorithm type for level set paraview print out

1 Standard
2 Node Fill
default: 2

● ● ● ●

P 20 int Print results every i^{th} iteration Choose between either P20, P30, P34 or P35.

default: -10

○ ● ○ ○

P 21 int Print time averaged velocity, pressure and temperature (in case of heat transfer)

0 OFF
1 ON
default: 0

○ ● ○ ○

P 22 double Start averaging after transients mean values

default: 0.0

● ● ● ●

P 23 int Print test array vtu file

0 OFF

1 ON

default: 0

○ ○ ○ ●

P 24 int Print density to vtu file

0 OFF

1 ON

default: 0

○ ○ ● ●

P 25 int Print solid to vtu file

0 OFF

1 ON

default: 0

○ ○ ○ ●

P 26 int Print cbed and conc to vtu file

0 OFF

1 ON

default: 0

○ ○ ○ ●

P 27 int Print topo to vtu file

0 OFF

1 ON

default: 0

○ ○ ● ●

P 28 int Print floating body level set function to vtu file

0 OFF

1 ON

default: 0

○ ○ ○ ●

P 29 int Print walldist to vtu file

0 OFF

1 ON

default: 0

○ ○ ○ ●

P 30 double Print Paraview results every i^{th} second Choose between either P20, P30, P34 or P35.

default: -1.0

○ ○ ○ ●

P 34 double Print Paraview results every i^{th} second based on sediment transport time Choose between either P20, P30, P34 or P35.

default: -1.0

○ ● ● ●

P 35 * double Start print t, double End t, double results every i^{th} second

Choose between either P20, P30, P34 or P35.

default: 0.0 ; 0.0 ; 0.0

● ● ○ ●

P 40 int Print state file for hotstart, hydrodynamic coupling (SFLOW to CFD and FNPF to CFD) or grid nesting.
For hotstart functionality, see I 40 for full initialization from state file.
For hydrodynamic coupling, see

The CFD state file contains the velocities, pressure, level set function, k, eddy viscosity, ϵ or ω, topo, cbed and conc.

0 OFF

1 print into one file

2 print into consecutive file

default: 0

○ ● ○ ●

P 41 int Print state file every ith Iteration

default: 1

○ ● ○ ●

P 42 double Print state file every ith second

default: na

○ ● ○ ○

P 43 Define print out area for FNPF state files (optional); double x_{start}, double x_{end}, double y_{start}, double y_{end}

default: na ; na ; na ; na

○ ● ○ ○

P 44 int Print 3D velocity potential to sate file

0 OFF

1 ON

default: 0

○ ● ○ ○

P 45 int State file type

1 single file per time step

2 continuous file

default: 2

○ ● ● ●

P 46 double Start print t, int End t, double state files every i^{th} iteration

default: 0 ; 0

○ ● ● ●

P 47 double Start print t, double End t, double state files every i^{th} second

default: 0.0 ; 0.0

● ● ● ●

P 50 * double x-location and double y-location of height gauges for wave theory

default: na ; na

● ● ● ●

P 51 * double x-location and double y-location of height gauges

default: na ; na

● ● ● ●

P 52 * double y-location of water surface line in x-direction

default: na

● ● ● ●

P 53 int for NWT: add theoretical wave to wsfline file

0 OFF
1 ON
default: 0

● ● ● ●

P 54 int Print wsfline files every i^{th} iteration

default: 10

● ● ● ●

P 55 double Print wsfline files every ith second

default: -1.0

● ● ● ●

P 56 * double x-location of water surface line in y-direction

default: na

○ ○ ○ ●

P 58 * double x-location, double y-location and double duration in *sec* for wave timeries

default: 0.0 ; 0.0 ; 12800

○ ● ○ ○

P 59 int for FNPF: print out breaking wave Log

0 OFF
1 ON
default: 0

○ ○ ○ ●

P 61 * double x-location, double y-location and double z-location of point probes

default: na

○ ○ ○ ●

P 62 * double $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$ of line probes

default: na

● ○ ○ ○

P 63 * double x-location and double y-location of depth averaged point probes

default: na

○ ○ ○ ●

P 64 * double x-location, double y-location and double z-location of pressure probes

default: na

○ ● ● ●

P 65 * double x-location, double y-location and double z-location of velocity probes

default: na

○ ● ● ●

P 66 * double x-location, double y-location and double z-location of velocity probes from wave theory

default: na

○ ○ ○ ●

P 71 int print out viscosity to vtu file

0 OFF

1 ON

default: 0

○ ○ ○ ●

P 72 int print out VOF function to vtu file

0 OFF

1 ON

default: 0

○ ○ ○ ●

P 75 int print out vorticity to vtu file

0 OFF

1 ON

default: 0

○ ○ ○ ●

P 76 int print out sediment bedload to vtu file: q_{be}, q_b

0 OFF

1 ON

default: 0

○ ○ ○ ●

P 77 int print out bedslope sediment parameters to vtu file: $\alpha,\beta,\theta,\gamma,\phi$

0 OFF

1 ON

default: 0

○ ○ ○ ●

P 78 int print out sediment parameters to vtu file: dh, bedchange, reduce, threshold, slideflag

0 OFF

1 ON

default: 0

○ ○ ○ ●

P 79 int print out bed shear stress to vtu file

0 OFF

1 bed shear stress

2 shear velocity

3 shields paramters

default: 0

○ ● ○ ○

P 80 int Print P 85 and P 88 results every ith iteration

default: 1

○ ○ ○ ●

P 81 double force calculation box: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ● ● ○

P 85 * double Morison force calculation using ALE : $x_{centre}, y_{centre}, radius, C_d, C_m$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ● ○ ○

P 88 * double Print wave kinematics files x_{centre}, y_{centre}

default: 0.0 ; 0.0

○ ○ ○ ●

P 92 int Calculate force from water or water and air combined

0 force only from water
1 force from water and air combined
default: 0

○ ○ ○ ●

P 101 int Print sloshing force and moment file

0 OFF
1 ON
default: 0

○ ● ○ ○

P 110 int Print significant wave height

0 OFF
1 ON
default: 0

○ ● ○ ○

P 111 double Start averaging after transients for significant wave height calculation

default: 0.0

● ● ● ●

P 120 int Print sediment files every i^{th} iteration

default: 1

● ○ ○ ●

P 121 double x-location and double y-location of bed level gauges

default: na

○ ○ ○ ●

P 122 int Maximum bedchange gauge

0 OFF
1 ON
default: 0

● ○ ○ ●

P 123 * double y-location of sediment profile in x-direction

default: na

● ○ ○ ●

P 124 * double x-location of sediment profile in y-direction

default: na

○ ○ ○ ●

P 125 double x-location and double y-location of bed shear stress gauges

default: na

○ ○ ○ ●

P 126 int Maximum bedshear stress print out

0 OFF
1 ON
default: 0

○ ○ ● ○

P 131 int maximum wetdry (or runup) into vtp file

0 OFF
1 ON

default: 0

○ ○ ● ○

P 133 * double y-location of maximum runup gage in x-direction cross section

default: na

○ ○ ○ ●

P 150 int Print data from DIVEMesh interpolation

0 OFF
1 ON
default: 0

○ ○ ○ ●

P 151 int Type of data

1 scalar
2 signed distance function
default: 1

○ ○ ○ ●

P 152 int Type of boundary condition for data array

1 x-velocity
2 y-velocity
3 z-velocity
4 scalar with Neuman boundary conditions
default: 1

○ ○ ○ ●

P 166 int print out discharge cross section to console

0 OFF
1 ON
default: 0

○ ○ ○ ●

P 167 * double x-location of discharge in cross section

default: na

○ ○ ○ ●

P 168 * double x-location, z_{start} and z_{end} of discharge window in cross section

default: na ; na ; na

○ ○ ○ ●

P 180 int Print free surface vtp file

0 OFF
1 ON
default: 0

○ ○ ○ ●

P 181 double i^{th} iteration for fsf print out

default: 10

○ ○ ○ ●

P 182 double Print out fsf every i^{th} second

default: -1.0

○ ○ ○ ●

P 184 * int Start iteration, int End iteration, int fsf print results every ith iteration
Choose between either P181, P182, P184 or P185.

default: 0 ; 0 ; 0

○ ○ ○ ●

P 185 * double Start t, double End t, double fsf print results every ith second
Choose between either P181, P182, P184 or P185.

default: 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

P 190 int Print topo vtp file

0 OFF

1 ON

default: 0

○ ○ ○ ●

P 191 double i^{th} iteration for topo print out

default: 10

○ ○ ○ ●

P 192 double Print out topo every i^{th} second

default: -1.0

○ ○ ○ ●

P 194 * int Start iteration, int End iteration, int topo print results every i^{th} iteration

Choose between either P191, P192, P194 or P195.

default: 0 ; 0 ; 0

○ ○ ○ ●

P 195 * double Start t, double End t, double topo print results every i^{th} second

Choose between either P191, P192, P194 or P195.

default: 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

P 230 double * x-coordinate for FlowFile printing from 2D CFD simulations

default: 0

○ ○ ○ ●

P 240 double * x-coordinate for PotentialFile printing from 2D FNPF simulations

default: 0

3.11 S :: Sediment

Valid for:
S F N C
● ○ ○ ●

S 10 int Sediment Transport Module

0 OFF
1 ON (impermeable sediments)
2 ON (porous sediment)
default: 0

○ ○ ○ ●

S 11 int Bedload Transport Formula

0 OFF
1 van Rijn
2 Meyer-Peter Müller
3 Engelund and Fredsøe
default: 0

○ ○ ○ ●

S 12 int Suspended transport formula

0 OFF
1 van Rijn
default: 0

○ ○ ○ ●

S 13 double Maximum time step size sediment transport

default: 10.0 *sec*

○ ○ ○ ●

S 14 double Relaxation factor for time step size sediment transport The timestep for the morphodyanmic calculations is by default decoupled from the hydrodynamics timestep (see S15), which typically is determined through adaptive timestepping (N48). The morphodynamic timestep is determined by the Courant criterion by analyzing the rate of bed elevation change.

default: 0.3

○ ○ ○ ●

S 15 int Sediment timestep selection

0 adaptive from S13 and S14

1 from flow solver

2 fixed from S13

default: 0

○ ○ ○ ●

S 16 int Bed shear stress formulation

1 wall function/velocity based

2 friction coefficient/velocity based

3 velocity based

4 turbulent kinetic energy based

7 depth-averaged velocities, based on Chezy-formula

default: 1

○ ○ ○ ●

S 19 double Maximum modeled time for sediment transport

default: $1.0 \cdot 10^{19}$

○ ○ ○ ●

S 20 double Sediment d_{50}

default: 0.001 m

○ ○ ○ ●

S 21 double Factor for d_{50} in calculation of k_s in bedshear routine

default: 3.0

○ ○ ○ ●

S 22 double Sediment density

default: 2650.0 kg

○ ○ ○ ●

S 23 double Sediment settling velocity m/s

default: automatic calculation

○ ○ ○ ●

S 24 double Porosity of the sediment layer

default: 0.5

○ ○ ○ ●

S 30 double Shields parameter

default: 0.047

○ ○ ○ ●

S 31 int Type of Exner formulation

1 v1
2 v2
default: 1

○ ○ ● ●

S 32 int Exner equation discretization

1 FOU
2 CDS2
4 WENO5 FLUX
5 WENO5 HJ
default: 4

○ ○ ○ ●

S 34 int Type suspended load bed change calculation

1 bedload-suspended load interface
2 flow depth integration

default: 1

○ ○ ○ ●

S 37 int Number of topo reinitialization time steps The mobile sediment bed is represented by a level set method. After erosion or deposition has taken place, it needs to be reinitialized in order keep its signed distance properties.

default: 2

○ ○ ○ ●

S 41 int Type of sediment start criterion

1 iterations

2 flow simulation time

3 t/T

default: 1

○ ○ ○ ●

S 42 int Type of sediment interval criterion

1 iterations

2 flow simulation time

3 t/T

default: 1

○ ○ ○ ●

S 43 int Number of water iterations, before sediment transport starts

default: 1000

○ ○ ○ ●

S 44 int Number of water iterations, between bed calculations

default: 10

○ ○ ○ ●

S 45 double Flow simulation time, before sediment transport starts

default: 1.0

○ ○ ○ ●

S 46 double Flow simulation time between bed calculation

default: 1.0

○ ○ ○ ●

S 47 double t/T, before sediment transport starts

default: 1.0

○ ○ ○ ●

S 48 double t/T between bed calculation

default: 1.0

○ ○ ○ ●

S 50 int Fix topo level set for inflow or outflow

1 Inflow fixed
2 Outflow fixed
3 Fix Both
4 Fix None
default: 3

○ ○ ○ ●

S 57 double Sediment end, z-direction for whole domain

default: $-1.0 \cdot 10^{2}0\ m$

○ ○ ○ ●

S 60 int Time scheme suspended sediments

3 3rd-order TVD Runge-Kutta
11 1st-order Euler Implicit
default: 0

○ ○ ○ ●

S 71 double Start of erosion in x-direction

default: -1.0e20

○ ○ ○ ●

S 72 double End of erosion in x-direction

default: 1.0e20

○ ○ ○ ●

S 73 * double value, double distance, double line angle, double line x-origin, double line y-origin; for use of relaxation method for inhibiting sediment transport in the relaxation zone and smoothly transitions to the active bed.

This method is preferred over S 71 and S72 because of the smooth moderation of the sediment bed. It gives also more flexibility as multiple relaxation lines with variable orientation can be chosen.

default: na

○ ○ ○ ●

S 77 double Active sediment transport algorithm region in x-direction: x_{start} and x_{end}

default: -1.0e20 ; +1.0e20

○ ○ ○ ●

S 80 int Type of critical shear stress reduction for sloping bed

0 off

1 Parker and Kovacs

2 Dey empirical

3 Dey analytical

4 Fredsøe

default: 0

○ ○ ○ ●

S 81 double Angle of repose (midphi parameter)

default: 35.0

○ ○ ○ ●

S 82 double Deltaphi parameter

default: 5.0

○ ○ ○ ●

S 83 int Type of bedslope calculation

2 CDS
5 WENO
default: 2

○ ○ ○ ●

S 84 int Type of limiter for critical bed shear stress reduction

1 Fredsøe
2 function based
default: 1

○ ○ ○ ●

S 85 int Bed load direction correction using the method proposed by Koch and Flokstra (1980)

0 OFF
1 ON
default: 1

○ ○ ○ ●

S 90 int Sandslide algorithm

0 off
1 Burkow geometrisch
2 Burkow geometrisch mod
3 Wu
4 Burkow PDE
default: 0

○ ○ ○ ●

S 91 int Number of sandslide iterations

default: 1

○ ○ ○ ●

S 93 double Delta angle for sandslide correction

default: 0.0°

○ ○ ○ ●

S 100 int Number of outer spatial filter iterations for the sediment bed for the predictor step (larger numbers increase the filter effect, recommended value: 1)

default: 0

○ ○ ○ ●

S 101 int Number of inner spatial filter iterations for the sediment bed for the corrector step (larger numbers move the bed closer to the uncorrected values, recommended value: between 1 and 5)

default: 0

3.12 T :: Turbulence

Valid for:
S F N C
○ ○ ○ ●

T 10 int Turbulence Model

0 OFF

1 k-ϵ model

2 k-ω model

12 EARSM based on k-ω model

21 URANS with k-ϵ model

22 URANS with k-ω model

31 LES with Smagorinsky SGS model

33 LES with WALE model

default: 0

○ ○ ○ ●

T 12 int convection discretization for turbulence model

0 OFF
1 FOU
5 WENO HJ
default: 5

○ ○ ○ ●

T 21 int Type of LES filter

0 box
1 f1
2 f2
default: 0

○ ○ ○ ●

T 31 double factor for eddy viscosity limiter in phase 1

default: 0.816

○ ○ ○ ●

T 32 double factor for eddy viscosity limiter in phase 2

default: 0.816

○ ○ ○ ●

T 35 double factor for eddy viscosity limiter near-wall

default: 0.212

○ ○ ○ ●

T 36 int FSF boundary condition for turbulent dissipation

0 OFF
1 ON
2 ON, including distance from the nearest wall
default: 0

○ ○ ○ ●

T 37 double Virtual origin for distance y' above the free surface

default: 0.07

○ ○ ○ ●

T 38 double ϵ_δ factor for width of free surface turbulence damping

default: 1.6

○ ○ ○ ●

T 41 int k-ω stabilization

0 OFF
1 ON
default: 0

○ ○ ○ ●

T 44 int Add buoyancy term to k-equation

0 OFF
1 ON
default: 0

3.13 W :: Water and Material Properties

Valid for:
S F N C
● ● ● ●

W 1 double Density Phase 1

default: 998.2 kg/m^3

● ● ● ●

W 2 double Viscosity Phase 1

default: $1.004 \cdot 10^{-6}$ m^2/s

○ ○ ○ ●

W 3 double Density Phase 2

default: 1.205 kg/m^3

○ ○ ○ ●

W 4 double Viscosity Phase 2

default: $1.41 \cdot 10^{-5}$ m^2/s

○ ○ ○ ●

W 5 double Surface tension

default: 0.0 N/m

● ○ ● ●

W 10 double Discharge

default: 0.0 m^3/s

○ ○ ○ ●

W 11 double Inflow velocities U,V,W on surface 1 (requires inflow boundary condition on surface 1 in DIVEMesh)

Options W 11 - W 16 are made for benchmark and other simpler test and validation cases. For more complex and real word cases, either B 60 1 (ioflow) or the B 400s (patchBC) options are recommended.

default: 0.0 0.0 0.0 m/s

○ ○ ○ ●

W 12 double Inflow velocities U,V,W on surface 2 (requires inflow boundary condition on surface 2 in DIVEMesh)

default: 0.0 0.0 0.0 m/s

○ ○ ○ ●

W 13 double Inflow velocities U,V,W on surface 3 (requires inflow boundary condition on surface 3 in DIVEMesh)

default: 0.0 0.0 0.0 m/s

○ ○ ○ ●

W 14 double Inflow velocities U,V,W on surface 4 (requires inflow boundary condition on surface 4 in DIVEMesh)

default: 0.0 0.0 0.0 m/s

○ ○ ○ ●

W 15 double Inflow velocities U,V,W on surface 5 (requires inflow boundary condition on surface 5 in DIVEMesh)

default: 0.0 0.0 0.0 m/s

○ ○ ○ ●

W 16 double Inflow velocities U,V,W on surface 6 (requires inflow boundary condition on surface 6 in DIVEMesh)

default: 0.0 0.0 0.0 m/s

● ● ● ●

W 20 double Acceleration due to gravity, x-component

default: 0.0 m/s^2

● ● ● ●

W 21 double Acceleration due to gravity, y-component

default: 0.0 m/s^2

● ● ● ●

W 22 double Acceleration due to Gravity, z-component

default: 0.0 m/s^2

○ ○ ○ ●

W 29 double Additional constant pressure gradient for periodic boundary conditions in x-,y- and z-direction

default: 0.0 0.0 0.0

○ ○ ○ ●

W 30 int Compressibility for phase 2, using ideal gas theory for constant temperature

0 OFF

1 ON

default: 0

○ ○ ○ ●

W 31 double Constant gas temperature with compressibility

default: 20.0 C^o

○ ○ ○ ●

W 41* double Velocity vertical line source phase 1: $x_{center}, y_{center}, z_{start}, z_{end}, velocity, \beta$

Here, β ist the direction of the flow.

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

● ○ ○ ●

W 90 int Non-Newtonian rheology

0 OFF

1 Herschel-Bulkley

● ○ ○ ●

W 95 double ν_0

default: 1.0

● ○ ○ ●

W 96 double τ_0

default: 1.0

● ○ ○ ●

W 97 double K Consistency index

default: 0.00001

○ ○ ○ ●

W 98 double n Power law index

default: 1.0

● ○ ○ ●

W 101 int Use Mohr-Coulomb

0 OFF
1 ON
default: 0

○ ○ ○ ●

W 102 double ϕ_0, c

default: 30.0 0.0

○ ○ ○ ●

W 103 double Transition factor for Mohr-Coulomb

default: 1.0

○ ○ ○ ●

W 110 int Numerical rreatment of rheology

1 use viscosity for rheology and yield stress
2 use viscosity for rheology and momentum source term for yield stress
3 use momentum source term for rheology and for yield stress
default: 1

○ ○ ○ ●

W 111 int Pressure for Mohr-Coulomb yield stress

1 hydrostatic
2 hydrostatic and dynamic pressure
3 hydrostatic pressure away from the interface and dynamic pressure closer to the interface (see W 112)
default: 1

○ ○ ○ ●

W 112 double Threshold factor for W 111 3

default: 2.1

3.14 X :: 6DOF

Valid for:
S F N C
● ○ ● ●

X 10 int Turn on 6DOF algorithm for floating bodies

0 OFF

1 Two-Way Coupling (CFD and NHFLOW)

2 One-Way Coupling (SFLOW, NHFLOW, CFD)

3 External pressure term (SFLOW and NHFLOW, requires X 400)

default: 0

○ ○ ● ●

X 11 int Turn the individual degrees of freedom of the floating body, the linear and the angular velocities: u, v, w, p, q, r
When degrees of freedom are chosen to be prescribed, then they can be prescribed with either fixed external linear or angular velocities or different kinds of motion files.

0 OFF

1 solver 6DOF equation

2 prescribed

default: 1 ; 1 ; 1 ; 1 ; 1 ; 1

○ ○ ○ ●

X 14 int Tangential velocity treatment for direct forcing

1 standard

2 reduced forcing

default: 1

● ○ ● ●

X 19 int print out interval for 6DOF log files

default: 1

○ ○ ● ●

X 21 double Floating body homogenous density ρ
Give either X 21 or X 22.

default: 900.0

○ ○ ● ●

X 22 double Floating body mass m
Give either X 21 or X 22.

default: 0.0

○ ○ ● ●

X 23 double Floating body center of gravity x_G, y_G, z_G
If not given, it will be calculated automatically. The center of gravity is also the origin of the non-inertial coordinate system.

default: 0.0 ; 0.0 ; 0.0

○ ○ ● ●

X 24 double Floating body moments of inertia I_x, I_y, I_z
If not given, it will be calculated automatically.

default: 0.0 ; 0.0 ; 0.0

○ ○ ● ●

X 25 double Damping coefficient for rotational motion around the three axes

default: 0.0 ; 0.0 ; 0.0

○ ○ ● ●

X 26 double Damping coefficient for translational motion along the three axes

default: 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

X 31 int Boundary conditions on floating solid for parallel velocities

1 slip

2 no-slip

3 floating body velocities

4 2nd-order extrapolation

default: 4

○ ○ ○ ●

X 33 int Boundary conditions on floating solid for pressure

1 Neumann

2 Extend

default: 1

○ ○ ○ ●

X 34 int Boundary treatment for new floating solid velocity

0 OFF

1 ON

default: 0

○ ○ ○ ●

X 41 double Thickness of continuous direct forcing layer.

default: 0.6

○ ○ ○ ●

X 45 int type of free surface level set convection discretization inside the floating body

0 convection inside floating body

1 no-slip wall v1

2 no-slip wall v2

3 no-slip wall v3

3 no-slip wall v4

default: 0

○ ○ ○ ●

X 46 int density smoothing inside fb

0 OFF

1 ON

default: 0

○ ○ ● ●

X 50 int Type of print out format for 6DOF structure

Since the VTP print out is binary, it is significantly faster than the STL print out. This becomes runtime relevant for STL files with a large amount of triangles.

1 VTP

2 STL

default: 1

○ ○ ● ●

X 60 int Type of hydrodynamic force calculation

1 STL

2 LSM

default: 0

○ ○ ● ●

X 101 double Initial Euler angles in [deg]: ϕ_0, θ_0, ψ_0
Positive angles rotate the body clockwise.

default: 0.0 ; 0.0 ; 0.0

○ ○ ● ●

X 102 double Initial linear floating body velocity: u_0, v_0, w_0

default: 0.0 ; 0.0 ; 0.0

○ ○ ● ●

X 103 double Initial angular floating body velocity: p_0, q_0, r_0

default: 0.0 ; 0.0 ; 0.0

● ○ ● ●

X 110 double Rectangular box floating body: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ● ●

X 131 double Cylinder in x-direction floating body: $radius, height, x_{center}, y_{center}, z_{center}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ;

○ ○ ● ●

X 132 double Cylinder in y-direction floating body: $radius, height, x_{center}, y_{center}, z_{center}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ;

● ○ ● ●

X 133 double Cylinder in z-direction floating body: $radius, height, x_{center}, y_{center}, z_{center}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ;

○ ○ ● ●

X 153 double Two-sided wedge in z-direction floating body: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ● ●

X 163* double wedge floating body, each of the 6 points is given by the coordinates x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4, x_5, y_5, z_5, x_6, y_6, z_6

default: [6x] 0.0 ; 0.0 ; 0.0

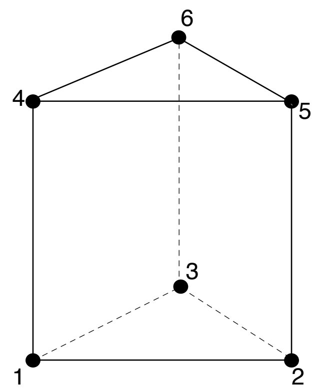


Figure 3.1: Definition of the wedge points for X 163

○ ○ ● ●

X 164 * double hexahedron floating body, each of the 8 points is given by the coordinates $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4, z_4, x_5, y_5, z_5, x_6, y_6, z_6, x_7, y_7, z_7, x_8, y_8, z_8$

default: [8x] 0.0 ; 0.0 ; 0.0

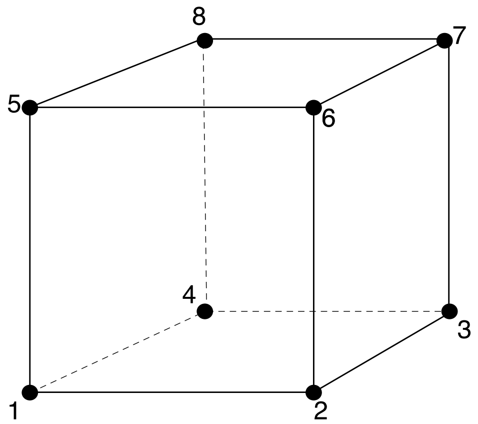


Figure 3.2: Definition of the hexahedron points for X 164

○ ○ ● ●

X 180 int Read STL file “floating.stl” for floating body geometry

0 OFF

1 ON

default: 1

○ ○ ● ●

X 181 double Scale STL geometry in x-, y-, and z-direction

default: 1.0 ; 1.0 ; 1.0

○ ○ ● ●

X 182 double 3D linear translation of STL object; $\Delta x, \Delta y, \Delta z$

default: 0.0 ; 0.0 ; 0.0

○ ○ ● ●

X 183 double 3D rotation; $x_{origin}, y_{origin}, z_{origin}, \phi, \theta, \psi$
Positive angles rotate the body clockwise.

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ● ●

X 185 int Type of geometry refinement for triangulated surface mesh

1 based on averaged dx

2 based on global minimum dx

3 based on local minimum dx

default: 1

○ ○ ● ●

X 186 double Geometry refinement factor for triangulated surface mesh

default: 0.7

● ○ ● ○

X 205 int External motion ramp up function type

1 linear

2 sine

default: 1

● ○ ● ○

X 206 double External motion ramp up start time and end time (sec)

default: 0.0 ; 0.0

● ○ ● ○

X 207 double Draft ramp up start time and end time (sec)

default: 0.0 ; 0.0

○ ○ ● ●

X 210 double Prescribing constant linear velocities of the floating body u, v, w

default: 0.0 ; 0.0 ; 0.0

○ ○ ● ●

X 211 double Prescribing constant angular velocities of the floating body p, q, r

default: 0.0 ; 0.0 ; 0.0

● ○ ● ●

X 240 int Read motion file for 6DOF motions. CoG = course over ground / yaw angle in degrees

0 OFF

11 Format: t, x, y, CoG

default: 0

○ ○ ● ●

X 310 int Turn on mooring modelling for floating bodies

0 OFF

1 Static catenary

2 Quasi-Static FEM

3 Dynamic FDM

4 Spring (using X 312)

default: 0

○ ○ ● ●

X 311 * double Specify mooring line configuration $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}, w, \rho_c, EA, d, l, H, P, Q$.

Each mooring line of length $l[m]$ and diameter $d[m]$ has to be specified separately using this option. $\vec{x}_{start}$ is the mounting point at the bottom, $\vec{x}_{end}$ the mounting point at the body. The weight of the line is calculated from the specific weight in air $w[kg/m]$, its density $\rho_c[kg/m^3]$ and its elasticity times area EA. H represents the number of elements for the numerical approaches, P is the polynomial order for the dynamic approach. A relaxation technique can be included for this approach using the time factor Q.

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0 ; 0 ; 0.0

○ ○ ● ●

X 312 * double Specify spring configuration $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}, k, T_0$.

Each spring with stiffness $k[N/m]$ has to be specified separately using this option. $\vec{x}_{start}$ is the mounting point at the bottom, $\vec{x}_{end}$ the mounting point at the body. T_0 is the pre-tension.

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ● ●

X 313 int Initial rotation of mooring end points with the floating body (specified with X 101).

0 off

1 on

default: 0

○ ○ ● ●

X 314 * double Maximum tension force at which each mooring line breaks. If specified, it needs a value for each line. Disable breaking for individiual line by setting the tension force to zero.

default: na

○ ○ ● ●

X 315 * double Maximum time at which each mooring line breaks. If specified, it needs a value for each line. Disable breaking for individiual line by setting the time to zero.

default: na

○ ○ ○ ●

X 320 int Turn on net modelling. Each net has to be specified separately using this option.

0 off

2 Static net cylinder

3 Static net wall

4 Static sheets following floating body

12 Dynamic net cylinder

13 Dynamic net wall

default: 0

○ ○ ○ ●

X 321 * double Specify net material $Sn, d_t, l_t, d_k, \rho, N_d, N_l$.

Each net has to be specified separately using this option. Sn is the solidity, d_t is the diameter of each twine, l_t its length, d_k is the diameter of the knots (0.0 if knotless), ρ is the density of the net material. The net is discretised in N_d meshes along D and N_l meshes along L (to be specified in X 322).

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

X 322 * double Specify net dimensions $D, L, x_0, y_0, z_0, \phi, \theta, \psi$.

Each net has to be specified separately using this option. D is the diameter or width of the net, L is the height. The structure is first rotated around the origin with the given angles around the x-, y- and z-axis. Then, the centre point of the bottom of the structure is moved to the position (x_0, y_0, z_0).

default: 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0 ; 0.0

○ ○ ○ ●

X 323 * double mass in air, double diameter and double length of sinker for dynamic net models.

default: 0.0; 0.0; 0.0

○ ○ ○ ●

X 324 * double x-location, double y-location and double z-location of knot probes.

The dynamic net model searches for the nearest knot to track.

default: na

○ ○ ○ ●

X 325 * double time step, double relaxation in x-direction double relaxation in y-direction double relaxation in z-direction for dynamic net models. Set time step to zero to use the fluid domain time stepping.

default: 0.0; 0.0; 0.0; 0.0

○ ○ ○ ●

X 400 int External moving pressure field for SFLOW (requires X 401)

0 off

2 Box (defined with X 23 and X 110)

10 STL (automatic x_g and y_g)

default: 0

○ ○ ○ ●

X 401 * double p_0, double c_l, double c_b double a Coefficients for the definition of the external moving pressure field.

default: na na na na

3.15 Z :: FSI

Valid for:
S F N C

○ ○ ○ ●

Z 10 int Turn on FSI algorithm for flexible bodies

0 Off

1 Strips

default: 0

○ ○ ○ ●

Z 11 * double x, double y, double z, double L, double W, double T double ρ, double E, double I_x, double I_y, double I_z, double ν, double N

Each object has to be specified separately using this option. (x, y, z) is the initial position of the central bottom point of the strip. L is the length of the object, W its width and T its thickness ($L >> W >> T$ has to be valid). Further, ρ is the density of the material, E is the Young modulus, (I_x, I_y, I_z) are the second moments of area, ν is the Poisson ratio. N defines the number of elements per object.

default: na na na na na na na na na na na na na

○ ○ ○ ●

Z 12 * double $c_{d,X}$, double $c_{d,Y}$, double $c_{d,Z}$, double $c_{k,X}$, double $c_{k,Y}$, double $c_{k,Z}$

Adding velocity-related viscous damping in local X, Y, Z direction of the beam. $\vec{c}_d$ for translational motions, $\vec{c}_k$ for rotational motions.

default: 0.0; 0.0; 0.0; 0.0; 0.0; 0.0

4. Compiling the Code

REEF3D depends on gcc compilers, OpenMPI, HYPRE and Eigen. If the system has included all prerequisites, section 4.1 can be skipped. Users with Windows 10 systems can follow the instructions in section 4.4.
For a usage only compilation, see section 4.2; for developers see section 4.3.

4.1 Install Required Packages

Here, the exemplary installation of REEF3D on macOS is shown. Installations on Linux systems can be done in a similar way.

4.1.1 Installing GCC GNU compiler on macOS

- On macOS, open the terminal and install the command line tools, which include gcc and g++ using:
  xcode-select --install

- IMPORTANT: If any problem in the compilation procedure occurs while the command line tools are already installed, a possible solution appears to be to first remove the command line tools and then reinstall them. The problem appears to be related to a version mismatch between the command line tools and the gfortran installation. The command to remove the command line tools is:
  sudo rm -rf /Library/Developer/CommandLineTools

- Install gfortran for macOS from github:
  https://github.com/fxcoudert/gfortran-for-macOS/releases.

  Select the standalone installer, make sure you select the correct one depending on whether you run Intel-based Macs or are using Apple Sillicon.

- Confirm working C, C++ and fortran compilers (in a new terminal, using the commands individually):
  gcc -v
  g++ -v
  gfortran -v

The output of each of these commands is a path. If you do not get any output or an error message, that particular compiler is not installed or has had problems. This needs to be fixed before you proceed to the next step.

4.1.2 Installing OpenMPI

REEF3D depends on the third-party library MPI for the parallelization of the code. MPI is very portable and the de-facto standard for the parallelization of high performance computations. Several interesting documents can be found on the offical MPI homepage: http://www.mpi-forum.org/. Depending on the operation system, several MPI distributions exist. We recommend OpenMPI for macOS and Linux.

- Download source code - http://www.open-mpi.org/software/ and unzip the tar.gz archive.
- Open 'Terminal' and navigate to the location of the archive and type:
  ./configure --prefix=/usr/local/openmpi
- Next compile:
  make -j n all (with n being the number of available cores on your computer)
- Finally, install by:
  sudo make install
- Add the following openmpi locations to the PATH:
  /usr/local/openmpi/lib
  /usr/local/openmpi/bin

  This can be done for example using the following command:

  Option Linux:
  echo 'export PATH=/usr/local/openmpi/bin/:/usr/local/openmpi/lib/:$PATH' >>~/.bashrc

  Remember that the file .bashrc is a hidden file in your home folder. If you obtain an error message that tells you the file does not exist, you can use the file .bash_profile. One of these two files will be available on your system. You can find this by typing the command:
  ls -a
  when you are in your home folder.

  Option 1 Mac/Linux:
  sudo pico /etc/paths

  Option 2 Mac/Linux:
  sudo vi /etc/paths

  Make sure to save the path file when exiting pico or vi.
- Close the terminal and open a new terminal to check if the addition to the path has been successful using:
  echo $PATH (should return several paths, amongst which the ones you just added are the first to be listed). This should be in order before you proceed to the next step.

4.1.3 Installing hypre

hypre Cen [2015], Falgout et al. [2006] is a library for high-performance linear solvers and precondioners and can be downloaded from hypre's github repository:
https://github.com/hypre-space/hypre/releases

- Download source code and unzip the archive.
- Open 'Terminal' and navigate to the "src" folder inside the HYPRE archive and type:

  ./configure --prefix=/usr/local/hypre F77=/usr/local/bin/gfortran

- Next compile:

  make -j n all (with n at least the number of available cores)
- Finally, install by:

  sudo make install

- Add the following HYPRE locations to the PATH:
  /usr/local/hypre/include
  /usr/local/hypre/lib

  The command is the same as that given above for the MPI installation, but with the paths listed above.

4.2 User Compilation & Installation: Makefile

For users who are interested in running the code only, the easiest way is the compilation of REEF3D and DIVEMesh with the Makefiles given in the code repository. After installing MPI and HYPRE, open the terminal and go to the main directories of the code.

- For DIVEMesh, type:
  `make -j n` (with n at least the number of available cores)
- For REEF3D, type:
  `make -j n` (with n at least the number of available cores)

The code will compile in the /build directory and the executable is placed in the /bin directory. In case of errors during the compilation, check the Makefile for correct paths to the external libraries.

4.3 Developer Compilation & Installation: CodeLite

REEF3D is written in highly modular C++ in order to provide efficient code development and maintenance. REEF3D is designed with cross-platform usage in mind. The source code can be compiled under macOS, Windows and Linux. Since the user experience should be the same on all platforms when developing the code, CodeLite was chosen for the Integrated Development Environment (IDE). The reasons for this are:

- it is free under the GPL license
- it is cross-platform and works under macOS , Windows and Linux
- it works with a variety of compilers
- it works with the MPI library
- support for parallel builds

CodeLite can be downloaded here. When the REEF3D source is downloaded, a CodeLite project file is included, so all header and source files are sorted in a tree structure. This should make the orientation and navigation in the source code easier.

The following path settings are to be set in Codelite to compile REEF3D. First, go to settings > build settings and add a new compiler named mpicxx. Rest of the settings are as shown in the following screenshots: (PS: check the location of your gcc, g++ and mpicxx using 'which gcc', 'which g++' and 'which mpicxx' to enter the appropriate paths for the compiler settings in Codelite)

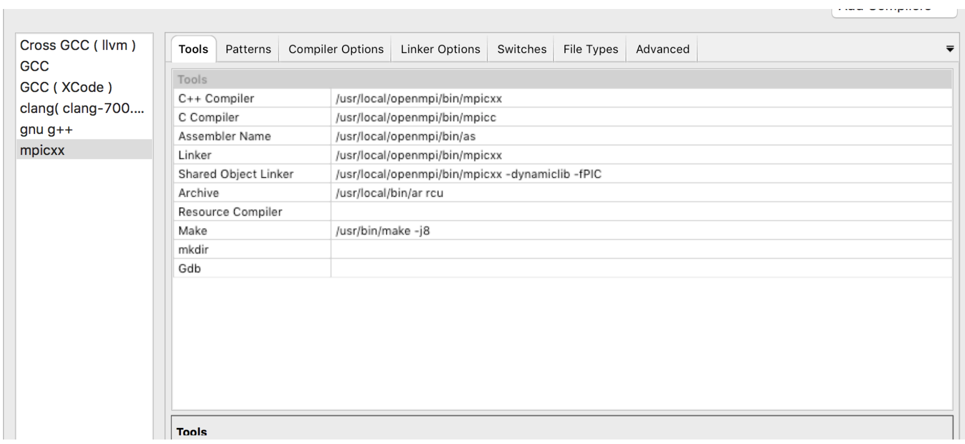


Figure 4.1: Setup of the mpicxx compiler in CodeLite.

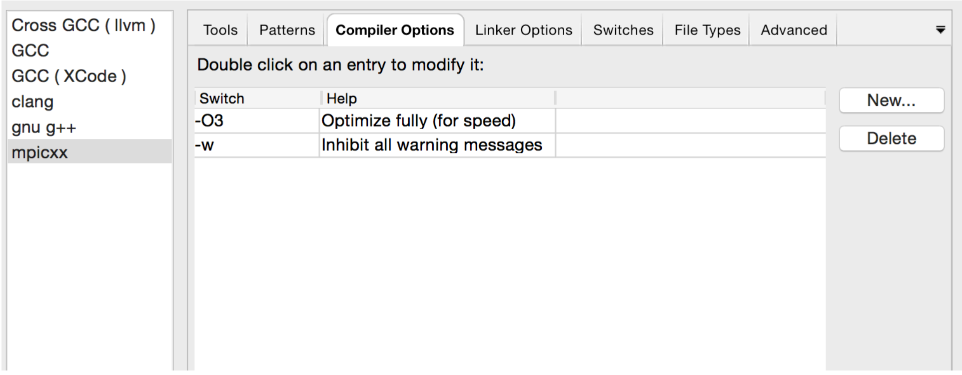


Figure 4.2: Compiler options for mpicxx.

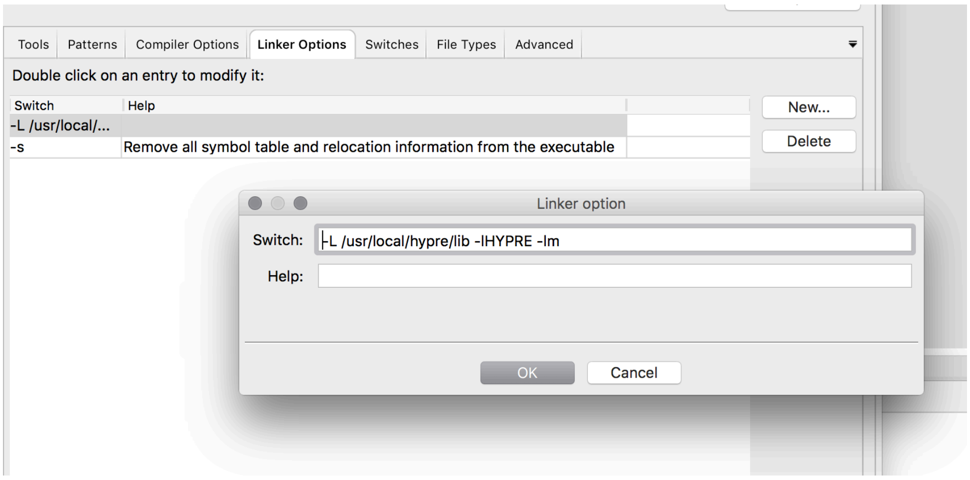


Figure 4.3: Linker options for mpicxx, link to HYPRE.

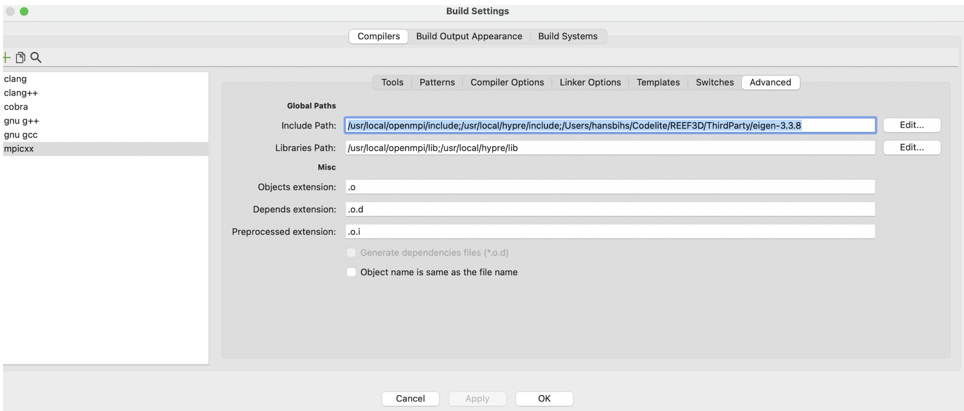


Figure 4.4: Add the Include and Libraries Paths in the advanced tab of the mpicxx compiler. Adapt the path to Eigen according to file structure of the computer.

Afterwards, the code can be compiled by clicking Build > Build Project or pressing F7.

4.4 Installing REEF3D on Windows 10

Windows 10 offers the feature to run a Unix bash terminal, given that you have kept your Windows up-to-date.

- Have administrator privileges on your 64-bit computer
- Check the version of the Windows 10 by navigating to Settings> System> About.
  You need to have a minimum of version 1809. If you have an earlier version, check for updates and install the update giving you version 1809 or above. Restart the computer when prompted. This can take several restarts. No support is provided for Windows update or for versions below 1809.

4.4.1 Getting Unix Terminal on Windows 10

- Look for Ubuntu App in the Microsoft App Store. Ensure that all prerequisites are fulfilled and update the system if necessary.
- Search for 'Ubuntu' in the Start menu and open the bash terminal. You will be prompted to enter a username and password. This can be different from your Windows credentials and has nothing to do with the Windows credentials.
- The Ubuntu terminal starts up at your Ubuntu home folder. To access your Windows C:\Users location, use the command:
  cd /mnt/c/Users/
- Your Unix home directory is unfortunately a hidden folder. You will have to activate the option to View hidden and system files in order to access this using the Windows explorer. To do this:
  - Open your Windows explorer.
  - Click on the View tab at the top of the window
  - Click on the last icon on the toolbar that says Options
  - In the pop-up window, click on the View tab
  - Under the title Hidden Files and Folders, Click on the radio button for "Show hidden files, folders and drives"

- Your Ubuntu home folder can now be accessed at C:\Users\<Windows user name>\AppData\Local \Packages \CanonicalGroupLimitedUbuntuonWindows\LocalState \rootfs \home \<your ubuntu username> .

4.4.2 Installing Compilers

- Open the bash terminal and update the apt package (package manager) using:
  sudo apt update

- To install the necessary GCC and fortran compilers, type:
  sudo apt install gcc g++ gfortran cmake libtool libglu1-mesa

- Use the following commands to find the path to the compilers and note down for later use:
  which gcc
  which g++
  which gfortran

- Get openmpi using:
  sudo apt install libopenmpi-dev

- Continue to install Hypre using the steps listed in Sec 4.1.3.

4.4.3 Additional step for Windows for visualization using Paraview

Ubuntu on Windows does not support graphics and the Windows OS treats the Ubuntu system as hidden directory. In order to overcome this hurdle in Windows and visualise the results using Paraview, the following additional steps are necessary.

- Download the same version of Paraview for both Windows and Linux from paraview.org

- Move the Linux version, the .tar.gz archive, to the Ubuntu home folder by opening an Ubuntu terminal and using:
  cp /mnt/c/Users/<windows username>/Downloads/<ParaviewLinuxArchiveName> ./

- Use the following command to unpack the archive:
  tar -xvf <ParaviewArchiveName>

- Install Paraview on Windows as well, using the installer that you downloaded. Make sure that you are indeed installing the SAME version of Paraview on both Ubuntu and Windows.

4.5 Installing & Compiling REEF3D using Docker

Another alternative route for REEF3D and DIVEMesh compilation and installation irrespective of the underlying OS. In those cases, where the Ubuntu on Windows approach does not work, this may be a viable option. The installation steps are as follows:

- Download + Install Docker Desktop from:
  https://www.docker.com/products/docker-desktop/
  Adjust the number of CPU cores and RAM in the settings (optional), see Fig.

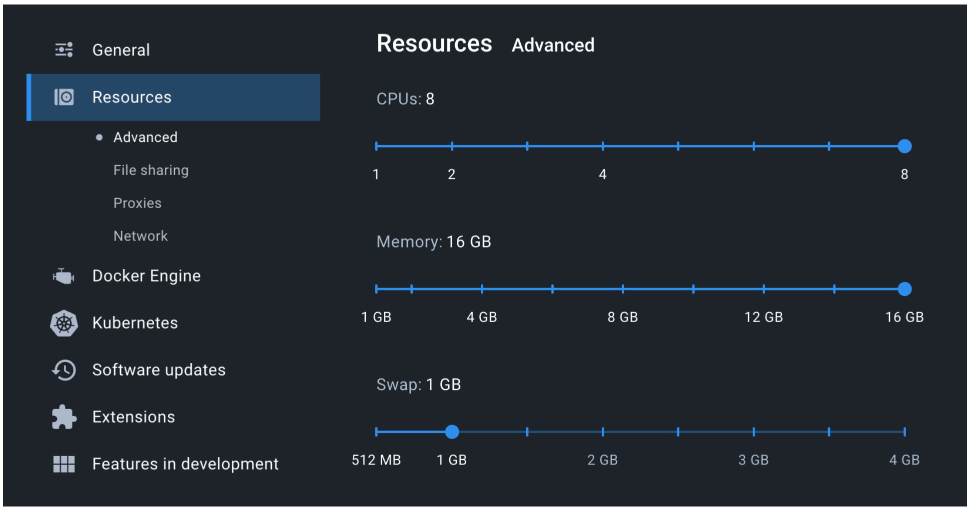


Figure 4.5: Docker resources

- Save the dockerfile (inside the REEF3D folder) in a folder and open the terminal in the folder.
- Type the following command:
  docker build . -t reef3d

  The command creates an image for the container, the image is called reef3d. The process takes a couple of minutes. Afterwards all required dependencies are installed and DIVEMesh and REEF3D are compiled. Then the container can be directly used for running simulations.
- Create a folder with the name '*simulations*' and copy the address.
- Start the docker container in the terminal (detached or attached) and adjust the path for the '*simulations*' folder within <>. For the attached approach, the container does not run in the background whereas for detached the container continues to run in the background.

  detached: **docker run -i –name reef3d -d -t -v </local/path/simulations>:/simulations reef3d**

  attached: **docker run -i –name reef3d -d -t -v /Users/NAME/simulations:/simulations reef3d**
- Connect with the container:

  docker attach reef3d
- Connect with the container:

  docker attach reef3d
- In order to generate a mesh, DIVEMesh needs to be execute in the terminal. Otherwise, the simulations can be started as otherwise described in this User Guide. The container

use the folder '*simulations*' as an interface between the container and the local machine and OS. With the exception of this folder, everything else is stored in the container.

- In the detached mode, the terminal is directly opened for the container.
- other useful commands:

start container:
docker start reef3d

stop container:
docker stop reef3d

restart container:
docker restart reef3d

5. Running the Code

5.1 DIVEMesh

After compilation, copy the DIVEMesh executable into the simulation folder. The simulation folder should be covered by the PATH variable. Make sure that the "control.txt" file is inside the simulation folder, which gives the input for DIVEMesh (see also DIVEMesh User's Guide for more information). Run DIVEMesh by double-clicking the executable or typing in the terminal while being in the path of the folder:

./DiveMESH

This will generate the grid files. If the program does not execute properly, please make sure that all input commands comply with the definitions in the User's Guide.

5.2 REEF3D

After compilation, copy the REEF3D executable into the simulation folder. The simulation folder should be covered by the PATH variable, otherwise the program will not find the MPI libraries. If the program does not execute properly, please make sure that all input commands comply with the definitions in the User's Guide. If a command does not have the required number of input variables, REEF3D will stop at "read ctrl" in the terminal print out.

All REEF3D cases are started with the following terminal command:
mpirun -n *number of processes* **reef3d**

For a computer with four processors, this command then becomes:
mpirun -n 4 reef3d

Alternatively this command can be used:
mpiexec -n 4 reef3d

6. Post-Processing

6.1 Paraview

Paraview is an open-source visualization tool. It is cross-platform and works on macOS, Windows and Linux. Paraview supports parallel output files as well as parallel rendering through distributed computing. Because it is very flexible, efficient and free, it is the preferred program for visualizing REEF3D's computed results.

6.2 Additional step for Windows users to use paraview

In order to access the results files created in the Ubuntu system and open them in the Wondows system, the following additonal steps are essential.

- Open a Ubuntu terminal and navigate to the location of the Paraview installation. If you have followed the instructions exactly, then it is:
  cd <ParaviewFolderName>/bin
- Start a paraview server using:
  ./pvserver
- You will get an output in the terminal that will show you a URL of the form:
  cs://ABCD789: 111111 (This is an example. The actual name will be different, use what you are shown on your screen)
  The first part- ABCD789 is the host name, the second part after the colon (:) is the port number. Make a note of this.
- Open Paraview on your Windows system. Click on the "Connect to server" button (two white towers with a green dot, third icon from the left on the topmost toolbar).
- In the pop-up dialog box, click on Add server. In the dialog box that opens now, enter your host name in the box for Host, and the port number in the box for Port. Use any name you would like in the topmost box for Name.
- Click on configure. And then on save. Click on the saved configuration you are shown on the list and click on Connect.

6.3 Visualising Two-Phase Flow Results in Paraview

REEF3D can write out a range of different result files, depending on the hydrodynamics module used. In this section, the procedure is presented how to visualize the free surface obtained from two-phase flow simulations through the CFD module. For these type of calculations, the level set method is employed which defines the free surface implicitly. The free surface is the zero-contour of the level set function (the variable phi in the paraview dropdown menu). In the following, the procedure is presented on how to extract the zero level set contour from the .vtu files printed out by the CFD module.

Open the .pvtu fileset in paraview through File> open and selecting the .pvtu file in the popup window. Click on outline in the paraview tool bar as shown in Fig.6.1 to obtain the outline of the wave tank. Click on the Slice filter on the toolbar (circled in Fig.6.1) and select "y-normal" in the properties sidebar to generate a 2D slice along the length of the numerical wave tank as shown in Fig.6.2. The slice can be colored according to the various variables available in the dropdown menu. To obtain the free surface, select the slice in the pipeline browser and click on the Contour filter on the tool bar and select Contour by: phi in the properties sidebar with value range 0 as shown in Fig.6.3. The geometry of the structure used in the wave tank is included by adding the .vtp file generated by DiveMesh to obtain the result shown in Fig.6.4.

To obtain a 3D visualization of the free surface, open the .pvtu file in Paraview and directly use the Contour filter on as shown in Fig.10.56.

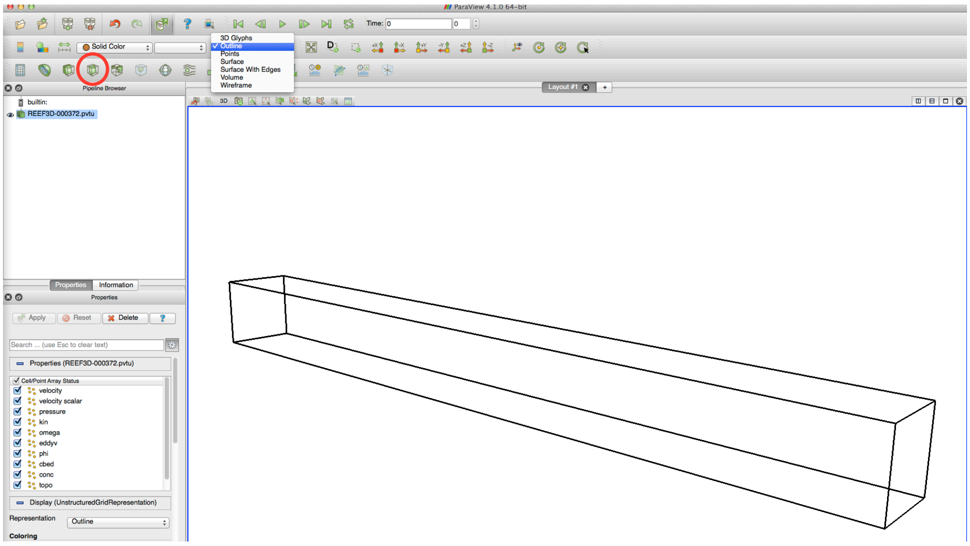


Figure 6.1: Outline of the numerical wave tank

Generating screenshots and animations from Paraview

Use File> Save Screenshot from the drop down menu, input the desired resolution, click OK and select the file type in the next window to save a screenshot from Paraview.

To generate an animation, use File>Save Animation from the dropdown menu. Enter the desired frame rate (recommended value: 1/P30) and the range of time steps to be included in the animation in Frame range. Select the desired file type in the next window.

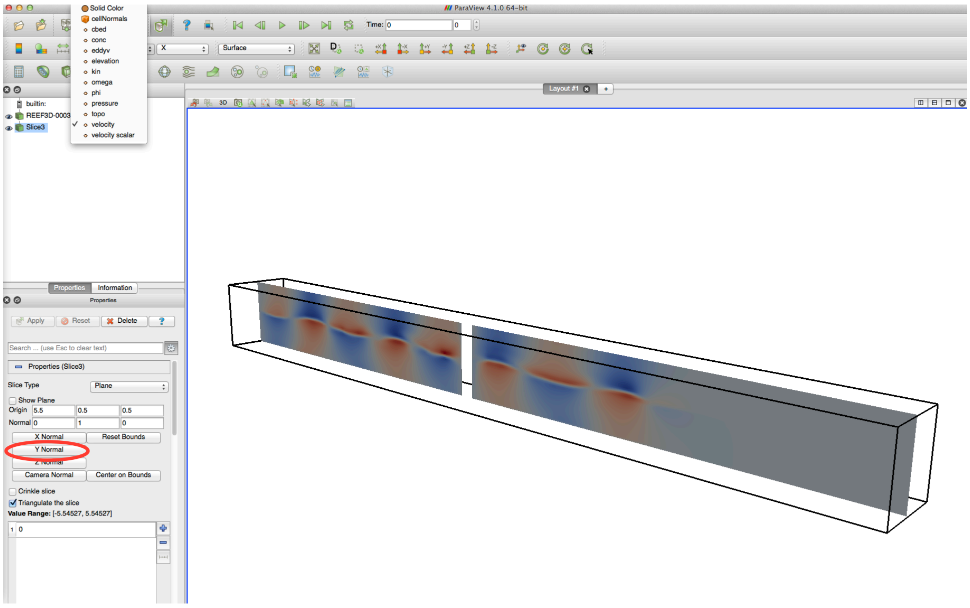


Figure 6.2: Creating a 2D slice

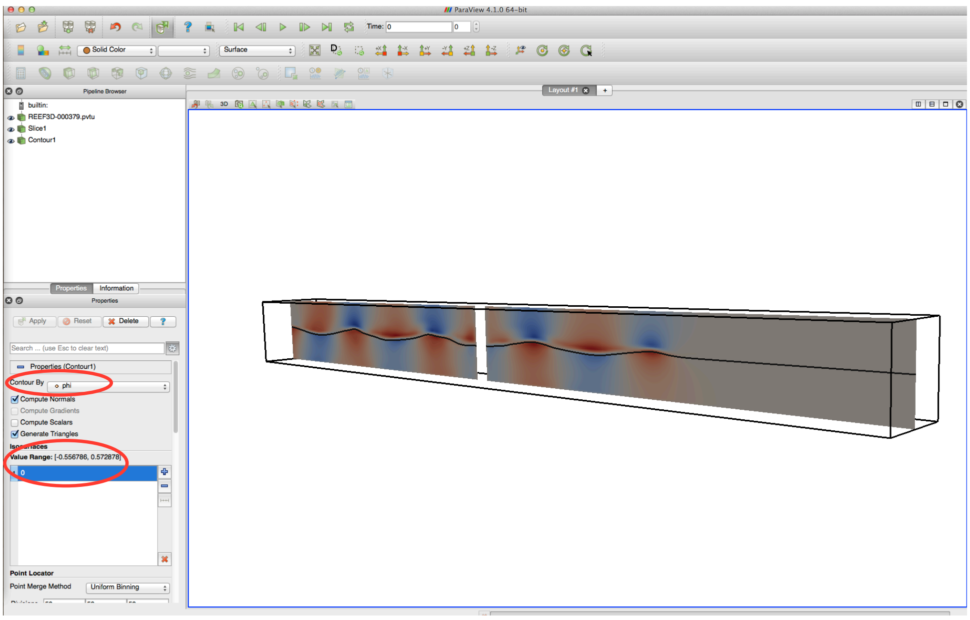


Figure 6.3: Generating the free surface

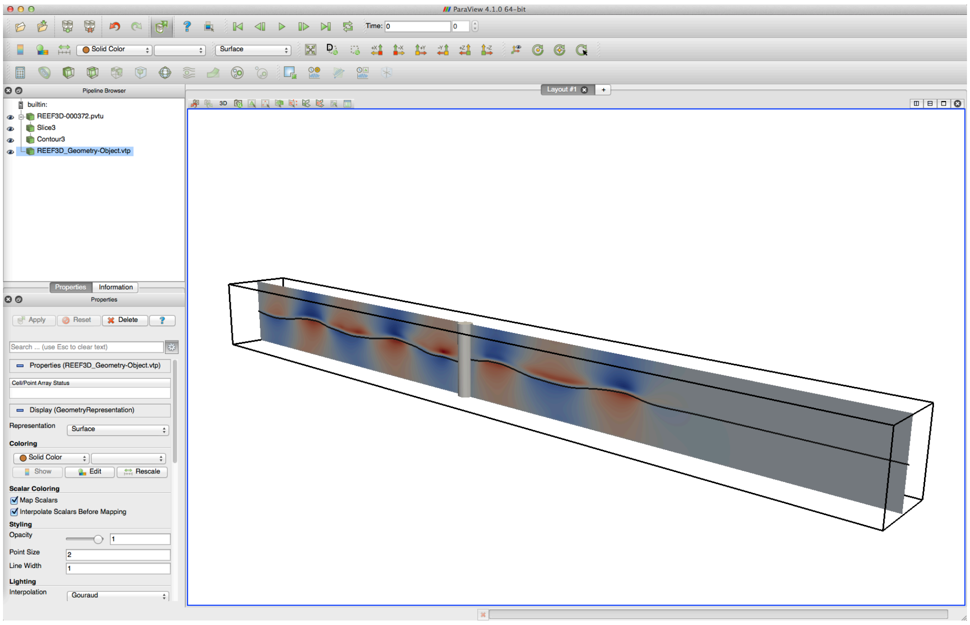


Figure 6.4: Including the geometry of the structure

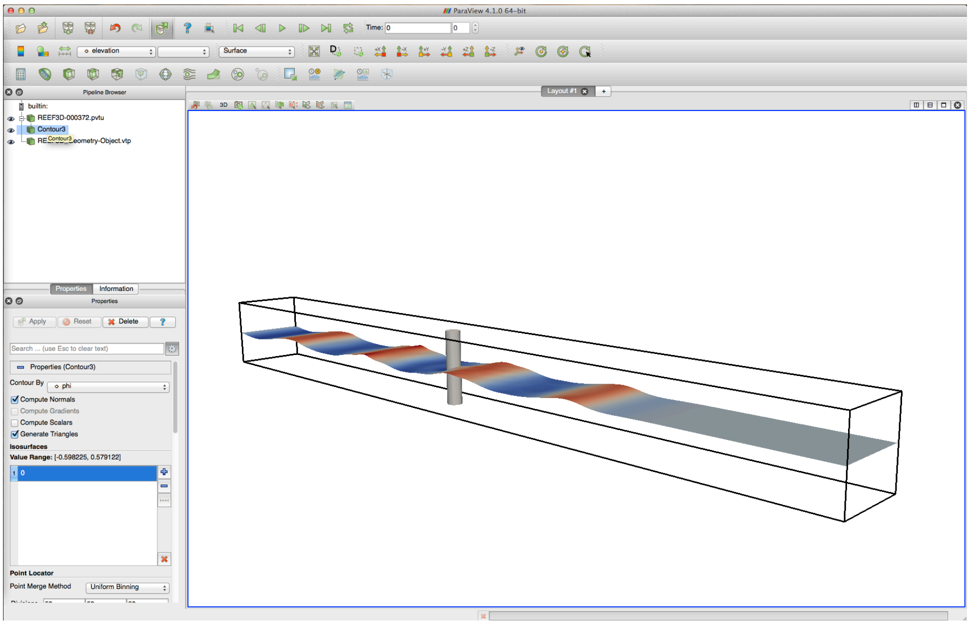


Figure 6.5: Visualizing the free surface in 3D

7. Running the Code

In this chapter some examples are given, the raw input files can be downloaded from the REEF3D website. The focus is on helping the user to understand the workflow of REEF3D. The control files are not necessarily optimized, when it comes to grid convergence and resolution. Rather, the mesh size are kept to reasonable sizes in order to be able to run the cases on laptop or desktop machines.

All REEF3D cases are started with the following terminal command:
mpirun -n *number of processes* reef3d
For a computer with four processors, this command then becomes:
mpirun -n 4 reef3d
Alternatively this command can be used:
mpiexec -n 4 reef3d

The 'ctrl.txt' file contains the input for REEF3D. The idea of the input structure is to use a capital letter for the type of functionality it describes, e.g. 'T' for turbulence. The letter is followed by a number for the individual option. All available options are listed in the previous sections of this document together with the required input, i.e. number of specified values, type of value (int or double) and the default value which will be used by REEF3D in case no input is given by the user.

The 'control.txt' files describe the mesh and are read by DIVEMesh, a separate open-source mesh editor. The structure is the same as in REEF3D and the overview over all available functions is available in the separate DIVEMesh User Guide. Upon executing DIVEMesh, grid files for each of the parallel subprocesses will be generated. The source code for DIVEMesh can also be downloaded from the REEF3D website.

8. Tutorial | REEF3D::SFLOW

For visualization, open the REEF3D_SFLOW_VTP and REEF3D_SFLOW_VTP_BED folders.

8.1 2nd-order Stokes waves

2nd-order Stokes waves over a flat bed.

8.1.1 DIVEMesh: control.txt

C 11 6 // left side: wave generation zone
C 12 3 // side: symmetry plane
C 13 3 // side: symmetry plane
C 14 7 // right side: numerical beach
C 15 21 // bottom: wall boundary
C 16 3 // top: wall boundary

B 1 0.02 // mesh size dx
B 10 0.0 28.0 0.0 0.02 0.0 1.0 // rectangular domain size

M 10 8 // number of processors
M 20 2 // decomposition method

8.1.2 REEF3D: ctrl.txt

A 10 2 // turn on sflow

B 90 1 // turn on iowave
B 92 4 // 2nd-order stokes wave
B 91 0.05 4.0 // wave amplitude and wavelength
B 96 4.0 8.0 // wave generation and absorption relaxation zone lengths
B 98 2 // use relaxation wave generation
B 99 1 // use relaxation wave absorption

F 60 0.5 // water depth

N 41 60.0 // simulation time
N 47 0.2 // adaptive time stepping cfl number

M 10 8 // number of processors

P 10 1 // simulation time print out .vtp files
P 30 0.1 // print out .vtp files interval based on simulation time

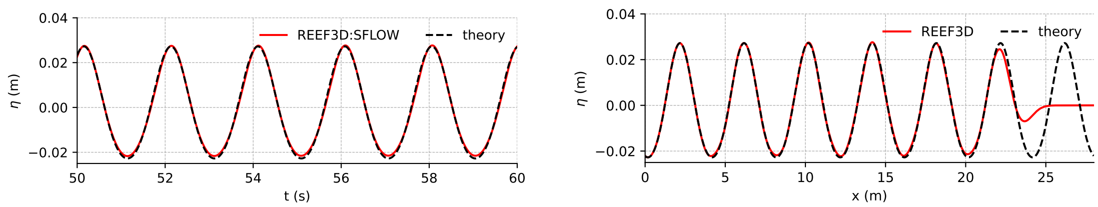
P 50 4.0 0.01 // wave gauge location for theory
P 51 4.0 0.01 // wave gauge location

P 50 14.5 0.01 // wave gauge location for theory
P 51 14.5 0.01 // wave gauge location

P 50 19.0 0.01 // wave gauge location for theory
P 51 19.0 0.01 // wave gauge location P 54 100 // print out wave profile in space every 100 iterations

W 22 -9.81 // gravity

8.1.3 Results



(a) Free surface elevation evolution over time at wave gauge 2 (x =14.5 m)

(b) Free surface elevation in space at t =60 s

Figure 8.1: Free surface elevations in the simulation of 2nd-order Stokes wave in intermediate water depth.

8.2 Cnoidal waves over constant water depth

Cnoidal waves over a flat bed.

8.2.1 DIVEMesh: control.txt

C 11 6 // left side: wave generation zone
C 12 3 // side: symmetry plane
C 13 3 // side: symmetry plane
C 14 7 // right side: numerical beach
C 15 21 // bottom: wall boundary
C 16 3 // top: wall boundary

B 1 0.02 // mesh size dx
B 10 0.0 28.0 0.0 0.02 0.0 1.0 // rectangular domain size

M 10 8 // number of processors
M 20 2 // decomposition method

8.2.2 REEF3D: ctrl.txt

A 10 2 // turn on sflow

B 90 1 // turn on iowave
B 92 8 // cnoidal wave
B 91 0.21 4.0 // wave amplitude and wavelength
B 96 4.0 8.0 // wave generation and absorption relaxation zone lengths
B 98 2 // use relaxation wave generation
B 99 1 // use relaxation wave absorption

F 60 0.5 // water depth

N 41 60.0 // simulation time
N 47 0.2 // adaptive time stepping cfl number

M 10 8 // number of processors

P 10 1 // simulation time print out .vtp files
P 30 0.1 // print out .vtp files interval based on simulation time

P 50 4.0 0.01 // wave gauge location for theory
P 51 4.0 0.01 // wave gauge location

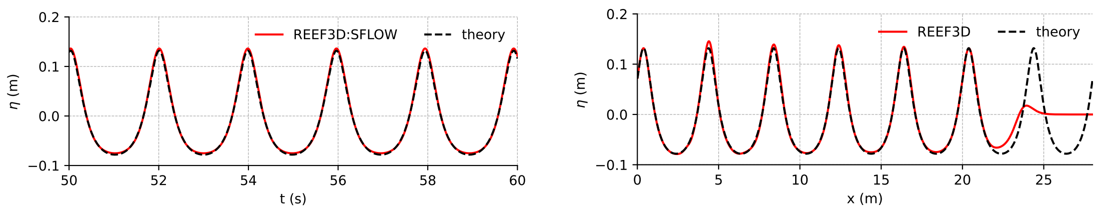
P 50 14.5 0.01 // wave gauge location for theory
P 51 14.5 0.01 // wave gauge location

P 50 19.0 0.01 // wave gauge location for theory
P 51 19.0 0.01 // wave gauge location

P 54 100 // print out wave profile in space every 100 iterations

W 22 -9.81 // gravity

8.2.3 Results



(a) Free surface elevation evolution over time at wave gauge 2 (x =14.5 m)

(b) Free surface elevation in space at t =60 s

Figure 8.2: Free surface elevations in the simulation of cnoidal wave in intermediate water depth.

8.3 Solitary wave over constant water depth

Solitary waves over a flat bed.

8.3.1 DIVEMesh: control.txt

C 11 6 // left side: wave generation zone
C 12 3 // side: symmetry plane
C 13 3 // side: symmetry plane
C 14 7 // right side: numerical beach
C 15 21 // bottom: wall boundary
C 16 3 // top: wall boundary

B 1 0.02 // mesh size dx
B 10 0.0 28.0 0.0 0.02 0.0 1.0 // rectangular domain size

M 10 8 // number of processors
M 20 2 // decomposition method

8.3.2 REEF3D: ctrl.txt

A 10 2 // turn on sflow

B 90 1 // turn on iowave
B 92 9 // solitary wave
B 91 0.05 4.0 // wave amplitude and wavelength
B 96 4.0 0.0 // wave generation and absorption relaxation zone lengths
B 98 2 // use relaxation wave generation
B 99 1 // use relaxation wave absorption

F 60 0.5 // water depth

N 41 20.0 // simulation time
N 47 0.2 // adaptive time stepping cfl number

M 10 8 // number of processors

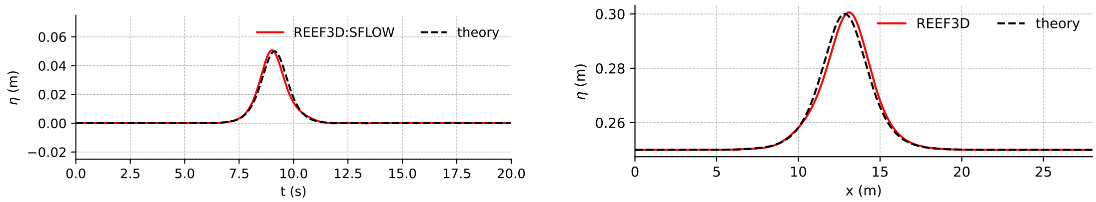
P 10 1 // simulation time print out .vtp files
P 30 0.1 // print out .vtp files interval based on simulation time

P 50 14.5 0.01 // wave gauge location for theory
P 51 14.5 0.01 // wave gauge location

P 54 100 // print out wave profile in space every 100 iterations

W 22 -9.81 // gravity

8.3.3 Results



(a) Free surface elevation evolution over time at wave gauge 2 (x =14.5 m)

(b) Free surface elevation in space at t =8.4155 s

Figure 8.3: Free surface elevations in the simulation of solitary wave in intermediate water depth.

8.4 Wave propagation over a submerged bar

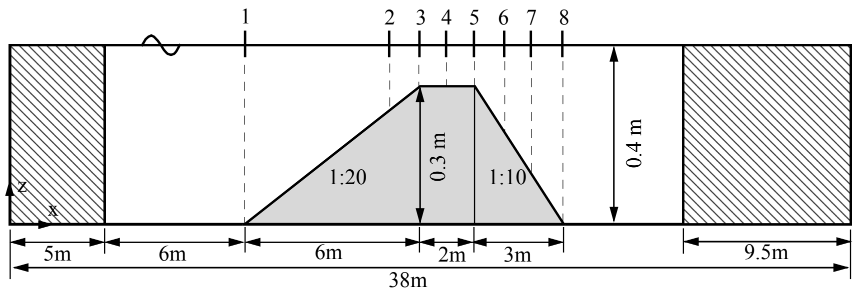


Figure 8.4: Experimtental setup submerged bar

8.4.1 DIVEMesh: control.txt

C 11 6 // left side: wave generation
C 12 3 // side: symmetry plane
C 13 3 // side: symmetry plane
C 14 7 // right side: numerical beach
C 15 21 // bottom: wall boundary
C 16 3 // top: symmetry plane

B 1 0.02 // horizontal mesh size dx
B 10 0.0 38.0 0.0 0.02 0.0 0.8 // rectangular domain size

S 61 11.0 17.0 0.0 0.04 0.0 0.3 // solid wedge object
S 10 17.0 19.0 0.0 0.04 0.0 0.3 // solid rectangular object
S 61 19.0 22.0 0.0 0.04 0.3 0.0 // solid wedge object

M 10 8 // number of processors
M 20 2 // decomposition method 2

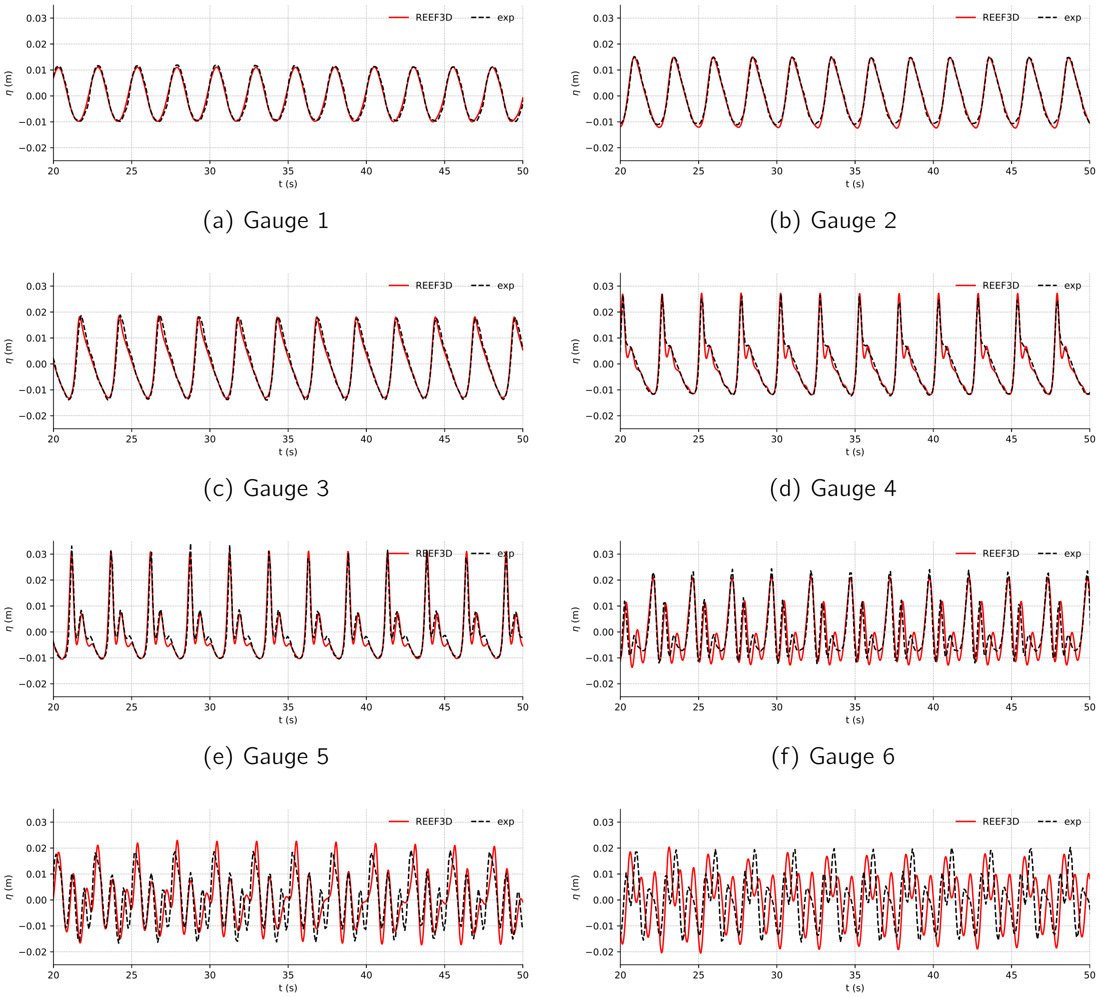
8.4.2 REEF3D: ctrl.txt

A 10 2 // turn on sflow

```
B 90 1 // wave input
B 92 4 // 2nd-order stokes wave
B 93 0.021 2.525 // wave height and wave period
B 96 5.0 10.0 // wave generation zone length and numerical beach length
B 98 2 // relaxation method 2 for wave generation
B 99 1 // relaxation method 1 for numerical beach

F 60 0.4 // still water depth

N 41 60.0 // simulation time
N 47 0.2 // cfl number

M 10 8 // number of processors

P 51 11.0 0.01 // x and y coordinate of wave gauge 1 for simulations
P 51 16.0 0.01 // x and y coordinate of wave gauge 2 for simulations
P 51 17.0 0.01 // x and y coordinate of wave gauge 3 for simulations
P 51 18.0 0.01 // x and y coordinate of wave gauge 4 for simulations
P 51 19.0 0.01 // x and y coordinate of wave gauge 5 for simulations
P 51 20.0 0.01 // x and y coordinate of wave gauge 6 for simulations
P 51 21.0 0.01 // x and y coordinate of wave gauge 7 for simulations
P 51 22.0 0.01 // x and y coordinate of wave gauge 8 for simulations

W 22 -9.81 // gravity
```

8.4.3 Results

Note that the high-frequency bounded waves emerging after the top of submerged bar are in deep water condition, thus the discrepancies with the experiment at wave gauges 6 and onwards. Be ware about the water depth condition when applying the depth-averaged models.



(a) Gauge 1 (b) Gauge 2

(c) Gauge 3 (d) Gauge 4

(e) Gauge 5 (f) Gauge 6

(g) Gauge 7 (h) Gauge 8

Figure 8.5: Free surface elevations in the simulation of wave propagation over a submerged bar.

9. Tutorial | REEF3D::FNPF

9.1 2nd-order Stokes wave in intermediate water depth

9.1.1 DIVEMesh: control.txt

C 11 6 // left side: wave generation
C 12 3 // side: symmetry plane
C 13 3 // side: symmetry plane
C 14 7 // right side: numerical beach
C 15 21 // bottom: wall boundary
C 16 3 // top: symmetry plane

B 1 0.1 // horizontal mesh size dx
B 2 400 1 10 // number of cells in x, y and z directions
B 10 0.0 40.0 0.0 0.1 0.0 2.0 // rectangular domain size

B 103 5 // vertical grid clustering
B 113 2.5 // the stretching factor for the vertical grid clustering
B 116 2.0 // the focal point for the vertical grid clustering, which is water depth here

M 10 8 // number of processors
M 20 2 // decomposition method 2

9.1.2 REEF3D: ctrl.txt

A 10 3 // choose the model reef::fnpf
A 310 3 // 3rd-order runge-kutta for fsfbc time treatment
A 311 4 // 5th-order weno for fsfbc spatial treatment
A 320 1 // 2nd-order laplace
A 343 0 // turn off wetting-drying

B 90 1 // wave input
B 92 4 // 2nd-order stokes wave
B 91 0.04 4.0 // wave height and wave length
B 96 4.0 8.0 // wave generation zone length and numerical beach length
B 98 2 // relaxation method 2 for wave generation
B 99 1 // relaxation method 1 for numerical beach

F 60 2.0 // still water depth

N 41 40.0 // simulation time
N 47 1.0 // cfl number

M 10 8 // number of processors

P 10 1 // turn on .vtu printout
P 30 0.5 // print out .vtu files interval based on simulation time

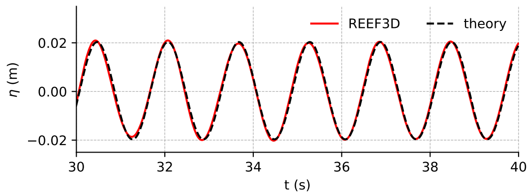
P 50 15.0 0.005 // x and y coordinate of wave gauge 1 for theory
P 50 20.0 0.005 // x and y coordinate of wave gauge 2 for theory
P 50 25.0 0.005 // x and y coordinate of wave gauge 3 for theory

P 51 15.0 0.005 // x and y coordinate of wave gauge 1 for simulations
P 51 20.0 0.005 // x and y coordinate of wave gauge 2 for simulations
P 51 25.0 0.005 // x and y coordinate of wave gauge 3 for simulations

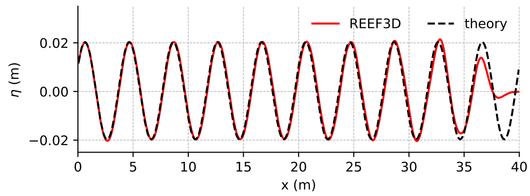
P 54 100 // print out wsfline files interval based on iteration

W 22 -9.81 // gravity

9.1.3 Results



(a) Free surface elevation evolution over time at wave gauge 2 (x =20 m)



(b) Free surface elevation in space at t =40 s

Figure 9.1: Free surface elevations in the simulation of 2nd-order Stokes wave in intermediate water depth.

9.2 Wave propagation over a submerged bar

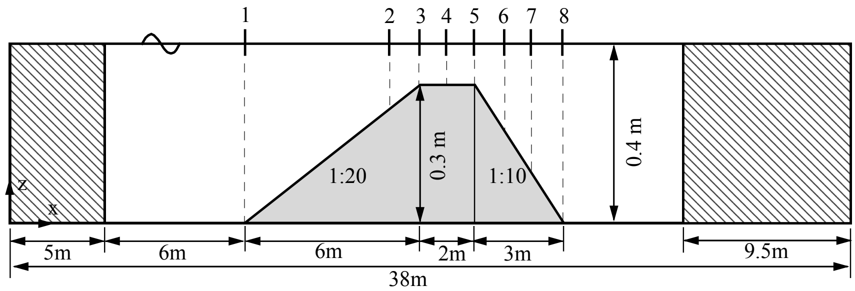


Figure 9.2: Experimtental setup submerged bar

9.2.1 DIVEMesh: control.txt

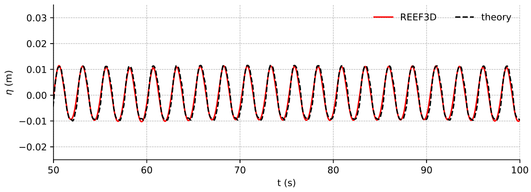
C 11 6 // left side: wave generation
C 12 3 // side: symmetry plane
C 13 3 // side: symmetry plane
C 14 7 // right side: numerical beach

```
C 15 21 // bottom: wall boundary
C 16 3 // top: symmetry plane
B 1 0.04 // horizontal mesh size dx
B 2 1000 1 10 // number of cells in x, y and z directions
B 10 0.0 40.0 0.0 0.04 0.0 1.0 // rectangular domain size
S 61 11.0 17.0 0.0 0.04 0.0 0.3 // solid wedge object
S 10 17.0 19.0 0.0 0.04 0.0 0.3 // solid rectangular object
S 61 19.0 22.0 0.0 0.04 0.3 0.0 // solid wedge object
B 103 5 // vertical grid clustering
B 113 3.0 // the stretching factor for the vertical grid clustering
B 116 1.0 // the focal point for the vertical grid clustering, which is water depth here
M 10 8 // number of processors
M 20 2 // decomposition method 2
```

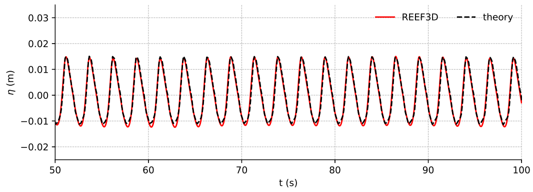
9.2.2 REEF3D: ctrl.txt

```
A 10 3 // choose the model reef::fnpf
A 310 3 // 3rd-order runge-kutta for fsfbc time treatment
A 311 4 // 5th-order weno for fsfbc spatial treatment
A 320 1 // 2nd-order laplace
A 343 0 // turn off wetting-drying
B 90 1 // wave input
B 92 4 // 2nd-order stokes wave
B 93 0.021 2.525 // wave height and wave period
B 96 5.0 10.0 // wave generation zone length and numerical beach length
B 98 2 // relaxation method 2 for wave generation
B 99 1 // relaxation method 1 for numerical beach
F 60 0.4 // still water depth
N 41 100.0 // simulation time
N 47 1.0 // cfl number
M 10 8 // number of processors
P 10 1 // turn on .vtu printout
P 30 0.5 // print out .vtu files interval based on simulation time
P 51 11.0 0.005 // x and y coordinate of wave gauge 1 for simulations
P 51 16.0 0.005 // x and y coordinate of wave gauge 2 for simulations
P 51 17.0 0.005 // x and y coordinate of wave gauge 3 for simulations
P 51 18.0 0.005 // x and y coordinate of wave gauge 4 for simulations
P 51 19.0 0.005 // x and y coordinate of wave gauge 5 for simulations
P 51 20.0 0.005 // x and y coordinate of wave gauge 6 for simulations
P 51 21.0 0.005 // x and y coordinate of wave gauge 7 for simulations
P 51 22.0 0.005 // x and y coordinate of wave gauge 8 for simulations
W 22 -9.81 // gravity
```

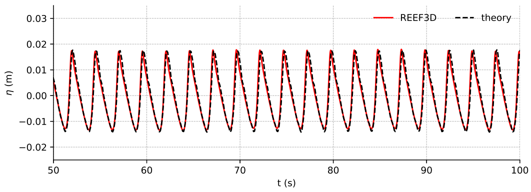
9.2.3 Results



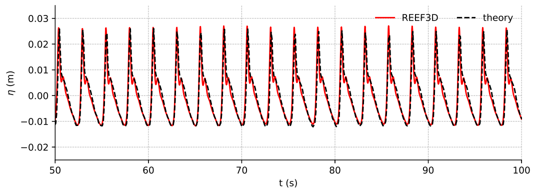
(a) Gauge 1



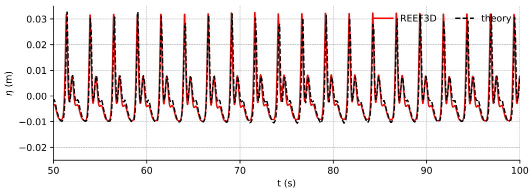
(b) Gauge 2



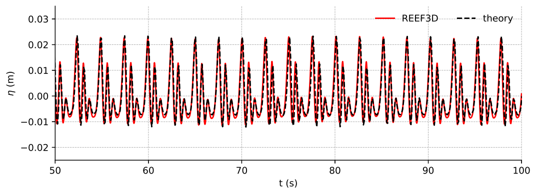
(c) Gauge 3



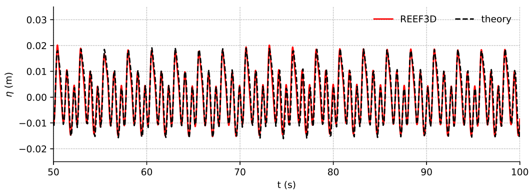
(d) Gauge 4



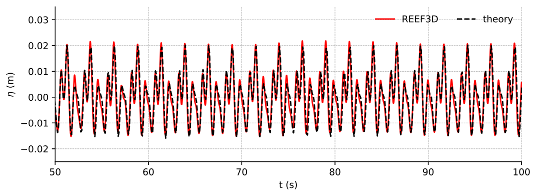
(e) Gauge 5



(f) Gauge 6



(g) Gauge 7



(h) Gauge 8

Figure 9.3: Free surface elevations in the simulation of wave propagation over a submerged bar.

9.3 Wave breaking over a mild slope

9.3.1 DIVEMesh: control.txt

C 11 6 // left side: wave generation
C 12 3 // side: symmetry plane
C 13 3 // side: symmetry plane
C 14 7 // right side: numerical beach
C 15 21 // bottom: wall boundary
C 16 3 // top: symmetry plane

B 1 0.05 // horizontal mesh size dx
B 2 640 1 10 // number of cells in x, y and z directions
B 10 0.0 32.0 0.0 0.05 0.0 0.748 // rectangular domain size

B 103 5 // vertical grid clustering
B 113 1.0 // the stretching factor for the vertical grid clustering
B 116 0.4 // the focal point for the vertical grid clustering, which is water depth here

S 61 5.8 32.0 0.0 0.05 0.0 0.748 // solid wedge object

M 10 8 // number of processors
M 20 2 // decomposition method 2

9.3.2 REEF3D: ctrl.txt

A 10 3 // choose the model reef::fnpf
A 310 3 // 3rd-order runge-kutta for fsfbc time treatment
A 311 5 // 5th-order weno for fsfbc spatial treatment with wetting an drying
A 320 1 // 2nd-order laplace
A 341 2.0 // size of coastal relaxation zone by factor of cell size
A 343 1 // turn on wetting-drying
A 350 1 // viscous damping based breaking energy dissipation algorithm
A 351 1 // shallow water breaking detection
A 352 0 // no additional
A 365 0.0025 // artificial viscosity for the breaking dissipation algorithm

B 90 1 // wave input
B 92 8 // 2nd-order stokes wave
B 93 0.128 5.0 // wave height and wave length
B 96 9.5 9.0 // wave generation zone length and numerical beach length
B 98 3 // neumann boundary for wave generation
B 99 1 // relaxation method 1 for numerical beach

F 60 0.4 // still water depth

N 41 40.0 // simulation time
N 47 1.0 // cfl number

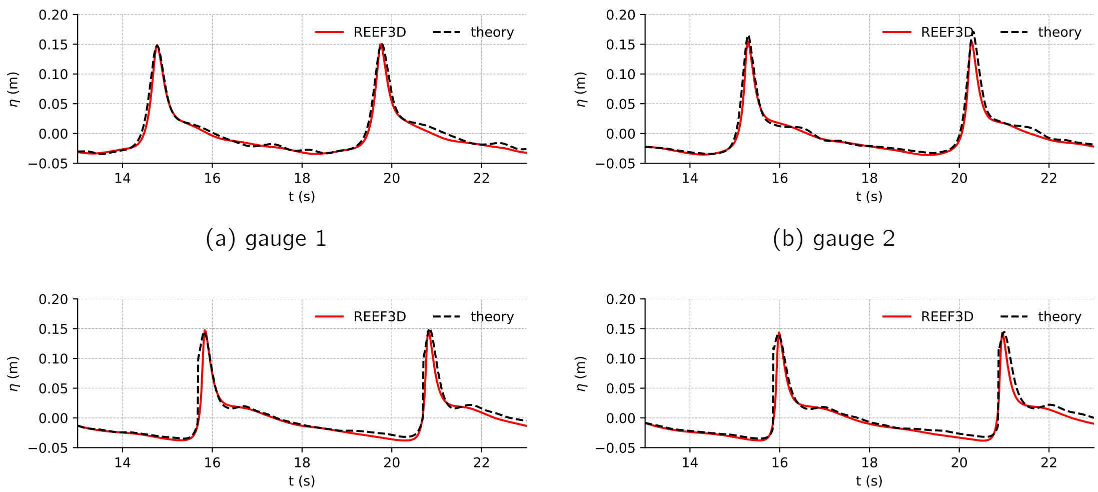
M 10 8 // number of processors

P 10 1 // turn on .vtu printout
P 30 0.05 // print out .vtu files interval based on simulation time

P 51 11.8 0.0025 // x and y coordinate of wave gauge 1 for simulations
P 51 12.8 0.0025 // x and y coordinate of wave gauge 2 for simulations
P 51 13.8 0.0025 // x and y coordinate of wave gauge 3 for simulations
P 51 14.1 0.0025 // x and y coordinate of wave gauge 4 for simulations

W 22 -9.81 // gravity

9.3.3 Results



(a) gauge 1 (b) gauge 2

(c) gauge 3 (d) gauge 4

Figure 9.4: Free surface elevations in the simulation of breaking wave over a mild slope.

9.4 Irregular wave sea state in 2D

9.4.1 DIVEMesh: control.txt

C 11 6 // left side: wave generation
C 12 3 // side: symmetry plane
C 13 3 // side: symmetry plane
C 14 7 // right side: numerical beach
C 15 21 // bottom: wall boundary
C 16 3 // top: symmetry plane

B 1 1.0 // horizontal mesh size dx
B 2 6000 1 20 // number of cells in x, y and z directions
B 10 0.0 6000.0 0.0 1.0 0.0 1.0 // rectangular domain size

B 103 5 // vertical grid clustering
B 113 3.0 // the stretching factor for the vertical grid clustering
B 116 2.0 // the focal point for the vertical grid clustering, which is water depth here

M 10 8 // number of processors
M 20 2 // decomposition method 2

9.4.2 REEF3D: ctrl.txt

A 10 3 // choose the model reef::fnpf
A 310 3 // 3rd-order runge-kutta for fsfbc time treatment
A 311 4 // 5th-order weno for fsfbc spatial treatment
A 320 1 // 2nd-order laplace
A 343 0 // turn off wetting-drying
A 350 1 // viscous damping based breaking energy dissipation algorithm
A 351 2 // deep water wave breaking detection
A 352 0 // no additional filtering for viscous damping breaking model

B 84 1 // peak enhance spectrum discretisation
B 85 2 // jonswap wave spectrum
B 86 2048 // number of wave components
B 87 0.32 1.4 // low and high frequency limits for the frequency band
B 88 3.0 // peak factor of the jonswap spectrum
B 90 1 // wave input
B 92 31 // 1st-order irregular wave
B 93 5.0 12.0 // significant wave height and peak period
B 96 600.0 1200.0 // wave generation zone length and numerical beach length
B 98 2 // relaxation method 2 for wave generation
B 99 1 // relaxation method 1 for numerical beach

F 60 600.0 // still water depth

N 41 12800.0 // simulation time in seconds
N 48 0 // no adaptive time stepping
N 49 0.03 // fixed time step in second

M 10 8 // number of processors

P 10 1 // turn on .vtu printout
P 35 12750.0 12800.0 1.0 // print out .vtu files interval based on a simulation time window

P 51 300.0 0.5 // x and y coordinate of wave gauge 1 for simulations
P 51 1500.0 0.5 // x and y coordinate of wave gauge 2 for simulations
P 51 3500.0 0.5 // x and y coordinate of wave gauge 3 for simulations

W 22 -9.81 // gravity

9.4.3 Results

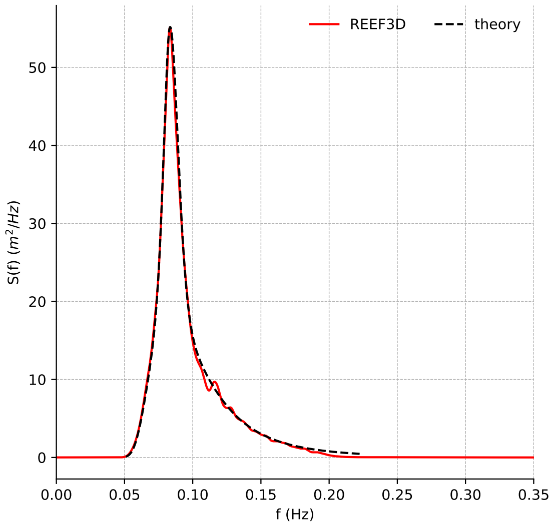


Figure 9.5: wave spectrum in the simulation of a 3-hour irregular sea state at wave gauge 2.

9.5 wave propagation over natural bathymetry

9.5.1 DIVEMesh: control.txt

C 11 6 // left side: wave generation
C 12 3 // side: symmetry plane
C 13 3 // side: symmetry plane
C 14 7 // right side: numerical beach
C 15 21 // bottom: wall boundary
C 16 3 // top: symmetry plane

B 1 5.0 // horizontal mesh size dx
B 2 800 800 10 // number of cells in x, y and z directions
B 10 0.0 4000.0 0.0 4000.0 0.0 1.0 // rectangular domain size

B 103 5 // vertical grid clustering
B 113 2.5 // the stretching factor for the vertical grid clustering
B 116 1.0 // the focal point for the vertical grid clustering, which is water depth here

G 10 1 // turn geodat on/off
G 15 2 // local inverse distance interpolation
G 20 0 // use automatic grid size off
G 31 14 // number of smoothing iterations
G 41 1 // print bottom file from interpolated geo points for the spectrum model swan

M 10 12 // number of processors
M 20 2 // decomposition method 2

9.5.2 REEF3D: ctrl.txt

A 10 3 // choose the model reef::fnpf
A 310 3 // 3rd-order runge-kutta for fsfbc time treatment
A 311 5 // 5th-order weno for fsfbc spatial treatment including wetting-drying
A 320 1 // 2nd-order laplace
A 341 2.0 // size of coastal relaxation zone by a factor of the horizontal cell size
A 343 1 // turn on wetting-drying
A 344 0.1 // wetting-drying water depth threshold
A 346 2.1 // added viscosity within the coastal relaxation zone
A 350 1 // viscosity damping wave breaking algorithm
A 351 3 // breaking wave detection for both deep and shallow water
A 352 3 // additional filtering for viscosity based breaking for both deep and shallow water
A 361 5 // filtering outer iterations
A 362 2 // filtering inner iterations
A 365 1.86 // artificial viscosity for breaking wave energy dissipation

B 90 1 // wave input
B 92 4 // 2nd-order stokes wave
B 93 2.0 10.0 // wave height and wave period
B 96 200.0 400.0 // wave generation zone length and numerical beach length
B 107 0.0 4000.0 0.0 0.0 200.0 // wave generation zone length and numerical beach length
B 107 0.0 4000.0 4000.0 4000.0 200.0 // customised numerical beach at the side walls
B 107 4000.0 4000.0 0.0 4000.0 200.0 // customised numerical beach at the end of the tank
B 107 0.0 0.0 2900.0 3500.0 200.0 // customised numerical beach at the side walls
B 108 0.0 0.0 200.0 2900.0 200.0 // customised wave generation zone
B 98 2 // relaxation method 2 for wave generation
B 99 2 // relaxation method 2 for numerical beach

F 60 300.0 // still water depth

G 50 1 // read in geo bathymetry

I 30 0 // turn off full tank initialisation, one can turn it on for a quick check of the setup

N 41 600.0 // simulation time
N 47 1.0 // cfl number

M 10 12 // number of processors

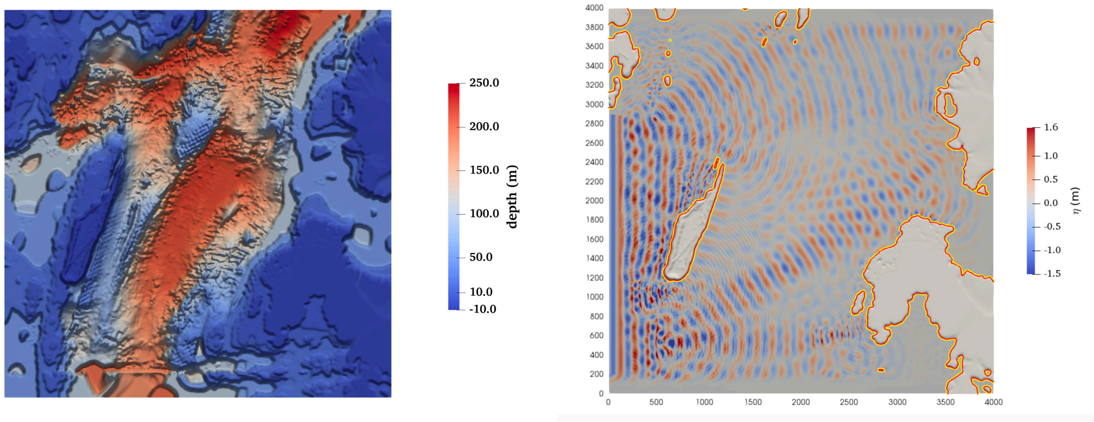
P 10 1 // turn on .vtu printout
P 30 10.0 // print out .vtu files interval based on simulation time

P 51 2000.0 2000.0 // x and y coordinate of wave gauge 1 for simulations

P 180 1 // turn on .vtp free surface printout
P 185 0.0 120.0 0.1 // print out .vtp files interval based on simulation time window
P 185 480.0 600.0 0.1 // print out .vtp files interval based on simulation time window

W 22 -9.81 // gravity

9.5.3 Results



(a) bathymetry (b) Free surface elevation

Figure 9.6: Free surface elevations in the simulation of wave propagation over natural bathymetry

9.6 3D short-crested multi-directional irregular wave

9.6.1 DIVEMesh: control.txt

C 11 6 // left side: wave generation
C 12 3 // side: symmetry plane
C 13 3 // side: symmetry plane
C 14 7 // right side: numerical beach
C 15 21 // bottom: wall boundary
C 16 3 // top: symmetry plane

B 1 2.0 // horizontal mesh size dx
B 2 1000 1000 10 // number of cells in x, y and z directions
B 10 0.0 2000.0 0.0 2000.0 0.0 1.0 // rectangular domain size

B 103 5 // vertical grid clustering
B 113 2.875 // the stretching factor for the vertical grid clustering
B 116 1.0 // the focal point for the vertical grid clustering, which is water depth here

M 10 8 // number of processors
M 20 2 // decomposition method 2

9.6.2 REEF3D: ctrl.txt

A 10 3 // choose the model reef::fnpf
A 310 3 // 3rd-order runge-kutta for fsfbc time treatment
A 311 4 // 5th-order weno for fsfbc spatial treatment
A 320 1 // 2nd-order laplace
A 343 0 // turn off wetting-drying
A 350 1 // viscous damping based breaking energy dissipation algorithm

A 351 2 // deep water wave breaking detection
A 352 0 // no additional filtering for viscous damping breaking model
A 365 1.86 // artificial viscosity for wave breaking algorithm

B 84 2 // equal energy spectrum discretisation
B 85 2 // jonswap wave spectrum
B 86 1024 // number of wave components
B 87 0.39 1.01 // low and high frequency limits for the frequency band
B 88 3.3 // peak factor of the jonswap spectrum
B 90 1 // wave input
B 92 31 // 1st-order irregular wave
B 93 4.5 12.0 // significant wave height and peak period
B 96 400.0 800.0 // wave generation zone length and numerical beach length
B 98 2 // relaxation method 2 for wave generation
B 99 1 // relaxation method 1 for numerical beach
B 130 2 // mitsuyasu directional spreading function
B 131 0.0 // principal direction
B 132 -90.0 90.0 // directional spreading range in degree, measured from the principal direction
B 133 16 // number of directions
B 134 10.0 // spreading parameter for the directional spreading functions
B 136 4 // equal energy method for directional spectrum

F 60 600.0 // still water depth

I 30 1 // full tank initialisation for quick check, turn it off for real simulations

N 41 12800.0 // simulation time in seconds
N 48 0 // no adaptive time stepping
N 49 0.08 // fixed time step in second

M 10 8 // number of processors

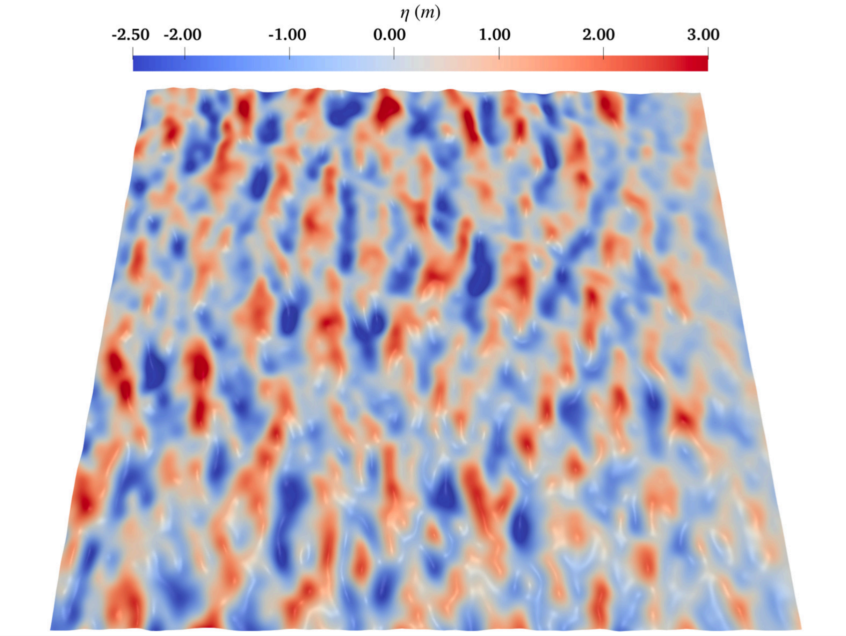
P 10 1 // turn on .vtu printout
P 35 12750.0 12800.0 1.0 // print out .vtu files interval based on a simulation time window

P 51 800.0 1000.0 // x and y coordinate of wave gauge for simulations
P 51 777.2464 983.4685 // x and y coordinate of wave gauge for simulations
P 51 808.6911 973.2515 // x and y coordinate of wave gauge for simulations
P 51 828.1250 1000.0000 // x and y coordinate of wave gauge for simulations
P 51 808.6911 1026.7485 // x and y coordinate of wave gauge for simulations
P 51 777.2464 1016.5315 // x and y coordinate of wave gauge for simulations

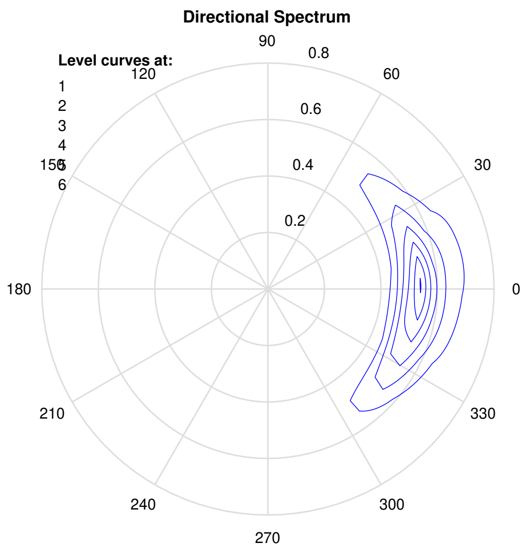
P 180 1 // turn on .vtp free surface printout
P 185 0.0 300.0 0.5 // print out .vtp files interval based on simulation time window
P 185 12500.0 12800.0 0.5 // print out .vtp files interval based on simulation time window

W 22 -9.81 // gravity

9.6.3 Results



(a) Free surface elevation



(b) directional spectrum

Figure 9.7: Free surface elevations and directional spectrum in the simulation of short-crested irregular sea.

10. Tutorial | REEF3D::CFD

10.1 2D Dam Break

This a 2D dam break case. Comments to the control files are marked by blue fonts.

10.1.1 DIVEMesh: control.txt

C 11 21 // left side: wall boundary

C 12 3 // side: symmetry plane

C 13 3 // side: symmetry plane

C 14 21 // right side: wall boundary

C 15 21 // bottom: wall boundary

C 16 21 // top: wall boundary

B 1 0.005 // mesh size dx

B 10 0.0 0.6 0.0 0.005 0.0 0.6 // rectangular domain size

M 10 4 // number of processors

10.1.2 REEF3D: ctrl.txt

D 10 4 // Conservative WENO discretization for velocities

D 20 2 // implicit diffusion treatment

D 30 1 // projection method for the pressure

F 30 3 // 3rd-order Runge-Kutta Scheme for Level Set Time Treatment

F 40 3 // 3rd-order Runge-Kutta Scheme for Reinitialization Time Treatment

F 50 4 // Level set function is not fixed at inlet or outlet

F 54 0.15 // x-coordinate for end fluid phase one

F 56 0.3 // z-coordinate for end fluid phase one

N 40 3 // 3rd-order Runge-Kutta time treatment for velocities

N 45 25000 // maximum number of iterations

N 47 0.1 // factor for CFL criterion

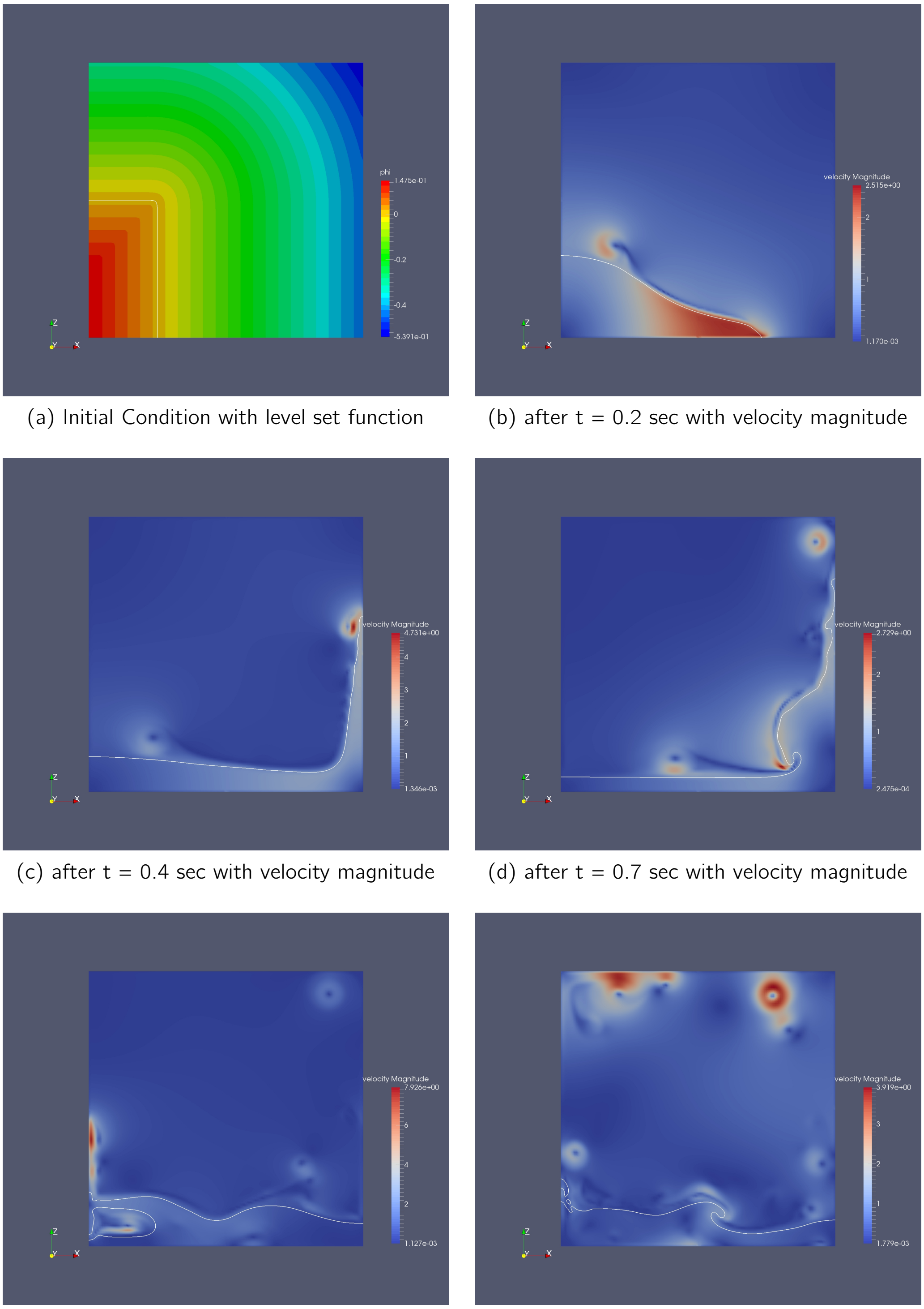
M 10 4 // number of parallel processes

P 10 1 // turn on .vtu print out

P 30 0.01 // print out interval based on simulation time

W 22 -9.81 // gravity

10.1.3 Results



(a) Initial Condition with level set function
(b) after t = 0.2 sec with velocity magnitude
(c) after t = 0.4 sec with velocity magnitude
(d) after t = 0.7 sec with velocity magnitude
(e) after t = 1.0 sec with velocity magnitude
(f) after t = 1.5 sec with velocity magnitude

Figure 10.1: Dam Break

10.2 3D Dam Break with Obstacle

This case is optimized for fast processing. In order to get a more detailed free surface result, just change B 1 to e.g. 0.0125.

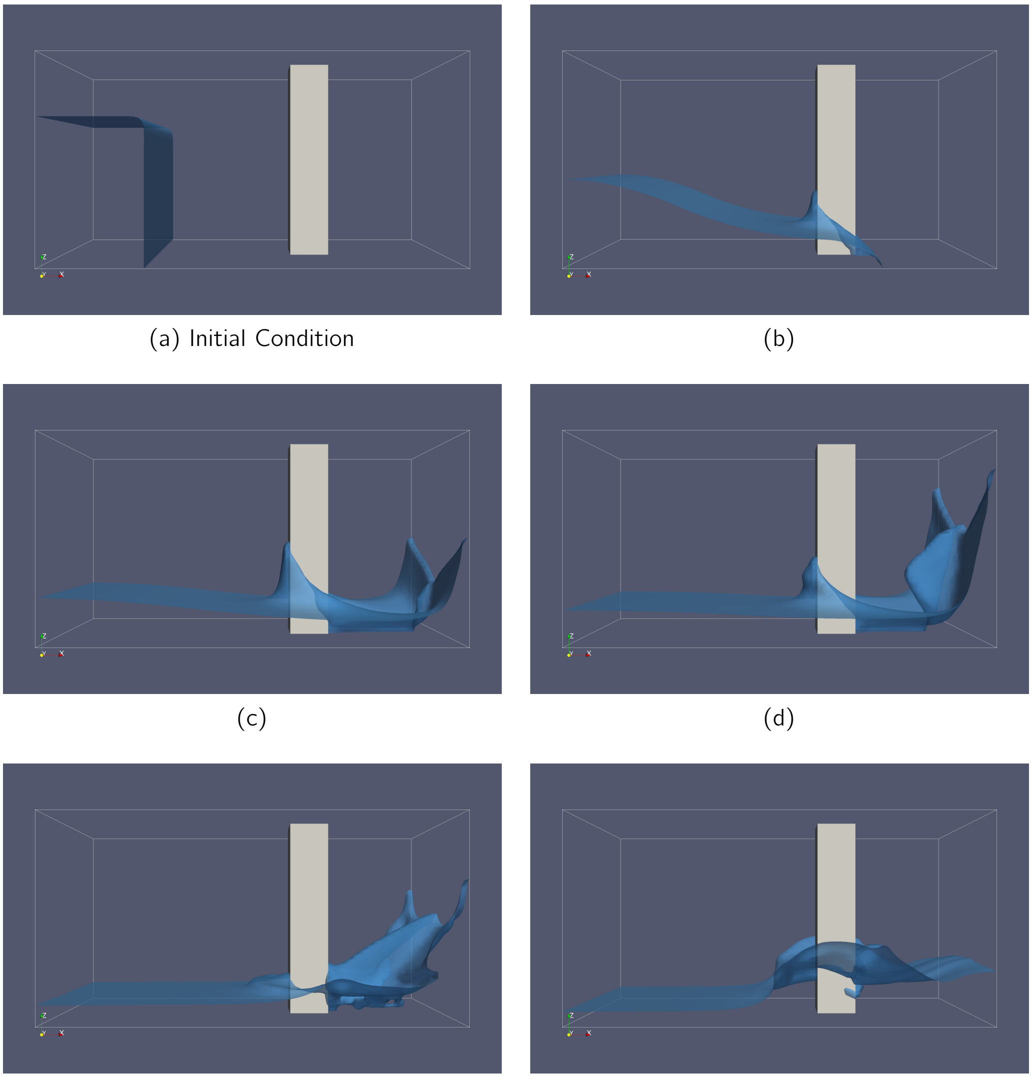
10.2.1 DIVEMesh: control.txt

C 11 21 // left side: wall boundary
C 12 21 // side: wall boundary
C 13 21 // side: wall boundary
C 14 21 // right side: wall boundary
C 15 21 // bottom: wall boundary
C 16 21 // top: wall boundary
B 1 0.025 // mesh size dx
B 10 0.0 2.0 0.0 1.0 0.0 1.0 // rectangular domain size
O 10 1.2 1.4 0.4 0.6 0.0 1.0 // rectangular obstacle size
M 10 4 // number of processors

10.2.2 REEF3D: ctrl.txt

D 10 4 // Conservative WENO discretization for velocities
D 20 2 // implicit diffusion treatment
D 30 1 // projection method for the pressure
F 30 3 // 3rd-order Runge-Kutta Scheme for Level Set Time Treatment
F 40 3 // 3rd-order Runge-Kutta Scheme for Reinitialization Time Treatment
F 50 4 // Level set function is not fixed at inlet or outlet
F 54 0.5 // x-coordinate for end fluid phase one
F 56 0.7 // z-coordinate for end fluid phase one
N 40 3 // 3rd-order Runge Runge time treatment for velocities
N 41 25.0 // maximum simulation time
N 45 25000 // maximum number of iterations
N 47 0.2 // factor for CFL criterion
M 10 4 // number of parallel processes
P 10 1 // turn on .vtu print out
P 30 0.01 // print out interval based on simulation time
T 10 0 // no turbulence model
W 22 -9.81 // gravity

10.2.3 Results



(a) Initial Condition (b)

(c) (d)

(e) (f)

Figure 10.2: 3D Dam Break with Obstacle

10.3 2D Vortex Shedding

10.3.1 DIVEMesh: control.txt

C 11 1 // left side: wall boundary

C 12 3 // side: wall boundary

C 13 3 // side: wall boundary

C 14 2 // right side: wall boundary

C 15 21 // bottom: wall boundary

C 16 21 // top: wall boundary

B 1 0.01 // mesh size dx

B 10 0.0 4.0 0.0 0.01 0.0 1.0 // rectangular domain size

O 32 1.0 0.5 0.1 // cylinder

M 10 4 // number of processors

10.3.2 REEF3D: ctrl.txt

B 10 0 // turn off wall functions for the velocities

B 50 0.0001 // wall roughness k_s

B 60 1 // turn on ioflow

D 10 4 // Conservative WENO discretization for velocities

D 20 1 // explicit diffusion treatment

D 30 1 // projection method for the pressure

F 30 0 // turn off level set transport

F 40 0 // turn off level set reinitialization

I 11 1 // Initialize Velocities with potential flow solver

N 40 3 // 3rd-order Runge Runge time treatment for velocities

N 45 10000 // maximum number of iterations

N 47 0.3 // factor for CFL criterion

M 10 4 // number of parallel processes

P 10 1 // turn on .vtu print out

P 20 10 // print out interval based on iterations

P 62 0.0 0.0 0.005 0.005 0.0 1.0 // line probe

T 10 0 // no turbulence model

W 10 0.000006 // discharge

10.3.3 Results

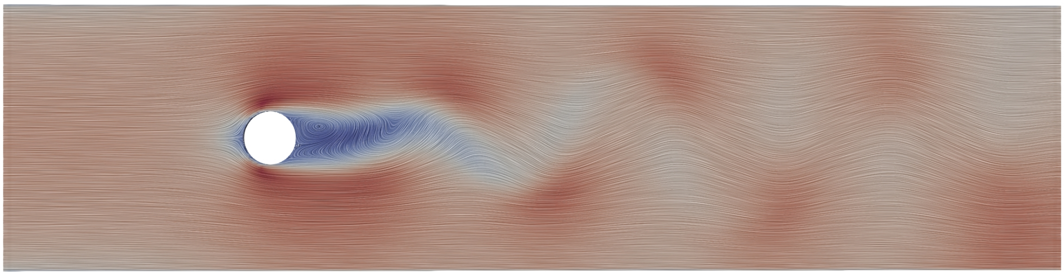


Figure 10.3: 2D vortex shedding with stream lines

10.4 Flow through a Narrow Contraction

For the flow through a narrow contraction the open channel flow capabilities in ioFlow are used. Just for information: the grid is rather coarse for this flow situation. The grid can be easily refined by changing the B1 parameter in the DIVEMesh 'control.txt' file, depending on the performance of your computer.

10.4.1 DIVEMesh: control.txt

C 11 1 // inflow boundary

C 12 21 // wall boundary

C 13 21 // wall boundary

C 14 2 // outflow boundary

C 15 21 // bottom: wall boundary

C 16 3 // top: symmetry plane

B 1 0.025 // mesh size dx

B 10 0.0 2.5 0.0 0.6 0.0 0.5

S 83 0.8 0.0 0.0 1.1 0.0 0.0 1.1 0.2 0.0 0.8 0.0 0.5 1.1 0.0 0.5 1.1 0.2 0.5 S 10 1.1 1.4 0.0 0.2 0.0 0.5

S 83 1.4 0.0 0.0 1.7 0.0 0.0 1.4 0.2 0.0 1.4 0.0 0.5 1.7 0.0 0.5 1.4 0.2 0.5

S 83 0.8 0.6 0.0 1.1 0.6 0.0 1.1 0.4 0.0 0.8 0.6 0.5 1.1 0.6 0.5 1.1 0.4 0.5

S 10 1.1 1.4 0.4 0.6 0.0 0.5

S 83 1.4 0.6 0.0 1.7 0.6 0.0 1.4 0.4 0.0 1.4 0.6 0.5 1.7 0.6 0.5 1.4 0.4 0.5

// Solid: contraction geometry based on wedges (S 83) and boxes (S 10)

M 10 4

10.4.2 REEF3D: ctrl.txt

B 10 1 // use wall functions for the velocities

B 50 0.0001 // wall roughness k_s

B 60 1 // turn on ioFlow

D 10 4 // Conservative WENO for velocities

D 20 2 // implicit diffusion treatment

D 30 1 // projection method for the pressure

F 30 3// 3rd-order Runge-Kutta scheme for the level set method

F 40 3 // 3rd-order Runge-Kutta scheme for reinitialization time treatment

F 42 0.5 // length for level set initialization, instead of maximum length in domain

F 50 2 // keep outflow water level fixed

F 60 0.15 // initial free surface location

I 10 1 // initialize everything, including velocities with potential flow solver

N 40 3 // 3rd-order Runge-Kutta method for the velocities

N 45 12000 // maximum number of iterations

N 47 0.3 // factor for CFL criterion

M 10 4 // number of parallel processes

P 10 1 // turn on .vtu printout

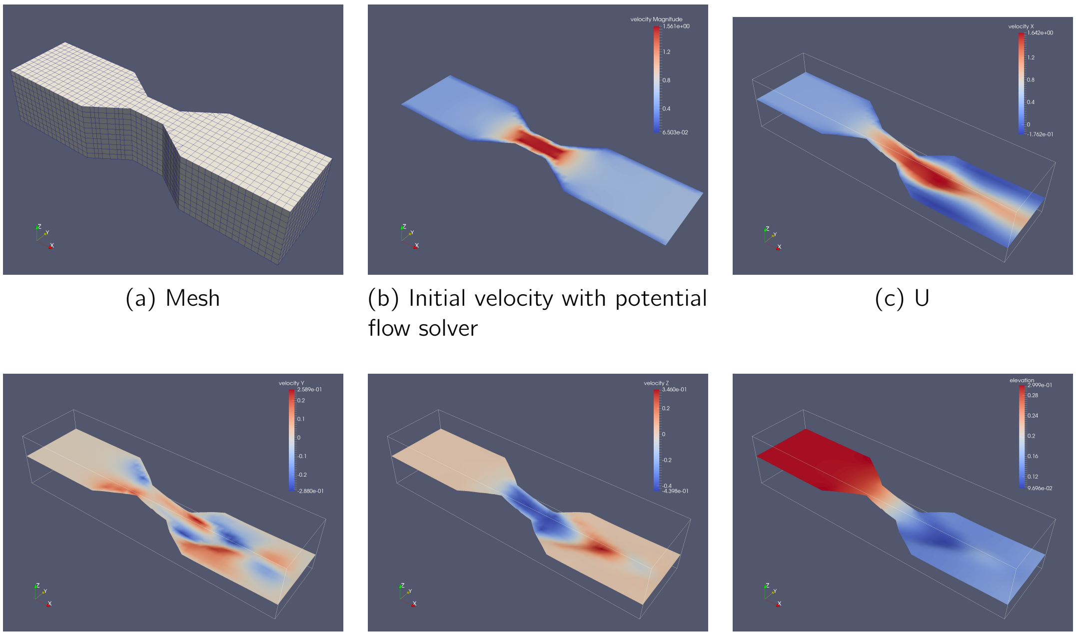
P 20 10 // print out interval based on iterations

T 10 2 // $k-\omega$ turbulence model

W 10 0.05 // inflow discharge

W 22 -9.81 // gravity

10.4.3 Results



(a) Mesh (b) Initial velocity with potential flow solver (c) U

(d) V (e) W (f) Free surface elevation

Figure 10.4: Narrow Contraction

10.5 Flow around a Circular Pier

10.5.1 DIVEMesh: control.txt

```
C 11 1 // inflow boundary

C 12 21 // wall boundary

C 13 21 // wall boundary

C 14 2 // outflow boundary

C 15 21 // bottom: wall boundary

C 16 3 // top: symmetry plane

B 1 0.025 // mesh size dx

B 10 0.0 1.5 0.0 1.0 0.0 0.4 // rectangular domain size

S 33 0.5 0.5 0.075 // vertical circular cylinder

M 10 4 // number of parallel processes
```

10.5.2 REEF3D: ctrl.txt

B 10 1 // use wall functions on the velocities

B 50 0.0001 // wall roughness k_s

B 60 1 // turn on ioFlow

D 10 4 // Conservative WENO discretization for the velocities

D 30 1 // SIMPLE method for the pressure

F 30 3 // second-order implicit time treatment for the level seth method

F 40 3 // 3rd-order Runge-Kutta scheme for reinitialization time treatment

F 50 2 // keep outflow water level fixed

F 60 0.2 // initial free surface location

I 10 1 // initialize everything, including velocities with potential flow solver

N 40 3 // 3rd-order TVD Runge-Kutta method

N 45 100000 // maximum number of iterations

N 47 0.3 // factor for CFL criterion

M 10 4 // number of parallel processes

P 10 1 // turn on .vtu print out

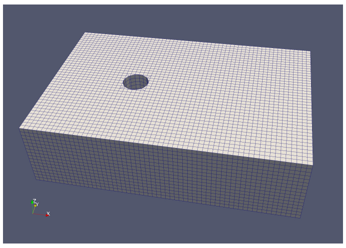
P 20 10 // print out interval based on iterations

T 10 2 // $k - \omega$ turbulence model

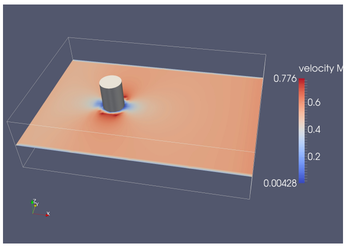
W 10 0.1 // inflow discharge

W 22 -9.81 // gravity

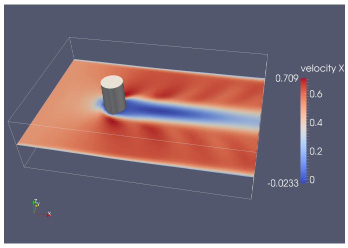
10.5.3 Results



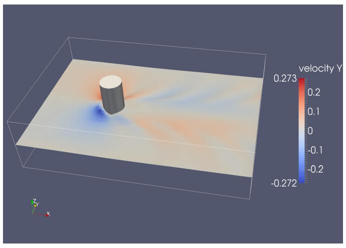
(a) Mesh



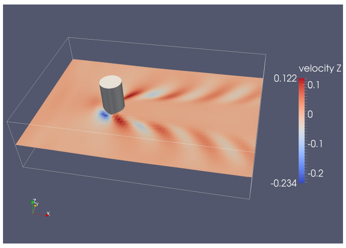
(b) Initial velocity with potential flow solver



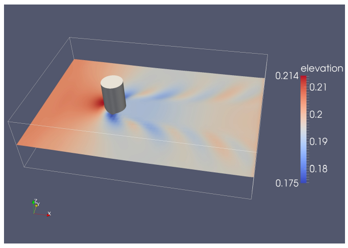
(c) U



(d) V



(e) W



(f) Free surface elevation

Figure 10.5: Flow around Circular Cylinder

10.6 Rectangular Wave Tank

For the case of a simple rectangular wave tank without obstacles, the simulation can be run in 2D. Linear waves are generated, and the propagating waves are compared with the theoretical solution throughout the wave tank. The idea is, that if the wave propagation is represented well by the numerical model, the numerical results will match the theoretical free surface profile in the tank. In this example, the wave has a relatively low steepness, and good convergence can be reached for $dx = 0.05m$ already, which improves further on finer grids. For steeper waves, finer grids may be required.

10.6.1 DIVEMesh: control.txt

C 11 6 // left side: wave relaxation zone

C 12 3 // side: symmetry plane

C 13 3 // side: symmetry plane

C 14 7 // right side: numerical relaxation beach

C 15 21 // bottom: wall boundary

C 16 21 // top: symmetry plane

B 1 0.05 // mesh size dx

B 10 0.0 30.0 0.0 0.05 0.0 1.0 // rectangular domain size

M 10 4 // number of parallel processes

10.6.2 REEF3D: ctrl.txt

B 10 1 // use wall functions for the velocities

B 50 0.0001 // wall roughness k_s

B 90 1 // turn on the numerical wave tank

B 92 2 // use linear waves

B 91 0.02 4.0 // wave height, wave length

B 96 4.0 8.0 // wavegen relaxation length, numerical beach length

B 98 2 // use relaxation method 2 for wave generation

B 99 2 // use relaxation method 2 for numerical beach

D 10 4 // Conservative WENO discretization for velocity convection

D 20 2 // Implicit diffusion for velocities

D 30 1 // Projection Method for the Pressure

F 30 3 // 3rd-order Runge-Kutta scheme for level set time treatment

F 40 3 // 3rd-order Runge-Kutta scheme for reinitialization time treatment

F 42 1.0 // length for level set initialization, instead of maximum length of domain

F 60 0.5 // still water level

I 12 1 // hydrostatic pressure initialization

N 40 3 // 3rd-order Runge-Kutta Scheme for velocity time treatment

N 41 90.0 // Maximum simulation time

N 47 0.25 // factor for CFL criterion

M 10 4 // number of parallel processes

P 10 1 // turn on .vtu printout

P 30 0.05 // print out .vtu files interval based on simulation time

W 22 -9.81 // gravity

P 52 0.025 // y-coordinate and print out water surface line (wfsline) in x-direction

P 53 1 // add theoretical wsfline to file

P 54 500 // print out interval based on iterations for wsflines

10.6.3 Results

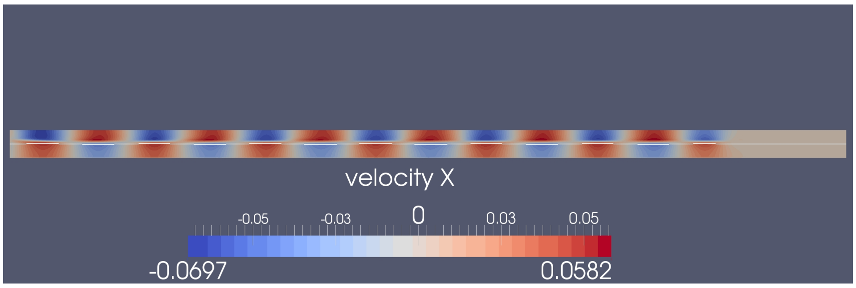


Figure 10.6: Wave tank with free surface and U

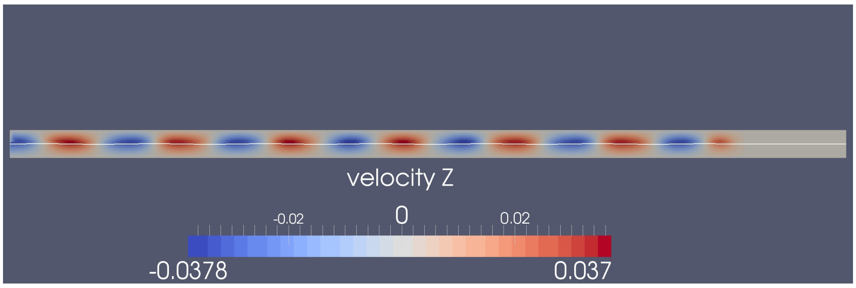


Figure 10.7: Wave tank with free surface and W

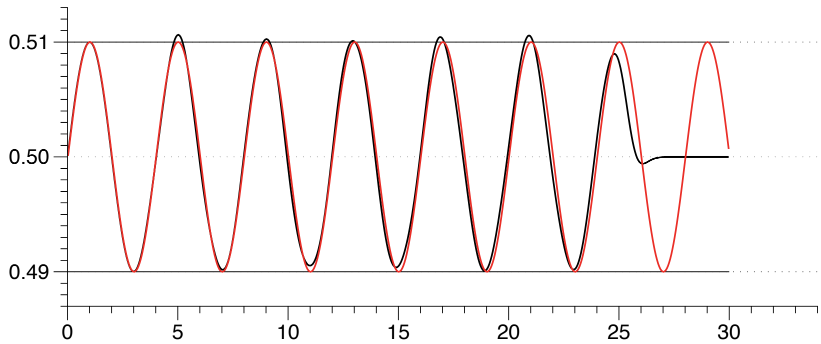


Figure 10.8: Free surface along the tank, numerical result (black line) vs theory (red line)

10.7 Rectangular Wave Tank with DWG and AWA

Similar to the previous chapter, a two-dimensional rectangular wave tank is used. Now, different wave generation and absorption methods are used, namely Dirichlet wave generation (DWG) and active wave absorption (AWA). With DWG, the values for the horizontal and vertical velocities are prescribed at the inlet boundary without further modulation. The level set function is allowed to move freely, but due to the correct flow momentum it will follow the prescribed wave theory. AWA generates a wave opposite to the reflected one at the outlet boundary, thus cancelling out the unwanted reflections. AWA is based on shallow water theory, which also has consequences for type of applications it should be used for, i.e. shallow to intermediate water with preferably longer waves. Overall, the AWA-DWG combination requires smaller tank or basin compared to the relaxation methods (RM). AWA, DWG and RM can be used for generation and absorption in any combination.

10.7.1 DIVEMesh: control.txt

C 11 6 // left side: wave generation

C 12 3 // side: symmetry plane

C 13 3 // side: symmetry plane

C 14 7 // right side: numerical beach AWA

C 15 21 // bottom: wall boundary

C 16 3 // top: symmetry plane

B 1 0.025 // mesh size dx

B 10 0.0 15.0 0.0 0.025 0.0 1.0 // rectangular domain size

M 10 4 // number of parallel processes

M 20 2 // advanced domain decomposition

10.7.2 REEF3D: ctrl.txt

B 10 1 // use wall functions for the velocities

B 50 0.0001 // wall roughness k_s

B 90 1 // turn on the numerical wave tank

B 92 5 // 5th-order stokes waves

B 91 0.1 2.0 // wave height, wave length

B 98 3 // Dirichlet Wave Generation (DWG)

B 99 3 // Active Wave Absorption (AWA)

D 10 4 // Conservative WENO discretization for velocity convection

D 20 2 // Implicit diffusion for velocities

D 30 1 // Projection Method for the Pressure

F 30 3 // 3rd-order Runge-Kutta scheme for level set time treatment

F 40 3 // 3rd-order Runge-Kutta scheme for reinitialization time treatment

F 42 1.0 // length for level set initialization, instead of maximum length of domain

F 60 0.5 // still water level

I 12 1 // hydrostatic pressure initialization

N 40 3 // 3rd-order Runge-Kutta Scheme for velocity time treatment

N 41 60.0 // Maximum simulation time

N 47 0.25 // factor for CFL criterion

M 10 4 // number of parallel processes

P 10 1 // turn on .vtu printout

P 30 0.1 // print out .vtu files interval based on simulation time

P 40 1 // turn state file print out for hot-start

P 42 1.0 // state file print out interval

T 10 0 // no turbulence model

W 22 -9.81 // gravity

P 52 0.0125 // y-coordinate and print out water surface line (wfsline) in x-direction

P 53 1 // add theoretical wsfline to file

P 55 0.50 // print out wsfline files interval based on simulation time

P 51 2.50 0.0125 // wave gage with x- and y-coordinates

P 51 5.00 0.0125 // wave gage with x- and y-coordinates

P 51 15.0 0.0125 // wave gage with x- and y-coordinates

P 51 20.0 0.0125 // wave gage with x- and y-coordinates

10.7.3 Results

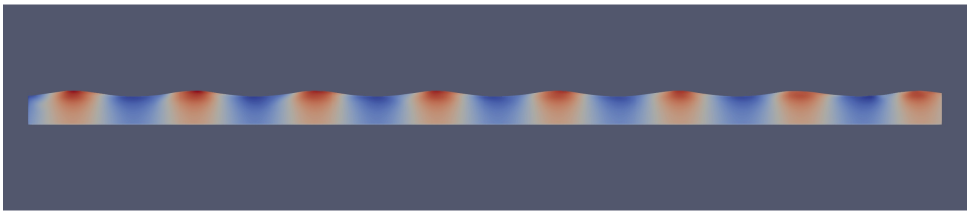


Figure 10.9: DWG-AWA Wave tank with free surface and U

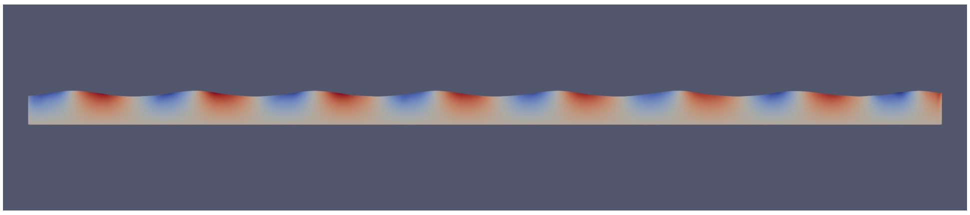


Figure 10.10: DWG-AWA Wave tank with free surface and W

10.8 Rectangular Wave Tank with Wavemaker

Following the NWT examples in the previous examples, now the waves are generated with wavemaker kinematics input, see e.g. Aggarwal et al. [2018a] for more information.

10.8.1 DIVEMesh: control.txt

C 11 6 // left side: wave generation

C 12 3 // side: symmetry plane

C 13 3 // side: symmetry plane

C 14 8 // right side: numerical beach AWA

C 15 21 // bottom: wall boundary

C 16 21 // top: symmetry plane

B 1 0.025 // mesh size dx

B 10 0.0 15.0 0.0 0.025 0.0 1.5 // rectangular domain size

M 10 4 // number of parallel processes

M 20 2 // advanced domain decomposition

10.8.2 REEF3D: ctrl.txt

B 10 1 // use wall functions for the velocities

B 50 0.0001 // wall roughness k_s

B 90 1 // turn on the numerical wave tank

B 92 22 // flap wavemaker

B 98 3 // Dirichlet Wave Generation (DWG)

B 99 3 // Active Wave Absorption (AWA)

B 111 -0.25 1.46 // z_{start} and z_{end} for flap wavemaker

B 116 2 // flap wavemaker input as angle

B 117 4.0 // time shift for wavemaker input

D 10 4 // Conservative WENO discretization for velocity convection

D 20 2 // Implicit diffusion for velocities

D 30 1 // Projection Method for the Pressure

F 30 3 // 3rd-order Runge-Kutta scheme for level set time treatment

F 40 3 // 3rd-order Runge-Kutta scheme for reinitialization time treatment

F 42 1.5 // length for level set initialization, instead of maximum length of domain

F 60 0.75 // still water level

I 12 1 // hydrostatic pressure initialization

N 40 3 // 3rd-order Runge-Kutta Scheme for velocity time treatment

N 41 60.0 // Maximum simulation time

N 47 0.25 // factor for CFL criterion

M 10 4 // number of parallel processes

P 10 1 // turn on .vtu printout

P 30 0.1 // print out .vtu files interval based on simulation time

P 40 1 // turn state file print out for hot-start

P 42 1.0 // state file print out interval

T 10 0 // no turbulence model

W 22 -9.81 // gravity

P 52 0.0125 // y-coordinate and print out water surface line (wfsline) in x-direction

P 55 0.50 // print out wsfline files interval based on simulation time

P 51 5.00 0.0125 // wave gage with x- and y-coordinates

P 51 10.0 0.0125 // wave gage with x- and y-coordinates

10.8.3 REEF3D: wavemaker.dat

The 'wavemaker.dat' file contains the time dependent kinematics of the wavemaker.

10.8.4 Results

Figure 10.11: Wavemaker generated waves with free surface and U

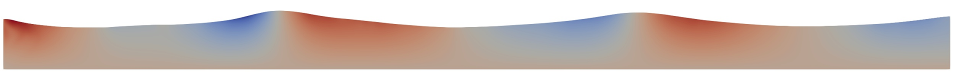


Figure 10.12: Wavemaker generated waves with free surface and W

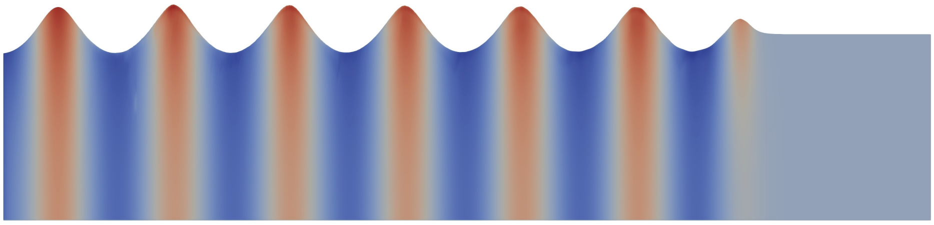


Figure 10.13: 5th-order Stokes Waves setup

10.9 5th-order Stokes Waves

This verification case shows the propagation of 5th-order Stokes waves in a 200 m long tank and highlights the use of velocity probes both from the model and the wave theory.

10.9.1 DIVEMesh: control.txt

C 11 6 // left side: wave relaxation zone

C 12 3 // side: symmetry plane

C 13 3 // side: symmetry plane

C 14 7 // numerical relaxation beach

C 15 21 // bottom: wall boundary

C 16 3 // top symmetry plane

B 1 0.025 // mesh size dx

B 10 0.0 200.0 0.0 0.025 0.0 8.0 // rectangular domain size

M 10 12 // number of parallel processes

10.9.2 REEF3D: ctrl.txt

B 10 1 // use wall functions for the velocities

B 90 1 // turn on the numerical wave tank

B 92 5 // 5th-order Stokes waves

B 91 1.0 4.5 // wave height, wave length

B 96 25.0 50.0 // wavegen relaxation length, numerical beach length

B 98 2 // use relaxation method 2 for wave generation

B 99 2 // use relaxation method 2 for numerical beach

D 10 4 // Conservative WENO for velocity convection

```
D 20 2 // Implicit diffusion for velocities
D 30 2 // Projection Correction Method for the pressure
F 30 3 // 3rd-order Runge-Kutta scheme for Level Set time treatment
F 40 3 // 3rd-order Runge-Kutta scheme for Reinitialization time treatment
F 42 0.8 // length for level set initialization, instead of maximum length of domain
F 60 4.01 // still water level
I 12 1 // hydrostatic pressure initialization
N 40 3 // 3rd-order Runge-Kutta scheme for velocity time treatment
N 41 120.0 // Maximum simulation time
N 47 0.3 // factor for CFD criterion
M 10 12 // number of parallel processes
P 10 1 // turn on .vtu printout
P 30 0.25 // print out interval for .vtu files based on simulation time
W 22 -9.81 // gravity
P 52 0.005 // y-coordinate and print out water surface line wsfline in x-direction
P 53 1 // add theoretical wsfline to file
P 54 10 // print out interval for wsflines based on iterations
P 50 15.0 0.005 // Wave gage 1 for wave theory
P 50 50.0 0.005 // Wave gage 2 for wave theory
P 50 75.0 0.005 // Wave gage 3 for wave theory
P 50 100.0 0.005 // Wave gage 4 for wave theory
P 50 125.0 0.005 // Wave gage 5 for wave theory
P 51 15.0 0.005 // Wave gage 1
P 51 50.0 0.005 // Wave gage 2
P 51 75.0 0.005 // Wave gage 3
P 51 100.0 0.005 // Wave gage 4
P 51 125.0 0.005 // Wave gage 5
P 65 25.0 0.025 3.0 // Velocity probe 1
P 65 75.0 0.025 3.0 // Velocity probe 2
P 65 125.0 0.025 3.0 // Velocity probe 3
P 66 125.0 0.025 3.0 // Velocity probe 1 for wave theory
P 66 25.0 0.025 3.0 // Velocity probe 2 for wave theory
P 66 75.0 0.025 3.0 // Velocity probe 3 for wave theory
```

10.9.3 Free Surface Results

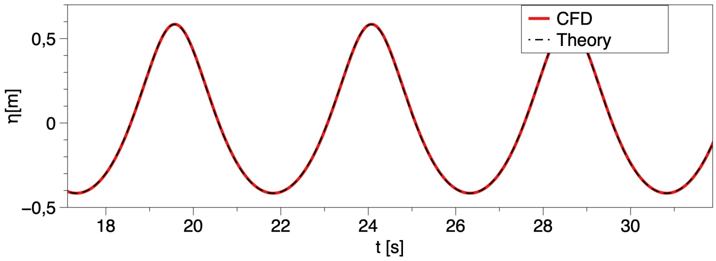


Figure 10.14: Wave gage 1, x = 25m

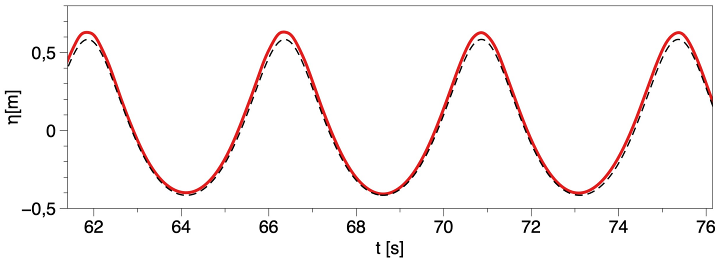


Figure 10.15: Wave gage 2, x = 50m

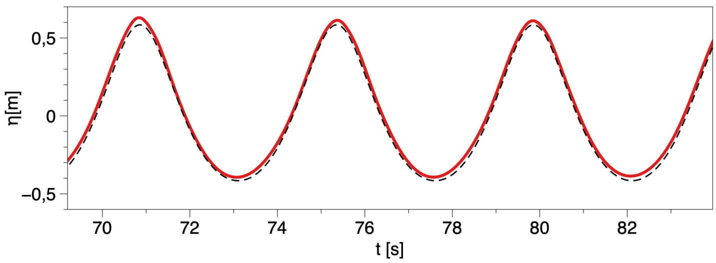


Figure 10.16: Wave gage 3, x = 75m

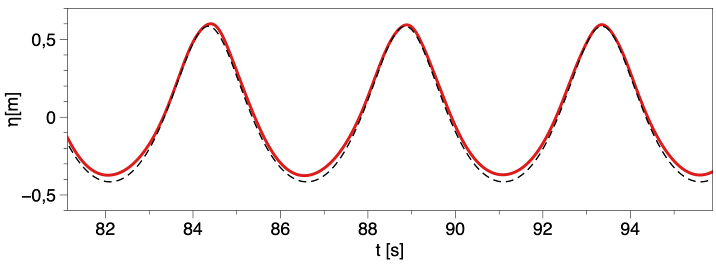


Figure 10.17: Wave gage 4, x = 100m

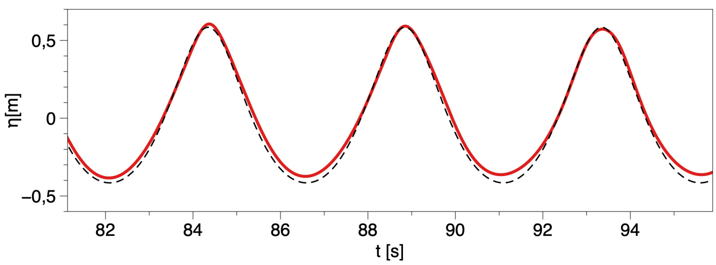


Figure 10.18: Wave gage 5, x = 125m

10.9.4 Velocity Results

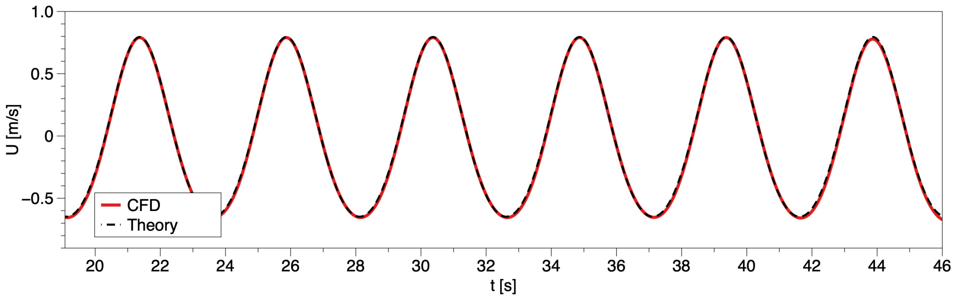


Figure 10.19: Horizontal Velocity, x = 25m, z=3m

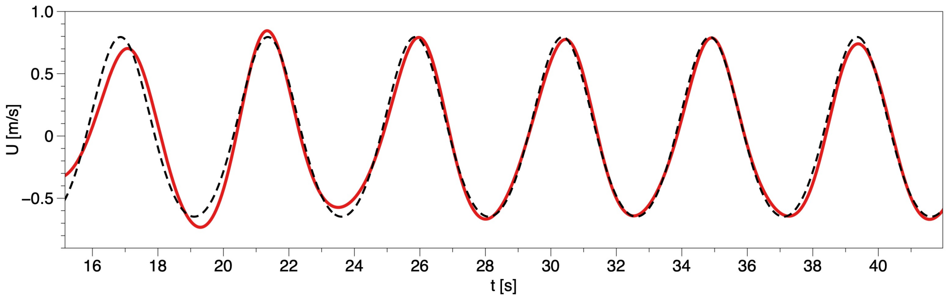


Figure 10.20: Horizontal Velocity, x = 75m, z=3m

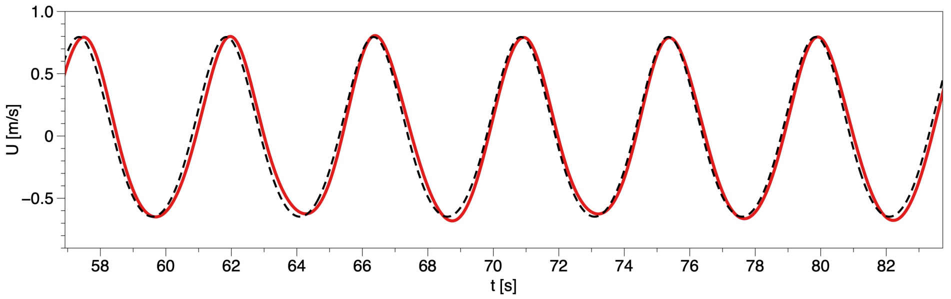


Figure 10.21: Horizontal Velocity, x = 125m, z=3m

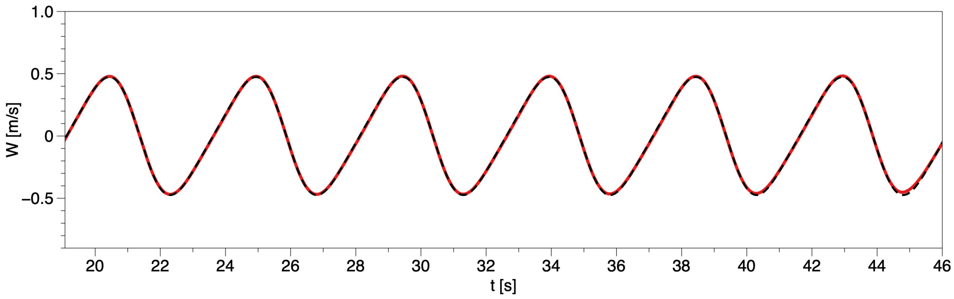


Figure 10.22: Vertical Velocity, x = 25m, z=3m

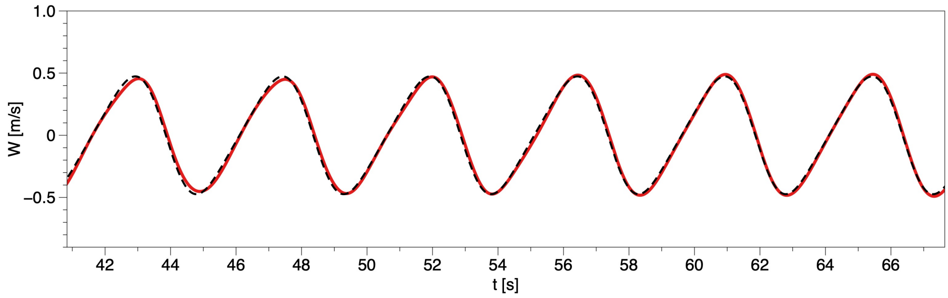


Figure 10.23: Vertical Velocity, x = 75m, z=3m

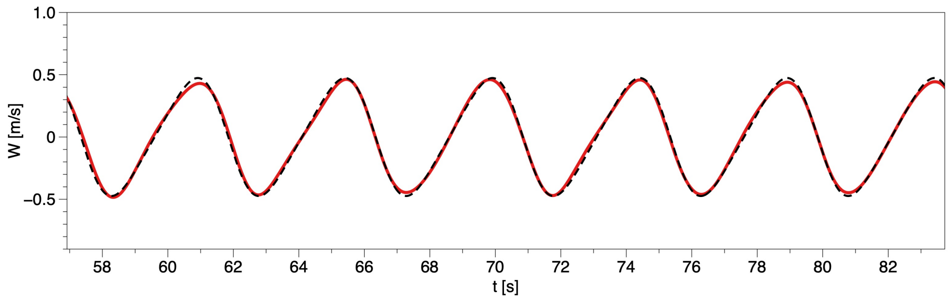


Figure 10.24: Vertical Velocity, x = 125m, z=3m

10.10 Wave Propagation over a Submerged Bar

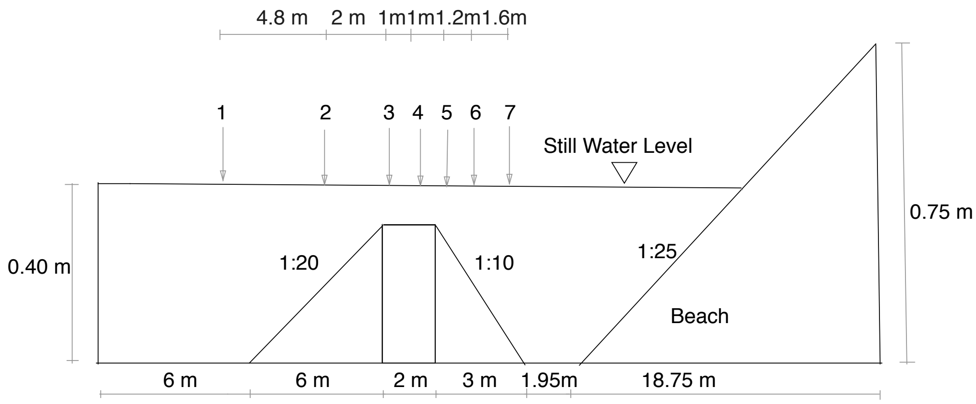


Figure 10.25: Experimtental setup submerged bar

The well-know benchmark case for wave propagation over a submerged bar is presented here. The description of the experimental setup can be found in the original paper by Beji and Battjes [1993]. The simulations are run in 2D with wave gages at several locations along the wave tank.

10.10.1 DIVEMesh: control.txt

C 11 6 // left side: wave relaxation zone

C 12 3 // side: symmetry plane

C 13 3 // side: symmetry plane

C 14 7 // numerical relaxation beach

C 15 21 // bottom: wall boundary

C 16 3 // top symmetry plane

B 1 0.01 // mesh size dx

B 10 0.0 24.0 0.0 0.01 0.0 0.8 // rectangular domain size

S 61 6.0 12.0 0.0 0.01 0.0 0.3 // front wedge of the bar

S 10 12.0 14.0 0.0 0.01 0.0 0.3 // middle section of the bar

S 61 14.0 17.0 0.0 0.01 0.3 0.0 // back wedge of the bar

M 10 4 // number of parallel processes

10.10.2 REEF3D: ctrl.txt

B 10 1 // use wall functions for the velocities

B 50 0.0001 // wall roughness k_s

B 90 1 // turn on the numerical wave tank

B 92 2 // use linear waves

B 91 0.02 3.73 // wave height, wave length

B 96 3.73 3.73 // wavegen relaxation length, numerical beach length

B 98 2 // use relaxation method 2 for wave generation

B 99 2 // use relaxation method 2 for numerical beach

D 10 4 // Conservative WENO for velocity convection

D 20 2 // Implicit diffusion for velocities

D 30 1 // Projection Method for the pressure

F 30 3 // 3rd-order Runge-Kutta scheme for Level Set time treatment

F 40 3 // 3rd-order Runge-Kutta scheme for Reinitialization time treatment

F 42 0.8 // length for level set initialization, instead of maximum length of domain

F 60 0.4 // still water level

I 12 1 // hydrostatic pressure initialization

N 40 3 // 3rd-order Runge-Kutta scheme for velocity time treatment

N 41 60.0 // Maximum simulation time

N 47 0.25 // factor for CFD criterion

M 10 4 // number of parallel processes

P 10 1 // turn on .vtu printout

P 30 0.5 // print out interval for .vtu files based on simulation time

W 22 -9.81 // gravity

P 52 0.005 // y-coordinate and print out water surface line wsfline in x-direction

P 53 1 // add theoretical wsfline to file

P 54 10 // print out interval for wsflines based on iterations

P 51 2.0 0.005 // Wave gage 1 location, will printed in the order given here

P 51 4.0 0.005 // Wave gage 2

P 51 5.2 0.005 // Wave gage 3

P 51 10.5 0.005 // Wave gage 4

P 51 12.5 0.005 // Wave gage 5

P 51 13.5 0.005 // Wave gage 6

P 51 14.5 0.005 // Wave gage 7

10.10.3 Results

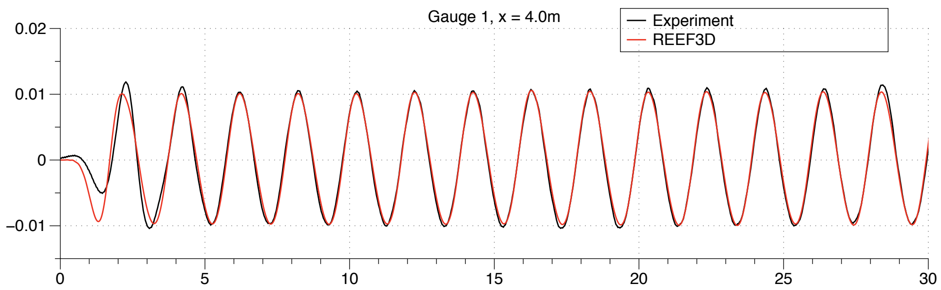


Figure 10.26: Wave gage 1

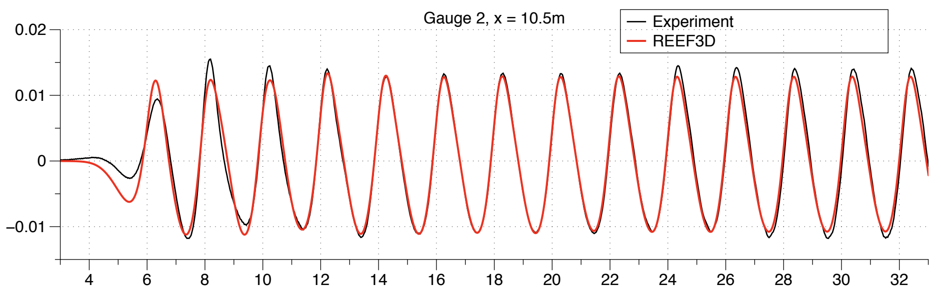


Figure 10.27: Wave gage 2

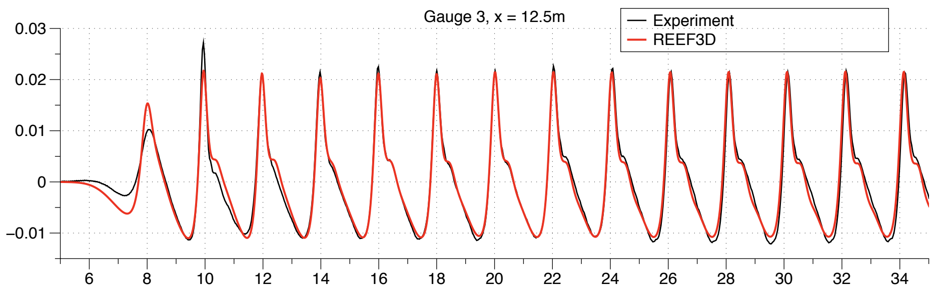


Figure 10.28: Wave gage 3

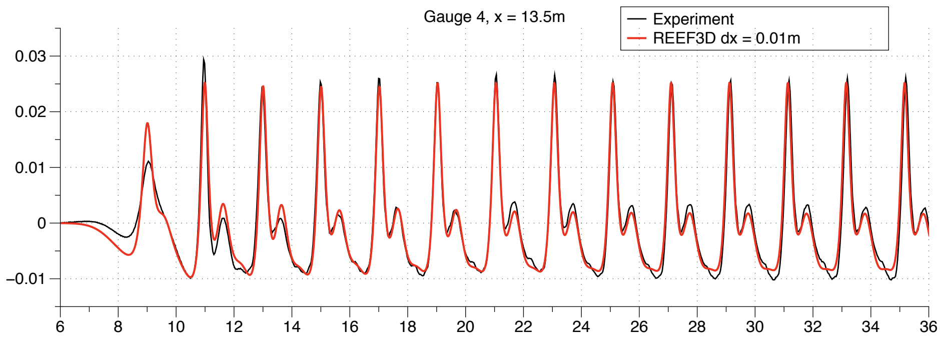


Figure 10.29: Wave gage 4

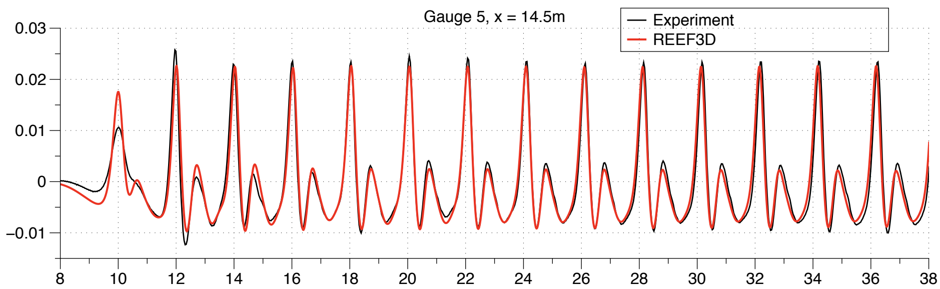


Figure 10.30: Wave gage 5

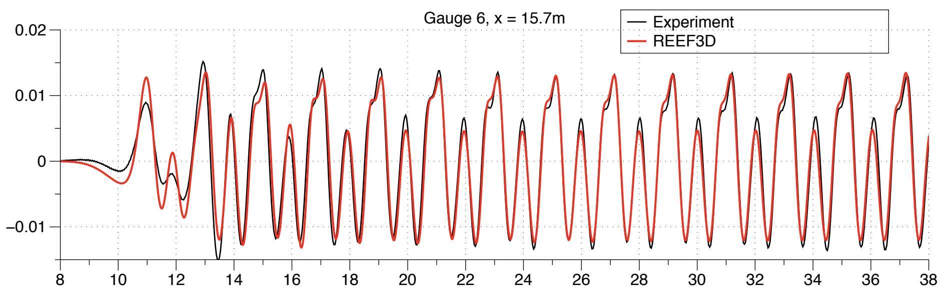


Figure 10.31: Wave gage 6

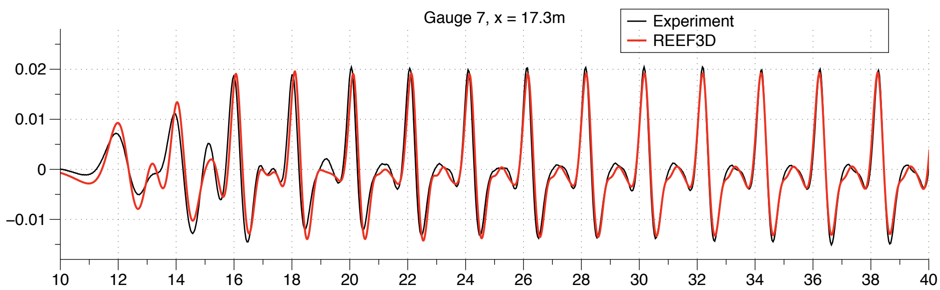


Figure 10.32: Wave gage 7

10.11 Plunging Breaking Waves over Slope

The well know benchmark case for plunging breaking waves over a submerged bar by Ting and Kirby [1995] is shown here. The breaking waves require a relatively finer grid, thus 128 cores on NOTUR's supercomputer facilities were used to simulate the case. For more information on this case and breaking waves over slopes in general, please have a look at theses in-depth studies: Alagan Chella et al. [2015b], Alagan Chella et al. [2015c] and Alagan Chella et al. [2015a].

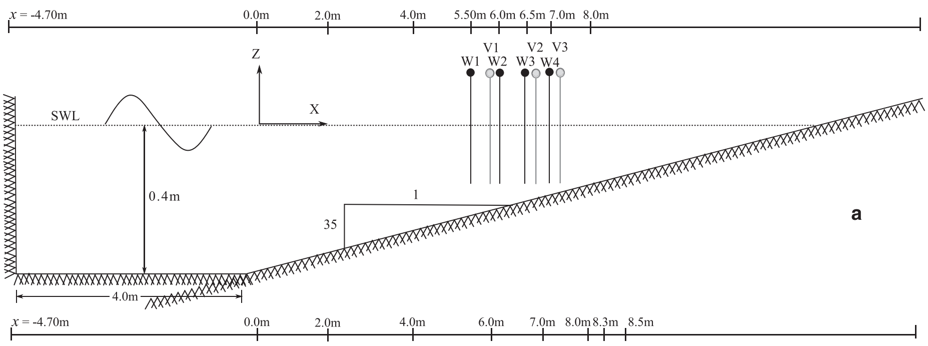


Figure 10.33: Experimtental setup plunging breaking waves over slope

10.11.1 DIVEMesh: control.txt

C 11 6 // left side: wave relaxation zone

C 12 3 // side: symmetry plane

C 13 3 // side: symmetry plane

C 14 7 // numerical relaxation beach

C 15 21 // bottom: wall boundary

C 16 3 // top symmetry plane

B 1 0.01 // mesh size dx

B 10 0.0 24.0 0.0 0.01 0.0 0.8 // rectangular domain size

O 61 6.0 12.0 0.0 0.01 0.0 0.3 // front wedge of the bar

O 10 12.0 14.0 0.0 0.01 0.0 0.3 // middle section of the bar

O 61 14.0 17.0 0.0 0.01 0.3 0.0 // back wedge of the bar

M 10 128 // number of parallel processes

M 20 2 // advanced domain decomposition

10.11.2 REEF3D: ctrl.txt

B 10 1 // use wall functions for the velocities

B 10 1 // use wall functions for the turbulence model

B 50 0.0001 // wall roughness k_s

B 90 1 // turn on the numerical wave tank

B 92 8 // use 5th-order cnoidal waves

B 93 0.128 5.0 // wave height, wave period

B 96 9.8 0.0 0.0 // wavegen relaxation length, numerical beach length

B 98 2 // use relaxation method 2 for wave generation

B 99 2 // no beach

D 10 4 // Conservative WENO for velocity convection

D 20 2 // Implicit diffusion for velocities

D 30 1 // Projection Method for the pressure

F 30 3 // 3rd-order Runge-Kutta scheme for Level Set time treatment

F 40 3 // 3rd-order Runge-Kutta scheme for Reinitialization time treatment

F 42 1.0 // length for level set initialization, instead of maximum length of domain

F 60 0.4 // still water level

I 12 1 // hydrostatic pressure initialization

N 40 3 // 3rd-order Runge-Kutta Scheme for velocity time treatment

N 41 60.0 // Maximum simulation time

N 47 0.25 // factor for CFD criterion

M 10 128 // number of parallel processes

P 10 1 // turn on .vtu printout

P 30 0.5 // print out .vtu files based on simulation time

P 40 1 // turn state file print out for hot-start

P 42 0.5 // state file print out interval

T 10 2 // k-ω turbulence model

T 36 2 // FSF boundary condition for turbulent dissipation

W 22 -9.81 // gravity

P 51 19.8 0.0025 // wave gage

```
P 51 20.8 0.0025 // wave gage

P 51 21.8 0.0025 // wave gage

P 51 22.1 0.0025 // wave gage

P 52 0.0025 // wsfline in x-direction

P 61 21.095 0.0025 0.35 // point probe

P 61 21.095 0.0025 0.25 // point probe

P 61 21.095 0.0025 0.4

P 61 22.025 0.0025 0.35 // point probe

P 61 22.025 0.0025 0.26 // point probe

P 61 22.025 0.0025 0.4 // point probe

P 61 21.595 0.0025 0.4 // point probe

P 61 22.145 0.0025 0.35 // point probe

P 61 22.145 0.0025 0.28 // point probe

P 61 22.145 0.0025 0.4 // point probe
```

10.11.3 Results

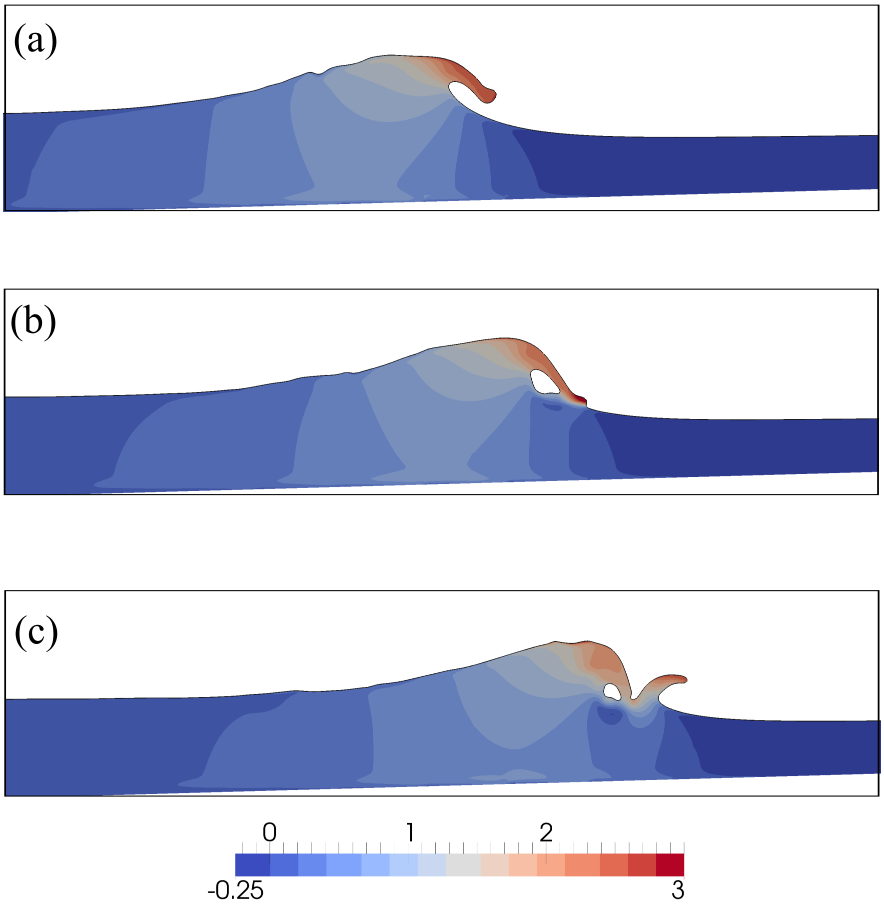


Figure 10.34: Variation of horizontal velocity, U (m/s) under the plunging breaker at t= 10.85s (a), 10.95s (b) and 11.05s (c)

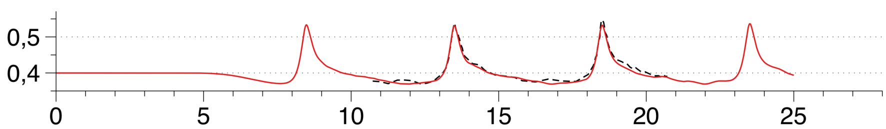


Figure 10.35: Wave gage 1, black experiment Ting and Kirby [1995], red REEF3D

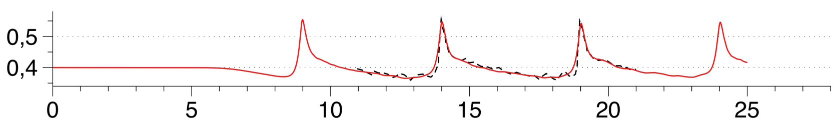


Figure 10.36: Wave gage 2, black experiment Ting and Kirby [1995], red REEF3D

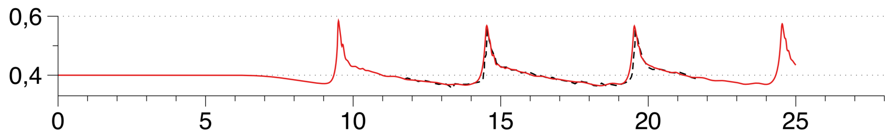


Figure 10.37: Wave gage 3, black experiment Ting and Kirby [1995], red REEF3D

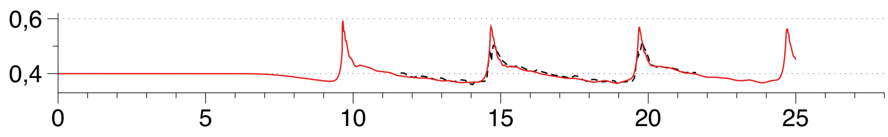


Figure 10.38: Wave gage 4, black experiment Ting and Kirby [1995], red REEF3D

10.12 Plunging Breaking Waves over Slope with Non-uniform Grid

The simulation of plunging breaking waves over a slope from the previous section by Ting and Kirby [1995] is shown here with a non-uniform grid (NUG) setup. The ctrl.txt is identical to the one above, with the exception of the M10 parameter.

10.12.1 DIVEMesh: control.txt

C 11 6 // left side: wave relaxation zone

C 12 3 // side: symmetry plane

C 13 3 // side: symmetry plane

C 14 7 // numerical relaxation beach

C 15 21 // bottom: wall boundary

C 16 3 // top symmetry plane

B 1 0.005

B 2 3000 1 80 // mesh size: N_x, N_y, N_z

B 10 0.0 30.0 0.0 0.005 0.0 1.0 // rectangangular domain size

S 61 13.8 30.0 0.0 0.005 0.0 0.463 // solid slope

M 10 12 // number of parallel processes

M 20 2 // advanced domain decomposition

B 101 5 // point focus using sinh stretching function in x-direction

B 111 2.5 // stretching factor in x-direction

B 114 20.0 // focus point for stretching in x-direction

B 103 5 // point focus using sinh stretching function in x-direction

B 113 2.5 // stretching factor in z-direction

B 116 0.40 // focus point for stretching in z-direction

10.13 Shoaling over Irregular Topography

In this case, waves propagation irregular topography is calculated using phase accurate irregular waves based on Boers' experiment Boers [1996]. Here, the bed topography is generated using a 'geo.dat' file with xyz-point data. These coordinates are interpolated using a local inverse distance algorithm in DIVEMesh. The waves are generated using wave reconstruction, are very convenient way to create phase accurate irregular waves based on a single wave gage Aggarwal et al. [2018b].

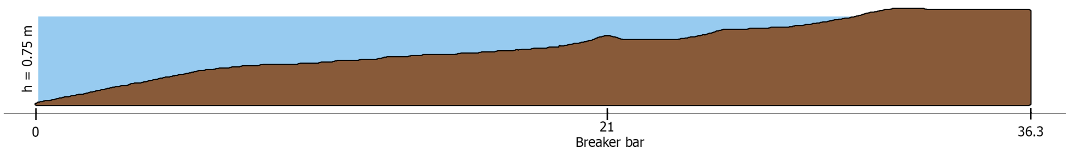


Figure 10.39: Experimtental setup plunging breaking waves over slope Elakel [2018]

10.13.1 DIVEMesh: control.txt

C 11 6 // left side: wave relaxation zone

C 12 3 // side: symmetry plane

C 13 3 // side: symmetry plane

C 14 7 // numerical relaxation beach

C 15 21 // bottom: wall boundary

C 16 3 // top symmetry plane

B 1 0.01 // mesh size dx

B 10 0.0 36.3 0.0 0.01 0.0 1.00 // rectangular domain size

G 10 1 // turn on geodat interpolation

G 15 2 // use local inverse distance interpolation

G 31 14 // number of smoothing iterations

M 10 128 // number of parallel processes

M 20 2 // advanced domain decomposition

10.13.2 DIVEMesh: geo.dat

In the 'geo.dat' file all bathymetry coordinates are given in ASCII format and are then used to interpolate into the bed by DIVEMesh. The bed can be visualized with the zero contour of 'topo' in the vtu files read by ParaView.

10.13.3 REEF3D: ctrl.txt

B 10 1 // use wall functions for the velocities

B 10 1 // use wall functions for the turbulence model

B 50 0.0001 // wall roughness k_s

B 90 1 // turn on the numerical wave tank

B 92 51 // wave reconstruction based on first-order irregular waves

B 98 3 // use Dirichlet wave generation

B 99 0 // no beach

D 10 4 // Conservative WENO for velocity convection

D 20 2 // Implicit diffusion for velocities

D 30 1 // Projection Method for the pressure

F 30 3 // 3rd-order Runge-Kutta scheme for Level Set time treatment

F 40 3 // 3rd-order Runge-Kutta scheme for Reinitialization time treatment

F 42 1.0 // length for level set initialization, instead of maximum length of domain

F 60 0.75 // still water level

I 12 1 // hydrostatic pressure initialization

N 40 3 // 3rd-order Runge-Kutta Scheme for velocity time treatment

N 41 200.0 // Maximum simulation time

N 47 0.25 // factor for CFD criterion

M 10 128 // number of parallel processes

P 10 1 // turn on .vtu printout

P 30 0.5 // print out .vtu files based on simulation time

P 40 1 // turn state file print out for hot-start

P 42 0.5 // state file print out interval

T 10 2// k-ω turbulence model

T 36 2 // FSF boundary condition for turbulent dissipation

G 50 1 // look for geodata points in grid file

W 22 -9.81 // gravity

P 51 0.0 0.005 // wave gage

P 51 4.0 0.005 // wave gage

P 51 8.0 0.005 // wave gage

P 51 12.0 0.005 // wave gage

P 51 16.0 0.005 // wave gage

P 51 20.0 0.005 // wave gage

P 61 14.37 0.005 0.74 // point probe

10.13.4 REEF3D: waverecon.dat

The 'waverecon.dat' files contains the individual wave components including amplitude, frequency and phase shift in order to generate phase accurate irregular waves.

10.13.5 Results

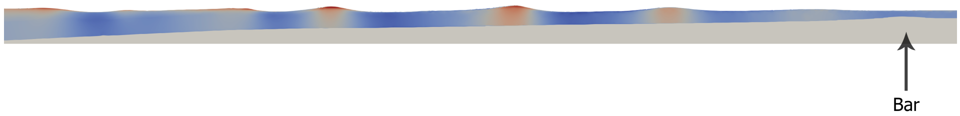


Figure 10.40: Shoaling of irregular waves for the Boers case Elakel [2018]

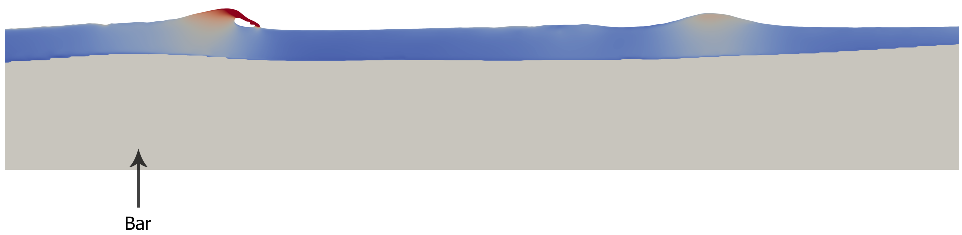


Figure 10.41: Breaking waves on the bar for the Boers case Elakel [2018]

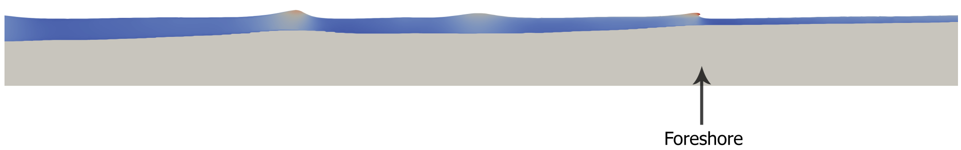


Figure 10.42: Breaking waves on the foreshore for the Boers case Elakel [2018]

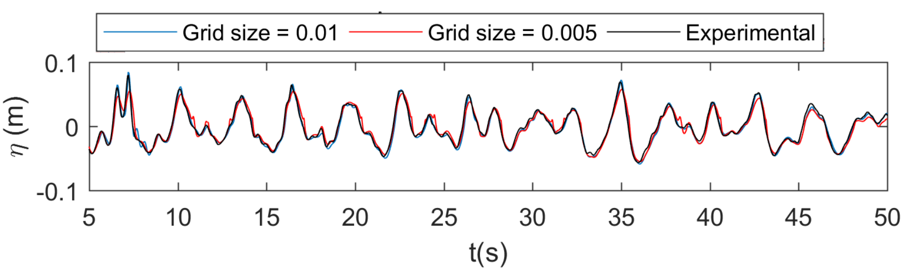


Figure 10.43: Wave gage 1

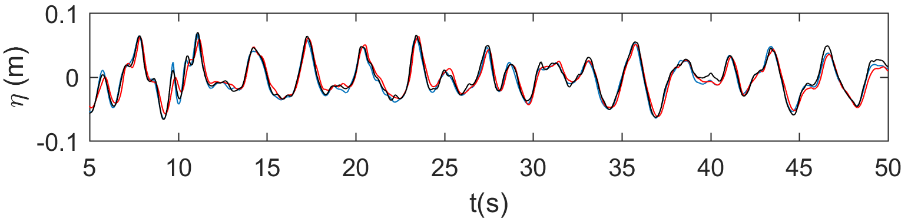


Figure 10.44: Wave gage 2

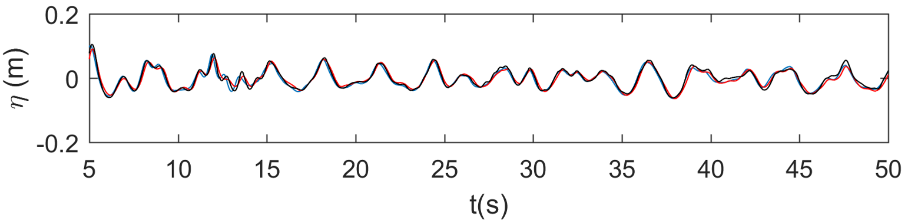


Figure 10.45: Wave gage 3

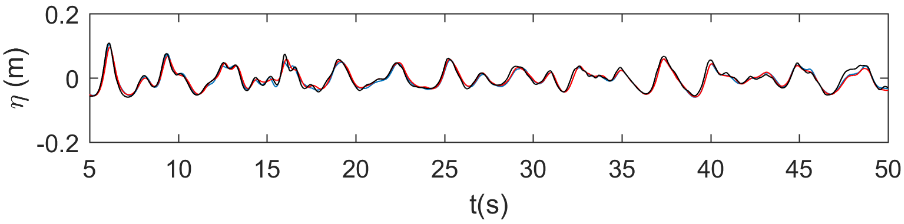


Figure 10.46: Wave gage 4

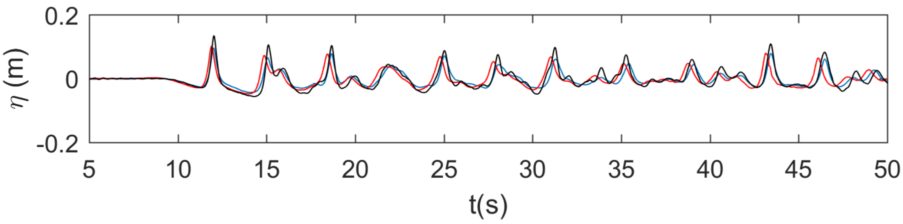


Figure 10.47: Wave gage 5

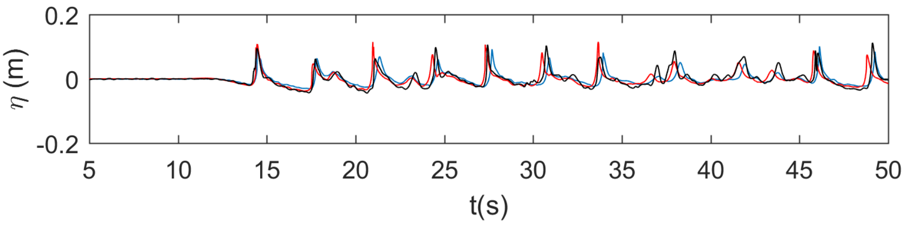


Figure 10.48: Wave gage 6

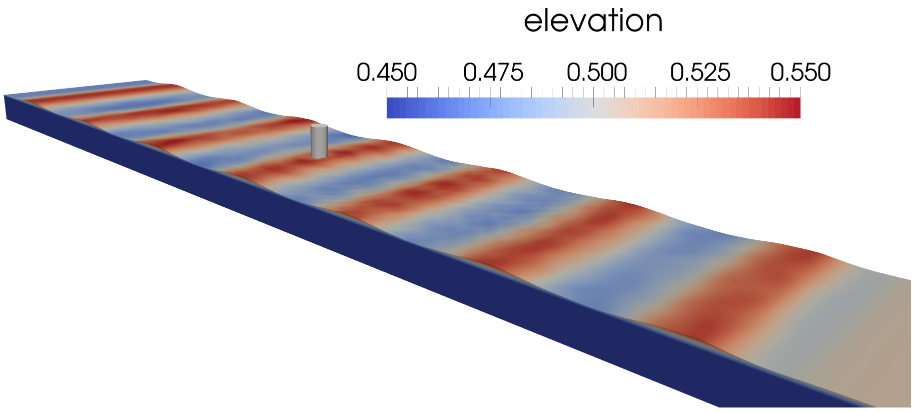


Figure 10.49: Non-breaking wave forces

10.14 Non-Breaking Wave Forces

In many scenarios, wave forces need to be computed. Based on the experiments by Chen et al. [2014], numerical benchmark results were produced, see Bihs et al. [2016a] for more detail. In addition to previous NWT examples, the use of the force box is shown in this example. The forces on a solid within the force box is calculated and printed out. It is important to give some margin around the solid in order capture all surface facets in the pressure integration algorithm. More REEF3D related literature on non-breaking wave forces can be found here: Kamath et al. [2015b], Kamath et al. [2016b] and Kamath et al. [2015a].

10.14.1 DIVEMesh: control.txt

C 11 6 // left side: wave relaxation zone

C 12 21 // side: wall boundary

C 13 21 // side: wall boundary

C 14 7 // numerical relaxation beach

C 15 21 // bottom: wall boundary

C 16 3 // top symmetry plane

B 1 0.025 // mesh size dx

B 10 0.0 18.0 0.0 3.0 0.0 1.0 // rectangular domain size

S 33 7.50 1.50 0.125 // solid vertical cylinder

M 10 8 // number of parallel processes

M 20 2 // advanced domain decomposition

10.14.2 REEF3D: ctrl.txt

B 10 1 // use wall functions for the velocities

B 10 1 // use wall functions for the turbulence model

B 50 0.0001 // wall roughness k_s

B 90 1 // turn on the numerical wave tank

B 92 4 // use 2nd-order Stokes waves

B 93 0.07 1.22 // wave height, wave period

B 96 2.11 4.22 // wavegen relaxation length, numerical beach length

B 98 2 // use relaxation method 2 for wave generation

B 99 2 // relaxation beach

D 10 4 // Conservative WENO for velocity convection

D 20 2 // Implicit diffusion for velocities

D 30 1 // Projection Method for the pressure

F 30 3 // 3rd-order Runge-Kutta scheme for Level Set time treatment

F 40 3 // 3rd-order Runge-Kutta scheme for Reinitialization time treatment

F 42 1.0 // length for level set initialization, instead of maximum length of domain

F 60 0.505 // still water level

I 12 1 // hydrostatic pressure initialization

N 40 3 // 3rd-order Runge-Kutta Scheme for velocity time treatment

N 41 50.0 // Maximum simulation time

N 47 0.25 // factor for CFD criterion

M 10 8 // number of parallel processes

P 10 1 // turn on .vtu printout

P 35 15.0 30.0 0.05 // print out interval for .vtu files based on simulation time

P 40 1 // turn state file print out for hot-start

P 42 1.0 // state file print out interval

T 10 2 // k-ω turbulence model

T 36 2 // FSF boundary condition for turbulent dissipation

W 22 -9.81 // gravity

P 51 5.00 1.50 // wave gage

P 55 1.0 // print out wave gage based on simulation time

P 81 7.325 7.675 1.325 1.675 0.0 1.0 // force box coordinates

10.14.3 Results

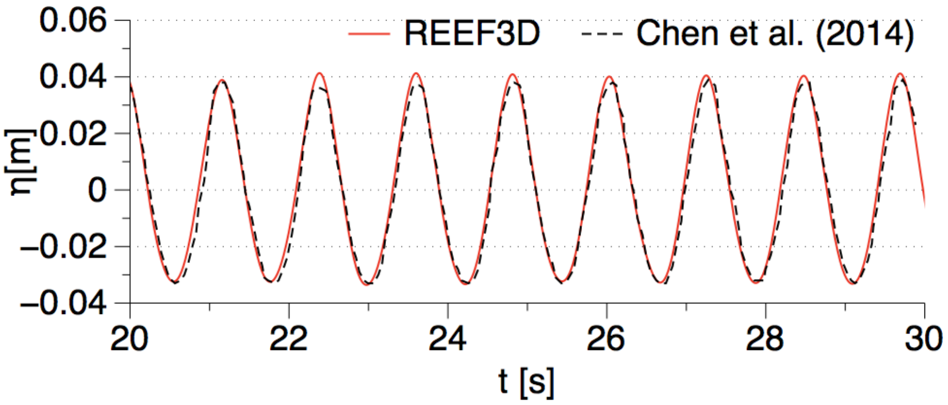


Figure 10.50: Free surface comparison between REEF3D and the experimental data Chen et al. [2014]

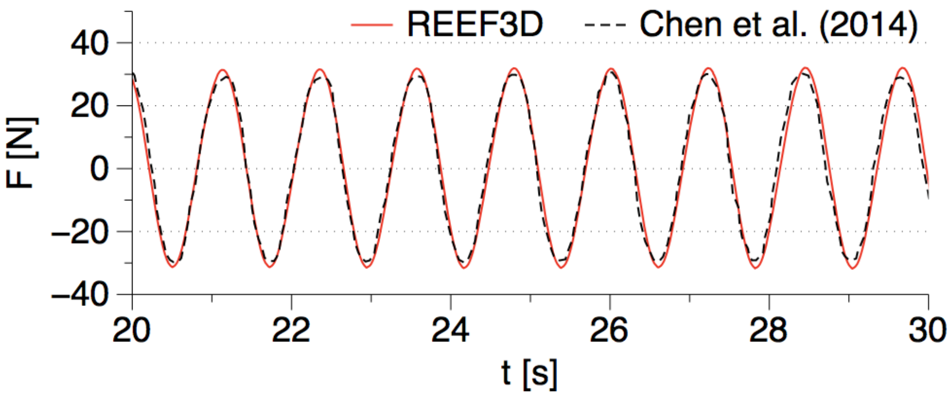


Figure 10.51: Wave Force comparison between REEF3D and the experimental data Chen et al. [2014]

10.15 Breaking Wave Forces

Extending the previous scenario, breaking wave forces are simulated by using the case from Kamath et al. [2016a] based on an experimental study in the GWK in Hannover Irschik et al. [2002]. The waves are shoaling on a 1:10 slope and are breaking directly at the vertical cylinder. The slamming forces a calculated using the force box approach. REEF3D has been extensively validated and applied for breaking wave forces: Bihs et al. [2016b], Alagan Chella et al. [2019c], Alagan Chella et al. [2019b], Alagan Chella et al. [2019a]

10.15.1 DIVEMesh: control.txt

C 11 6 // left side: wave relaxation zone

C 12 21 // side: wall boundary

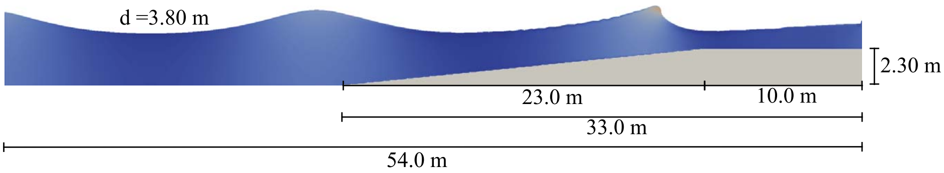


Figure 10.52: Breaking wave forces setup

C 13 21 // side: wall boundary

C 14 8 // numerical AWA beach

C 15 21 // bottom: wall boundary

C 16 3 // top symmetry plane

B 1 0.05 // mesh size dx

B 10 0.0 54.0 0.0 5.0 0.0 7.0 // rectangular domain size

S 61 21.0 44.0 0.0 5.0 0.0 2.30 // solid slope

S 10 44.0 54.0 0.0 5.0 0.0 2.30 // solid flat

O 33 44.0 2.50 0.35 // vertical cylinder

M 10 8 // number of parallel processes

M 20 2 // advanced domain decomposition

10.15.2 REEF3D: ctrl.txt

B 10 1 // use wall functions for the velocities

B 10 1 // use wall functions for the turbulence model

B 50 0.0001 // wall roughness k_s

B 90 1 // turn on the numerical wave tank

B 92 5 // use 5th-order Stokes waves

B 93 1.3 4.0 // wave height, wave period

B 96 21.0 0.0 // wavegen relaxation length, numerical beach length

B 98 2 // use relaxation method 2 for wave generation

B 99 3 // AWA beach

D 10 4 // Conservative WENO for velocity convection

D 20 2 // Implicit diffusion for velocities

D 30 1 // Projection Method for the pressure

F 30 3 // 3rd-order Runge-Kutta scheme for Level Set time treatment

F 40 3 // 3rd-order Runge-Kutta scheme for Reinitialization time treatment

F 42 7.0 // length for level set initialization, instead of maximum length of domain

F 60 3.8 // still water level

I 12 1 // hydrostatic pressure initialization

N 40 3 // 3rd-order Runge-Kutta Scheme for velocity time treatment

N 41 50.0 // Maximum simulation time

N 47 0.25 // factor for CFD criterion

M 10 8 // number of parallel processes

P 10 1 // turn on .vtu printout

P 35 23.5 25.2 0.05 // print out interval for .vtu files based on simulation time

P 40 1 // turn state file print out for hot-start

P 42 1.0 // state file print out interval

T 10 2 // k-ω turbulence model

T 36 2 // FSF boundary condition for turbulent dissipation¨

W 22 -9.81 // gravity

P 81 43.30 44.70 1.80 3.20 2.30 8.0 // force box coordinates

P 10 1 // print .vtu file

P 35 23.5 25.2 0.05 // print .vtu file in interval

P 180 1 // print fsf files

P 182 0.1 // print out fsf files based on simulation time

P 51 22.0 2.50 // wave gage

P 51 30.0 2.50 // wave gage

P 51 35.0 2.50 // wave gage

P 51 40.0 2.50 // wave gage

P 51 43.65 4.80 // wave gage

P 51 43.65 2.50 // wave gage

P 51 44.5 2.505 // wave gage

P 51 47.0 2.5 // wave gage

P 51 50.5 2.5 // wave gage

P 51 52.0 2.5 // wave gage

P 51 53.0 2.5 // wave gage

10.15.3 Results

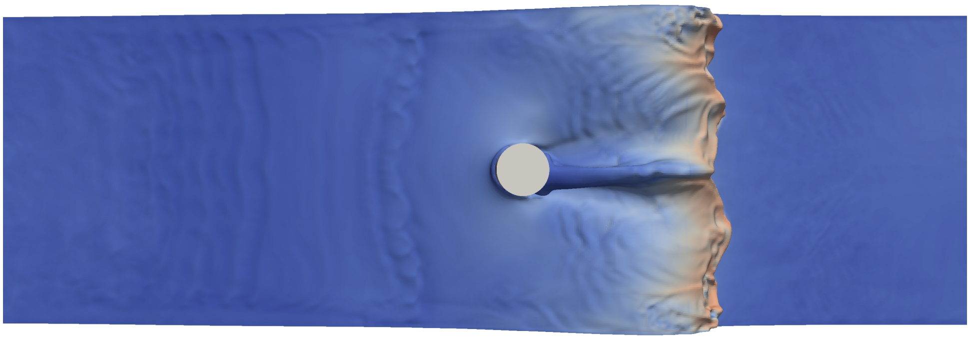


Figure 10.53: Breaking wave impact with REEF3D, plane view

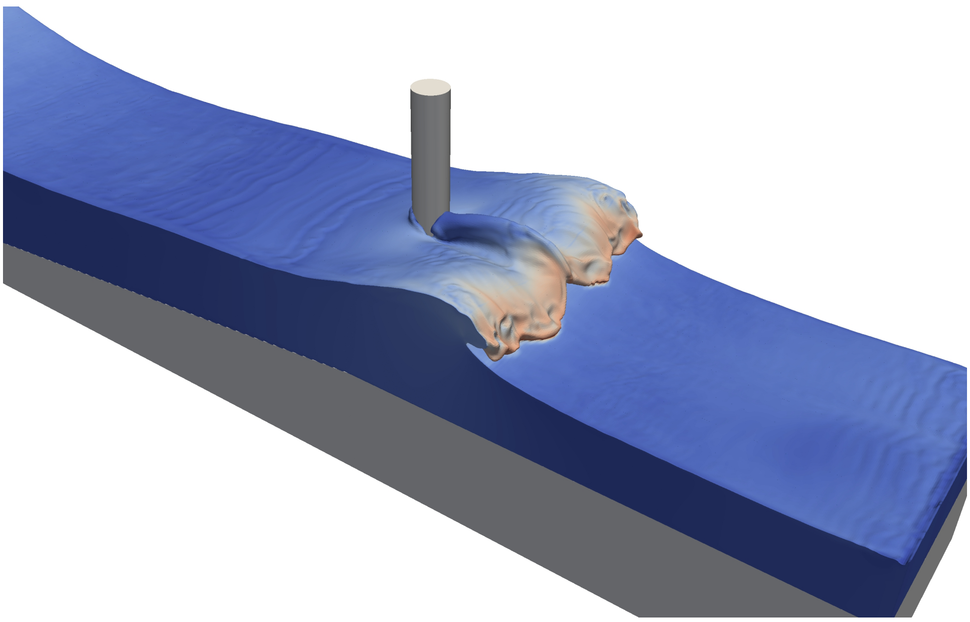


Figure 10.54: Breaking wave impact with REEF3D, side view

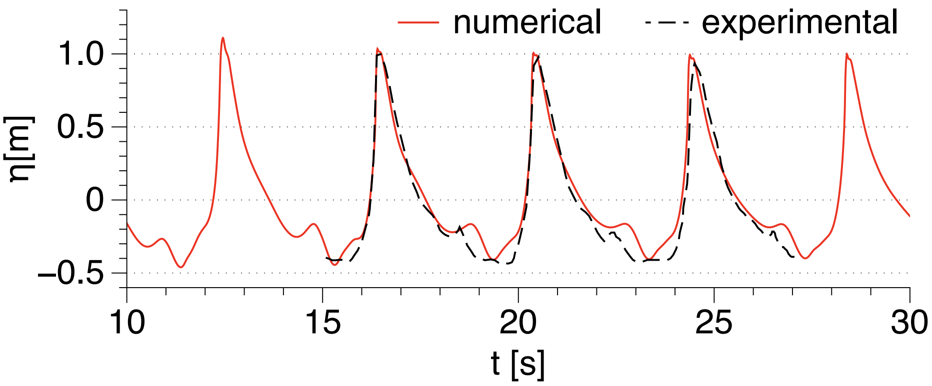


Figure 10.55: Wave Force comparison between REEF3D and the experimental data Irschik et al. [2002]

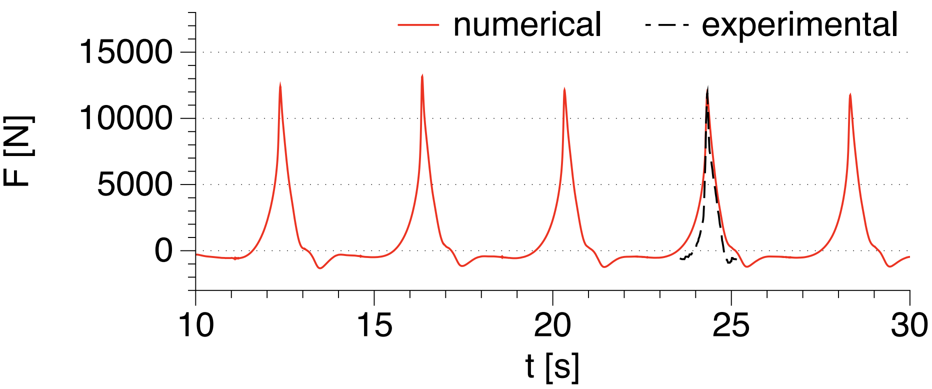


Figure 10.56: Free surface comparison between REEF3D and the experimental data Irschik et al. [2002]

10.16 Heave decay of a sphere with Non-uniform Grid

Setup for simulations including floating bodies using the continuous direct forcing approach. As an example, the heave decay of a sphere can be simulated using the following input files together with the STL-file provided in the tutorial folder. It is recommended to use the cell-based stretching shown in this example to define a box with uniform cell sizes around the floating body.

10.16.1 DIVEMesh: control.txt

C 11 21 // left side: wall boundary

C 12 21 // side: wall boundary

C 13 21 // side: wall boundary

C 14 21 // right side: wall boundary

C 15 21 // bottom: wall boundary

C 16 3 // top: symmetry plane

B 1 0.05 // This value should the first number in the cell-based stretching option

B 10 0.0 6.0 0.0 6.0 0.0 4.0 // rectangular domain size

B 101 11 // cell-based stretching function in x-direction

B 127 0.05 0.3 3.0 1.5 1.1 // options for cell-based stretching function in x-direction

B 102 11 // cell-based stretching function in y-direction

B 128 0.05 0.3 3.0 1.5 1.1 // options for cell-based stretching function in x-direction

B 103 11 // cell-based stretching function in z-direction

B 129 0.05 0.3 2.0 1.5 1.1 // options for cell-based stretching function in x-direction

M 10 4 // number of parallel processes

M 20 2 // advanced domain decomposition

10.16.2 REEF3D: ctrl.txt

B 10 1 // use wall functions for velocities (not floating body)

B 50 0.00001 // wall roughness

B 90 1 // turn on the numerical wave tank

B 99 2 // use relaxation method 2 for numerical beach

B 107 0.0 0.0 0.0 6.0 0.3 // define relaxation zone at domain boundaries

B 107 6.0 6.0 0.0 6.0 0.3 // define relaxation zone at domain boundaries

B 107 0.0 6.0 0.0 0.0 0.3 // define relaxation zone at domain boundaries

B 107 0.0 6.0 6.0 6.0 0.3 // define relaxation zone at domain boundaries

D 10 4 // conservative WENO discretization for velocity convection

D 20 2 // implicit diffusion for velocities

D 30 1 // projection method for pressure

F 30 3 // 3rd-order Runge-Kutta scheme for level set time treatment

F 35 5 // HJ-WENO discretization for level set convection

F 40 3 // 3rd-order Runge-Kutta scheme for reinitialization time treatment

F 60 2.0 // still water level

I 10 1 // initialize numerical wave tank

N 40 4 // floating algorithm uses low-storage 3rd-order Runge-Kutta scheme internally

N 41 10 // total simulation time

```
N 47 0.3 // factor for CFL criterion
M 10 4 // number of parallel processes
P 10 1 // turn on vtu printout
P 30 0.1 // print out vtl files interval based on simulation time
W 1 1000.0 // density of water
W 22 -9.81 // gravitational acceleration
X 10 1 // turn on 6DOF simulation
X 11 1 1 1 0 0 0 // enable translational motions
X 21 500.0 // density of floating body
X 180 1 // use STL geometry
X 182 3.0 3.0 2.2 // move geometry to initial position
```

Bibliography

M. A. Afshar. Numerical wave generation in OpenFOAM. Master's thesis, Chalmers University of Technology, 2010.

A. Aggarwal, M. Alagan Chella, H. Bihs, C. Pakozdi, P. A. Berthelsen, and Ø. A Arntsen. CFD-based study of steep irregular waves for extreme wave spectra. *International Journal of Offshore and Polar Engineering*, 28(2):164–170, 2018a.

A. Aggarwal, C. Pakozdi, H. Bihs, D. Myrhaug, and M. Alagan Chella. Free surface reconstruction for phase accurate irregular wave generation. *Journal of Marine Science and Engineering*, 6(105):1–23, 2018b.

M. Alagan Chella, H. Bihs, and D. Myrhaug. Characteristics and profile asymmetry properties of waves breaking over an impermeable submerged reef. *Coastal Engineering*, 100:26–36, 2015a.

M. Alagan Chella, H. Bihs, D. Myrhaug, and M. Muskulus. Breaking characteristics and geometric properties of spilling breakers over slopes. *Coastal Engineering*, 95:4–19, 2015b.

M. Alagan Chella, Hans Bihs, Dag Myrhaug, and Michael Muskulus. Hydrodynamic characteristics and geometric properties of plunging and spilling breakers over impermeable slopes. *Ocean Modelling, Virtual Special Issue: Ocean Surface Waves*, pages 1–20, 2015c.

M. Alagan Chella, H. Bihs, and Myrhau D. Numerical modeling of breaking wave kinematics and wave impact pressures on a vertical slender cylinder,. *Journal of Fluids and Structures*, 86:94–123, 2019a.

M. Alagan Chella, H. Bihs, Myrhau D., and Ø. A Arntsen. Numerical modeling of breaking wave kinematics and wave impact pressures on a vertical slender cylinder,. *Journal of Offshore Mechanics and Arctic Engineering*, 141(5):1–10, 2019b.

M. Alagan Chella, H. Bihs, A. Kamath, Myrhau D., and Ø. A Arntsen. Breaking wave interaction with a group of four vertical slender cylinders in two square arrangements. *Journal of Offshore Mechanics and Arctic Engineering*, 141(6):1–10, 2019c.

S. Beji and J. A. Battjes. Experimental investigation of wave propagation over a bar. *Coastal Engineering*, 19:151–162, 1993.

H. Bihs, A. Kamath, M. Alagan Chella, A. Aggarwal, and Ø. A. Arntsen. A new level set numerical wave tank with improved density interpolation for complex wave hydrodynamics. *Computers & Fluids*, 140:191–208, 2016a.

H. Bihs, A. Kamath, M. Alagan Chella, and Ø. A Arntsen. Breaking-wave interaction with tandem cylinders under different impact scenarios. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 2016b.

M. Boers. *Simulation of a surf zone with a barred beach. Report 1. Wave heights and wave breaking*. PhD thesis, Report, Department of Civil Engineering, Delft Technical University, 1996.

hyper high performance preconditioners - User's Manual. Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 2015.

L. F. Chen, J. Zang, A. J. Hillis, G. C. J. Morgan, and A. R. Plummer. Numerical investigation of wave–structure interaction using openFOAM. *Ocean Engineering*, 88:91—109, 2014.

M. Elakel. *Investigation of wave transformation and breaking processes in the coastal zone using REEF3D*. PhD thesis, Master Thesis, Marine Civil Engineering, NTNU Trondheim, 2018.

R. D. Falgout, J. E. Jones, and U. M. Yang. *Numerical Solution of Partial Differential Equations of Parallel Computers*, chapter The Design and Implementation of hypre, a Library of Parallel High Performance Preconditioners. Lecture Notes in Computational Science and Engineering. Springer, 2006.

K. Irschik, U. Sparboom, and H. Oumeraci. Breaking wave characteristics for the loading of a slender pile. In *Proc. 28th International Conference on Coastal Engineering, Cardiff, Wales*, 2002.

N. G. Jacobsen, D. R. Fuhrman, and J. Fredsøe. A wave generation toolbox for the open-source CFD library: OpenFOAM. *International Journal for Numerical Methods in Fluids*, 70(9):1073–1088, 2012.

A. Kamath, M. Alagan Chella, H. Bihs, and Ø. A. Arntsen. Evaluating wave forces on groups of three and nine cylinders using a 3D numerical wave tank. *Engineering Applications of Computational Fluid Mechanics*, 2015a.

A. Kamath, M. Alagan Chella, H. Bihs, and Ø. A Arntsen. Cfd investigations of wave interaction with a pair of large tandem cylinders. *Ocean Engineering*, 108:738–748, 2015b.

A. Kamath, M. Alagan Chella, H. Bihs, and Ø. A Arntsen. Breaking wave interaction with a vertical cylinder and the effect of breaker location. *Ocean Engineering*, 128:105–115, 2016a.

A. Kamath, H. Bihs, M. Alagan Chella, and Øivind A Arntsen. Upstream-cylinder and downstream-cylinder influence on the hydrodynamics of a four-cylinder group. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 2016b. doi: 10.1061/(ASCE)WW.1943-5460.0000339.

M. Peric. Why STAR-CCM+ is the next step for cfd. *dynamics*, 23:3–4, 2004.

F. C. K. Ting and J. T. Kirby. Dynamics of surf-zone turbulence in a strong plunging breaker. *Coastal Engineering*, 24:177–204, 1995.